Semantic Independence in DL-Programs*

Thomas Eiter, Michael Fink, and Daria Stepanova

Institute of Information Systems
Vienna University of Technology
Favoritenstraße 9-11, A-1040 Vienna, Austria
{dasha,eiter,fink}@kr.tuwien.ac.at

Abstract. Description Logic programs (DL-programs) are a prominent approach for a loose coupling of rules and ontologies, which has become a topic of increased interest. When computing answer sets of a DL-program, special DL-atoms, which provide query interface to an ontology, are evaluated under a possibly changing input that gives a context for the evaluation. Many different such contexts may exist and evaluating a DL-atom may be costly even for one context. Thus a natural question to ask is when the evaluation is independent of the context. Such information has immediate applications in optimization of DL-programs, but is also beneficial for other reasoning tasks, like inconsistency diagnosis and program repair. We provide an answer to this question based on a semantic notion of independence and provide a complete characterization of independent DL-atoms. We then extend the characterization to independence under additional information about inclusions among rule predicates. Moreover, we develop an axiomatization which allows one to derive all tautological DL-atoms in the general case and under predicate inclusions. A complexity analysis reveals that checking whether a DL-atom is independent, can be done efficiently.

1 Introduction

DL-programs are a prominent approach for the loose coupling of rules and ontologies, in which the rules and the ontology part exchange information via a well-defined interface.[1] In general, a DL-atom specifies an update of an ontology prior to querying it; e.g. $\mathrm{DL}[C \uplus p;\ C](t)$; means that assertions $C(t)$ are made for each individual t such that p is true for t in the rules part. Several semantics of such programs have been defined, cf. [4; 9; 13; 19; 17; 3], and the concept of DL-atom has been adopted and generalized by other formalisms e.g. [18; 10; 6].

Irrespective of a particular semantics, for the evaluation of DL-programs in practice (i.e., when computing models or answer sets) individual DL-atoms have to be evaluated under varying input in general. Thus, the possible ontology updates specified by a DL-atom define respective contexts for their evaluation, and many different such contexts may need to be considered. Moreover, even for one context evaluating the query specified by the DL-atom in this context may be costly. Therefore, developing optimization

* This work is partially supported by the Austrian Science Fund (FWF) project P20840, and by the EC FP7 project OntoRule (IST-2009-231875).

[1] For further discussion of loose and strong couplings and their strengths, cf. [15; 4; 8]

techniques, e.g. caching techniques [7], partial evaluation and atom merging [6], is necessary for the development of effective solvers.

Caching techniques, for instance, aim at memorizing the value of a DL-atom for some inputs, and to conclude about its value on a new input. However, the very question whether its value is on all inputs the same has not been considered so far; we call DL-atoms with this property *independent*. The identification of independent DL-atoms has immediate applications in optimization, as such atoms respectively rules involving them can be removed from the DL-program.

However, information about independence has also other uses. The loose coupling by DL-programs may result in inconsistency, that is, that no answer set (i.e., suitable model) of a DL-program exists. To remedy the situation, an inconsistency-tolerating approach was developed in [16; 8]. In this approach, one distills a set of DL-atoms (a "diagnosis") which has the "wrong value" in establishing an answer set, meaning that if these atoms and rules involving them are ignored, then an answer set exists. Based on such diagnoses, one can think of repairing the ontology part of the DL-program, by changing the axioms such that consistency is gained. However, the value of a DL-atom can be independent of the underlying ontology (or the initial one modulo a set of changes); thus some of the diagnoses are false positives, i.e., the opposite value can never be established.

This problem can be avoided by identifying independent DL-atoms, for instance tautologic ones. However, it is not always obvious that a certain DL-atom is tautologic. Let us illustrate this by an example.

Example 1. Consider the following DL-program representing information in the fruit domain: $P = (\Phi, \Pi)$ with underlying ontology Φ and the rules part

$$\Pi = \left\{ \begin{array}{ll} (1) \quad so(pineapple, chile). & (2) \quad vi(X) \leftarrow ex(X). \\ (3) \quad sw(X) \leftarrow ex(X), not\ bi(X). & (4) \quad ex(X) \leftarrow so(X, Y). \\ (5) \quad no(X) \leftarrow \mathrm{DL}[H \uplus vi, H \sqcup sw, A \sqcap ex;\ \neg A](X). & \end{array} \right\},$$

where predicate so stands for Southern fruit with its country of origin, vi for vitamin, ex for exotic, bi for bitter, sw for sweet, and no for non-African fruit, respectively. Moreover, H stands for the concept of healthiness and A for the concept of African fruit. Here (1) is the fact that pineapple is a Southern fruit possibly from Chile, rule (2) states that exotic fruits are rich of vitamin and rule (3) that exotic fruits are sweet, unless they are known to be bitter. Rule (4) says that Southern fruits are exotic. Finally, rule (5) contains a DL-atom in its body. Informally, it selects all objects o into no such that $\neg A(o)$, i.e., that it is a not an African fruit is provable from the ontology Φ, upon the (temporary) assertions that vitamin objects are healthy, sweet ones are unhealthy, and the restriction that only fruit known to be exotic may be African.

It is not straightforward for this DL-atom, nor for any of its instances, that it is tautologic; this however will be shown in Section 4.

If we adopt the reasonable assumption that the underlying ontology is satisfiable, another kind of independence is possible: DL-atoms which are *contradictory*, i.e., always evaluate to false.

Our contributions on identifying independent DL-atoms briefly are as follows:

• Based on a semantic notion of independence, we provide a syntactic characterization of independent DL-atoms. While tautologic DL-atoms have a rich structure, contradictory DL-atoms are simple and only possible without ontology update prior to query evaluation.
• We also consider relaxed forms of tautologies, relative to additional information on rule predicates (acting as constraints on the possible updates to the ontology). In particular, we study inclusion among rule predicates.
• We develop a complete axiomatization for deriving all tautologies by means of simple rules of inference, in the general case as well as under separable inclusion constraints, i.e., without projective input inclusions.
• We determine the complexity of the calculus. It turns out that tautology checking is feasible in polynomial time (more precisely, in NLogSpace in general, and in LogSpace, in fact it is first-order expressible, for non-negative queries), also relative to separable inclusion constraints (in this case, it is NLogSpace-complete). Thus, we establish that checking whether a given DL-atom is independent can be done efficiently.

Our results provide further insight into the nature of DL-programs. In particular, they might be useful for DL-programs that are automatically constructed (like the ones encoding a fragment of Baader and Hollunder's terminological default logic over ontologies [2]). They can be applied to simplify DL-programs, as well as in inconsistency analysis, e.g., to refine inconsistency-tolerating semantics of DL-programs [16].

2 Preliminaries

2.1 Description Logics

We assume that the reader is familiar with the basics of Description Logics (DLs) and their syntax and semantics [1]. We will consider DL knowledge bases defined over signatures $\Sigma_o = \langle \mathcal{F}, \mathcal{P}_o \rangle$ with a set $\mathcal{F}$ of individuals (constants) c and a set $\mathcal{P}_o = \mathcal{P}_c \cup \mathcal{P}_r$ of (atomic) concept names $\mathcal{P}_c$ and role names $\mathcal{P}_r$; concept expressions, role expressions are defined as usual, as well as concept inclusion axioms $C \sqsubseteq D$, role axioms (if any are available), and assertion axioms. A *DL knowledge base (or ontology)* in a DL $\mathcal{L}$ is then a (finite) set Φ of axioms in $\mathcal{L}$.

We do not commit to a particular DL $\mathcal{L}$ here, but as for DL-programs assume that

- assertions $C(a)$, $\neg C(a)$, $R(a,b)$, $\neg R(a,b)$ with $C \in \mathcal{P}_c$, $R \in \mathcal{P}_r$ and $a, b \in \mathcal{F}$ are admissible in $\mathcal{L}$ (or can be simulated), and
- $\Phi \models \phi$ denotes, under the usual model-based semantics of $\mathcal{L}$, logical entailment of a formula ϕ from Φ, i.e., each model Φ satisfies ϕ.

For instance, the DLs $\mathcal{SHIF}$, $\mathcal{SHOIN}$, and $\mathcal{SROIQ}$, which provide the logical underpinnings of OWL-Lite, OWL DL and OWL 2, respectively (see, e.g., [11; 12; 14]), and the lightweight DL $DL\text{-}Lite_{\mathcal{R}}$ fulfill this. [2]

In particular, we consider *DL-queries*, i.e., formulas $\phi = Q(\boldsymbol{t})$ such that Q is either

(a) a concept inclusion $C \sqsubseteq D$ or its negation $\neg(C \sqsubseteq D)$, with $\boldsymbol{t} = \epsilon$ (void),

[2] If negative role assertions can not be simulated, as e.g. basic DLs of the $DL\text{-}Lite$ family or in $\mathcal{EL}^{++}$, the syntax of DL-atoms can be accordingly restricted.

(b) a concept instance $C(t_1)$ or its negation $\neg C(t_1)$, with $\boldsymbol{t} = t_1$ a term (i.e., a constant from $\mathcal{F}$ or a variable), or
(c) a role instance $R(t_1, t_2)$ or its negation $\neg R(t_1, t_2)$, with $\boldsymbol{t} = t_1, t_2$ two terms,

where $C, D \in \mathcal{P}_c$, respectively $C, D \in \mathcal{P}_c \cup \{\top, \bot\}$ in case of a), and $R \in \mathcal{P}_r$. Satisfaction of a ground, i.e., variable free, $Q(\boldsymbol{t})$ in a model $\mathcal{I}$ of Φ, is defined by (a) $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ resp. $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$, (b) $t_1^{\mathcal{I}} \in C^{\mathcal{I}}$ resp. $t_1^{\mathcal{I}} \notin C^{\mathcal{I}}$, and (c) $(t_1^{\mathcal{I}}, t_2^{\mathcal{I}}) \in R^{\mathcal{I}}$ resp. $(t_1^{\mathcal{I}}, t_2^{\mathcal{I}}) \notin R^{\mathcal{I}}$. A general $Q(\boldsymbol{t})$ is satisfied by $\mathcal{I}$, if each of its ground instances $Q(\boldsymbol{t'})$ (obtained by replacing variables with constants from $\mathcal{F}$) is satisfied.

Note that entailment $\Phi \models Q(\boldsymbol{t})$ is monotonic, (in particular, $\neg(C \sqsubseteq D)$ is $C \not\sqsubseteq D$).

2.2 DL-Programs

Informally, a DL-program consists of a DL ontology Φ over Σ_o and a normal logic program Π over Σ_p, which may contain DL-queries to Φ in rule bodies. The latter are evaluated subject to hypothetical updates of Φ with assertions determined from the predicate extensions under an interpretation of Π.

Syntax. A signature $\Sigma = \langle \mathcal{F}, \mathcal{P}_o, \mathcal{P}_p \rangle$ for DL-programs consists of a set $\mathcal{F}$ of constant symbols and sets $\mathcal{P}_o, \mathcal{P}_p$ of predicate symbols such that $\Sigma_o = \langle \mathcal{F}, \mathcal{P}_o \rangle$ is a DL-signature and $\Sigma_p = \langle \mathcal{F}, \mathcal{P}_p \rangle = \langle \mathcal{F}, \mathcal{P} \rangle$ is an LP-signature, i.e., a function-free first-order signature over a nonempty finite set $\mathcal{F}$ of constant symbols and a nonempty finite set $\mathcal{P}$ of predicate symbols of arities ≥ 0. Terms over $\mathcal{F}$ and a set $\mathcal{V}$ of variables, and ordinary atoms $p(t_1, \ldots, t_n)$ are defined as usual, where $p \in \mathcal{P}$; a *classical literal* is either an ordinary atom a or its negation $\neg a$.

A *DL-atom* $a(\mathbf{t})$ has the form

$$\mathrm{DL}[S_1\ op_1\ p_1, \ldots, S_m\ op_m\ p_m;\ Q](\boldsymbol{t})\,, \qquad m \geq 0, \tag{1}$$

where 1. either $S_i \in \mathcal{P}_c$ and $p_i \in \mathcal{P}_p$ is unary, or $S_i \in \mathcal{P}_r$ and $p_i \in \mathcal{P}_p$ is binary, 2. $op_i \in \{\uplus, \cup, \cap\}$, and 3. $Q(\boldsymbol{t})$ is a DL-query. We call $\lambda = S_1\ op_1\ p_1, \ldots, S_m\ op_m\ p_m$, the *input signature* and $p_1, \ldots, p_m$ the *input predicates* of $a(\boldsymbol{t})$. We regard λ as unordered list—thus for any permutation π of $\{1, \ldots, n\}$, the DL-atom $\mathrm{DL}[\lambda_\pi;\ Q](\boldsymbol{t})$ where $\lambda_\pi = S_{\pi(1)}\ op_{\pi(1)}\ p_{\pi(1)}, \ldots, S_{\pi(m)}\ op_{\pi(m)}\ p_{\pi(m)}$ is a syntactic variant of (1)—and assume that its elements $S_i\ op_i\ p_i$ are pairwise different. We also write $S_i\ op_i\ p_i \in \lambda$. Intuitively, $op_i = \uplus$ (resp., $op_i = \cup$) increases S_i (resp., $\neg S_i$) by the extension of p_i, while $op_i = \cap$ constrains S_i to p_i.

A *DL-rule* r has the form

$$a_0 \leftarrow a_1, \ldots, a_k, not\ \ a_{k+1}, \ldots, not\ \ a_m\,, \qquad m \geq k \geq 0\,, \tag{2}$$

where a_0 is a classical literal, and every a_i is a classical literal or a DL-atom, $1 \leq i \leq m$, where a_0 may be absent (written as $\bot$). A *DL-program* $P = (\Phi, \Pi)$ consists of a DL ontology Φ and a finite set Π of DL-rules.

Example 2. Consider a DL-program $P = (\Phi, \Pi)$, s.t. $\Phi = \{C \sqsubseteq D\}$ and Π is given by:

$$\{p(a).;\ q(a).;\ r(b).;\ v(X) \leftarrow \mathrm{DL}[C \uplus p, D \cap q;\ D](X), not\ \mathrm{DL}[C \uplus r;\ \neg C](X).\}$$

In the first DL-atom intuitively the concept C is extended by the predicate p and the concept $\neg D$ is restricted by predicate q. Then, all instances of D are retrieved from the resulting ontology. The second DL-atom extends $\neg C$ by the extension of r and queries all instances of $\neg C$ from the respectively extended Φ.

Semantics. In what follows, let $P = (\Phi, \Pi)$ be a DL-program over $\Sigma = \langle \mathcal{F}, \mathcal{P}_o, \mathcal{P}_p \rangle$. By $gr(\Pi)$ we denote the grounding of Π wrt. $\mathcal{F}$, i.e., the set of ground rules originating from DL-rules in Π by replacing, per DL-rule, each variable by each possible combination of constants in $\mathcal{F}$.

An *interpretation* I (over Σ_p) is a consistent set of ground literals over Σ_p; I satisfies (i) a classical ground literal l under Φ, denoted $I \models^{\Phi} l$, iff $l \in I$, and (ii) a ground DL-atom a of the form (1), denoted $I \models^{\Phi} a$, iff $\Phi \cup \tau^I(a) \models Q(\mathbf{c})$, where $\tau^I(a) = \bigcup_{i=1}^{m} A_i(I)$, the DL-update of Φ under I by a, is defined as

- $A_i(I) = \{S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \in I\}$, for $op_i = \uplus$;
- $A_i(I) = \{\neg S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \in I\}$, for $op_i = \cup$;
- $A_i(I) = \{\neg S_i(\mathbf{e}) \mid p_i(\mathbf{e}) \notin I\}$, for $op_i = \cap$. [3]

We say that I *satisfies* a ground DL-rule r of form (2), denoted $I \models^{\Phi} r$, if either $I \models a_0$, $I \models a_j$ for some $k < j \leq m$, or $I \not\models a_i$ for some $1 \leq i \leq k$. I satisfies (is a model of) $P = (\Phi, \Pi)$, denoted $I \models P$, iff $I \models^{\Phi} r$ for all $r \in gr(\Pi)$.

Finally, an interpretation I is an *answer set* of P, iff I is a minimal (wrt. $\subseteq$) model of the FLP-reduct $P^I_{FLP} = \langle \Phi, \Pi^I_{FLP} \rangle$ of P wrt. I, where Π^I_{FLP} contains all ground DL-rules r of form (2) from $gr(\Pi)$ such that $I \models a_i$ for all $1 \leq i \leq k$, and $I \not\models a_j$, for all $k < j \leq m$. This is the ***FLP-semantics*** of DL-programs; several other semantics have been proposed, cf. [4; 13; 19; 17; 3], but the evaluation of DL-atoms is the same.

Example 3. Reconsider P from Example 2. It has one answer set $I = \{p(a), q(a), r(b), v(a)\}$. The DL-update of Φ under I by the DL-atom $\mathrm{DL}[C \uplus p, D \cap q; D](a)$ results in $A_1(I) \cup A_2(I)$, where $A_1(I) = \{C(a)\}$ and $A_2(I) = \{\neg D(b)\}$. Due to $C \sqsubseteq D$ in Φ, it holds that $\Phi \cup A_1(I) \cup A_2(I) \models D(a)$. Thus $\mathrm{DL}[C \uplus p, D \cap q; D](a)$ evaluates to true. On the other hand, the DL-update of Φ under I by $\mathrm{DL}[C \cup r; \neg C](a)$ is $A_3(I) = \{\neg C(b)\}$, and $\Phi \cup A_3(I) \not\models \neg C(a)$. Therefore, $\mathrm{DL}[C \cup r; \neg C](a)$ evaluates to false, and the FLP-reduct of P wrt. I contains the ground rule $v(a) \leftarrow \mathrm{DL}[C \uplus p, D \cap q; D](a), \mathit{not}\ \mathrm{DL}[C \cup r; \neg C](a)$. Finally, one can verify that I is the only answer set of P. Adding the "guessing" rules $v(c) \leftarrow \mathit{not}\ v(b)$ and $v(b) \leftarrow \mathit{not}\ v(c)$ to Π, however, results in two answer sets, namely $I_1 = I \cup \{v(b)\}$ and $I_2 = I \cup \{v(c)\}$.

3 Independent DL-atoms

We call a DL-atom a *independent*, if it always has the same truth value, regardless of the underlying ontology and the context in which it is evaluated, i.e, the interpretation I of the rules. This means that a amounts to one of the logical constants $\bot$ (false, i.e., is a contradiction) or $\top$ (true, i.e., is a tautology).

In formalizing this notion, we take into account that independence trivializes for unsatisfiable underlying ontologies, and thus restrict to satisfiable ones.

[3] If $\neg S_i(\mathbf{e})$ can not be expressed, the use of $\cup$ and $\cap$ is excluded, cf. Footnote 2.

Definition 1 (independent DL-atom). *A ground DL-atom a is* independent*, if for all satisfiable ontologies Φ, Φ' and all interpretations I, I' it holds that $I \models^{\Phi} a$ iff $I' \models^{\Phi'} a$.*

Furthermore, we call a tautologic *(resp.,* contradictory*), if for all satisfiable ontologies Φ and all interpretations I, it holds that $I \models^{\Phi} a$ (resp., $I \not\models^{\Phi} a$).*

Example 4. A DL-atom of the form $a = \mathrm{DL}[; \neg(C \sqsubseteq C)]()$ is contradictory. Indeed, the query $\neg(C \sqsubseteq C)$ is unsatisfiable, hence there does not exist any satisfiable ontology Φ, s.t. $\phi \models \neg(C \sqsubseteq C)$. Hence regardless of I, always $I \not\models^{\Phi} \neg(C \sqsubseteq C)$.

On the other hand consider a DL-atom $b = \mathrm{DL}[C \sqcap p, C \sqcup p; \neg C](c)$. It is tautologic, because under any interpretation I of p, it holds that $\neg C(c) \in \tau^I(b)$. Hence, it is true that $I \models^{\Phi} \neg C(c)$ for any ontology Φ (and any interpretation I).

In the following, we aim at a characterization of independent DL-atoms.

3.1 Contradictory DL-atoms

We defined above contradictory DL-atoms relative to satisfiable ontologies (otherwise, trivially no contradictory DL-atoms exist).

An obvious example of a contradictory DL-atoms is $\mathrm{DL}[; \top \sqsubseteq \bot]()$, where $\bot$ and $\top$ are the customary empty and full concept, respectively. Indeed, the DL-query $\bot \sqsubseteq \top$ is false in every interpretation, i.e., a logical contradiction. As it turns out, contradictory DL-atoms are characterized by such contradictions, and have a simple input signature.

We call a DL-query $Q(\boldsymbol{t})$ *satisfiable*, if there exists some satisfiable ontology L such that $L \models Q(\boldsymbol{t})$, and *unsatisfiable* otherwise. Then we have the following result.

Proposition 1. *A ground DL-atom $a = \mathrm{DL}[\lambda; Q](\boldsymbol{t})$ is contradictory if and only if $\lambda = \epsilon$ and $Q(\boldsymbol{t})$ is unsatisfiable.*

Proof. (if) If $\lambda = \epsilon$, then for every I, $I \models^{\Phi} a$ iff $\Phi \models Q(\boldsymbol{t})$; as $Q(\boldsymbol{t})$ is unsatisfiable, we have for every satisfiable L that $L \not\models Q(\boldsymbol{t})$. Thus a is contradictory.

(Only If). Suppose a is contradictory, i.e., $I \not\models^{\Phi} a$ for every satisfiable ontology Φ and every interpretation I, i.e., $L \cup \tau^I(a) \not\models Q(\boldsymbol{t})$. It follows that $Q(\boldsymbol{t})$ is unsatisfiable. To show $\lambda = \epsilon$, assume towards a contradiction that $\lambda \neq \epsilon$. Then there exists some interpretation I_0 such that $\tau^{I_0}(a) \neq \emptyset$, i.e., contains some assertion B. Consider an arbitrary satisfiable ontology L. As $L \cup \tau^{I_0}(a) \not\models Q(\boldsymbol{t})$, it follows that $L \not\models \neg.B$, where $\neg.B$ is the opposite of B. However, it is not difficult to see that satisfiable ontologies L_0 exist such that $L_0 \models \neg.B$.[4] This, however, raises a contradiction. Thus $\tau = \epsilon$. □

By this result, contradictory DL-atoms have a simple form. As concept and role instance queries are always satisfiable, Q must be a (possibly negated) concept inclusion query and of the form $\neg(C \sqsubseteq C)$, $\neg(C \sqsubseteq \top)$, $\neg(\bot \sqsubseteq C)$, $\neg(\bot \sqsubseteq \top)$, or $\top \sqsubseteq \bot$.

[4] If B is a negative (resp., positive) assertion, then $\neg B$ is a positive (resp. negative) assertion and we can take $L_0 = \{\neg B\}$. If $\neg B$ is not an admissible assertion, we can effect $\neg B$ by a set of possible more restrictive axioms (e.g. we can enforce a negative role assertion $\neg R(a, b)$ in basic *DL-Lite* e.g. by $L_0 = \{\exists R \sqsubseteq C, \exists R \sqsubseteq \neg C\}$ and in $\mathcal{EL}^{++}$ by $L_0 = \{\exists R \sqsubseteq \bot\}$). Note that if negative assertions were not explicitly available in the DL and the operators $\uplus, \sqcap$ disallowed in DL-atoms, still the above construction may be used as e.g. in case of *DL-Lite* and $\mathcal{EL}^{++}$, and thus the same characterization of contradictions holds.

3.2 Tautologic DL-atoms

For tautologic DL-atoms, the situation is more complex. First of all, clearly a DL-atom is tautologic if it has a tautologic query (i.e., it is satisfied by the empty ontology). This is, however, only possible for concept inclusion queries; instance queries $(\neg)C(\boldsymbol{t})$, resp. $(\neg)R(\mathbf{t_1}, \mathbf{t_2})$, are clearly not tautologic.

DL-atoms with tautologic queries are of the form $\mathrm{DL}[\lambda;\ C \sqsubseteq \top](), \mathrm{DL}[\lambda;\ \bot \sqsubseteq C](),$ $\mathrm{DL}[\lambda;\ C \sqsubseteq C]()$, or $\mathrm{DL}[\lambda;\ \top \not\sqsubseteq \bot]()$, where λ is an arbitrary input signature.

However, there are also tautologic DL-atoms whose query is not tautologic.

Example 5. Consider in the fruit scenario the DL-atom

$$a = \mathrm{DL}[EF \sqcap fr, S \uplus fr, S \uplus fr;\ \neg EF](c),$$

where EF stands for exotic fruit, S for sweet, fr for fruit.

Intuitively, we restrict here the concept $\neg EF$ and extend the concepts S and $\neg S$ by the predicate fr. Then we ask whether c is not an exotic fruit. No matter which interpretation I of the DL-program we consider and irrespective of Φ, we will always get that $\Phi \cup \tau^I(a) \models \neg EF(c)$. Indeed, if $fr(c) \in I$, then $\tau^I(a)$ is unsatisfiable; otherwise $\neg EF(c)$ is explicitly present in $\tau^I(a)$. Hence in both cases, $\tau^I(a) \models \neg EF(c)$. This means that a is tautologic.

In the rest of this section, we identify for each query type those forms of the input signature for which the DL-atom is tautologic, or prove nonexistence of such forms. We first consider concept queries, i.e., queries $(\neg)C(\boldsymbol{t})$ and $(\neg)(C \sqsubseteq D)$, and then role queries, for which similar results hold.

Concept queries

Concept instance. To start with, let us consider the query $C(\boldsymbol{t})$. No matter what input signature is considered for this type of the DL-atom, it can never be tautologic.

Proposition 2. *For no input signature λ, a ground DL-atom a of the form* $\mathrm{DL}[\lambda;\ C](\boldsymbol{t})$ *is tautologic.*

Proof. Consider a ground DL-atom $a = \mathrm{DL}[\lambda;\ C](\boldsymbol{t})$. Towards a contradiction, suppose that λ is a signature such that a is tautologic. Thus by definition, for all ontologies Φ and for all interpretations I it holds that $\Phi \cup \tau^I(a) \models C(\boldsymbol{t})$. Thus in particular, for $L = \emptyset$ it holds that $\tau^I(a) \models C(\boldsymbol{t})$. We consider two cases, according to the satisfiability of $\tau^I(a)$. (1) Suppose $\tau^I(a)$ is unsatisfiable. Then there must exist some S, such that $S(\boldsymbol{t}) \in \tau^I(a)$ and $\neg S(\boldsymbol{t}) \in \tau^I(a)$. The presence of $S(\boldsymbol{t})$ in $\tau^I(a)$ can only be ensured if some $S \uplus p$ occurs in the input signature λ of a for some p. Now consider the interpretation $I = \emptyset$. As $p(\boldsymbol{t}) \notin I$, we can not get $S(\boldsymbol{t}) \in \tau^I(a)$, which leads to contradiction.

(2) Now suppose $\tau^I(a)$ is satisfiable. Then $C(\boldsymbol{t})$ must be in $\tau^I(a)$. Similar to the previous case, this requires that $C \uplus p$ occurs in λ for some p. Again $I = \emptyset$ does not allow us to obtain $C(\boldsymbol{t}) \in \tau^I(a)$, hence $\tau^I(a) \not\models C(\boldsymbol{t})$. This contradicts our assumption. □

Concept inclusion. For DL-atoms with concept queries of the form $C \sqsubseteq D$ and $C \not\sqsubseteq D$, where $C \neq D$ and neither concept is $\top$ or $\bot$, we get the same result as for positive instance queries.

Proposition 3. *For no input signature λ, a ground DL-atom of the form* $\mathrm{DL}[\lambda;\ C \sqsubseteq D]()$ *or* $\mathrm{DL}[\lambda;\ C \not\sqsubseteq D]()$*, where $C \neq D$ are different concept names, is tautologic.*

Proof. Consider a ground DL-atom $a = \mathrm{DL}[\lambda;\ C \sqsubseteq D]()$, and suppose a is tautologic. Then for every ontology Φ and interpretation I, it holds that $\Phi \cup \tau^I(a) \models C \sqsubseteq D$. Let $\Phi = \emptyset$ and $I = \emptyset$. Observe that $\tau^I(a)$ is satisfiable, as it contains only negative assertions. Let c be a fresh constant; then $\Phi' = \Phi \cup \tau^I(a) \cup \{C(c), \neg D(c)\}$ is satisfiable, and $\Phi' \not\models C \sqsubseteq D$. By monotonicity of $\models$, it follows $\Phi \cup \tau^I(a) \not\models C \sqsubseteq D$. Thus a is not tautologic, which is a contradiction.

The proof for $a = \mathrm{DL}[\lambda;\ C \not\sqsubseteq D]()$ is similar. □

Out of the remaining concept queries, only the following (straightforwardly) give rise to tautologic DL-atoms.

Proposition 4. *A ground DL-atom of the form* $\mathrm{DL}[\lambda;\ Q]()$ *is a tautology iff $Q = C \sqsubseteq C$, $Q = C \sqsubseteq \top$, or $Q = \top \not\sqsubseteq \bot$, for any $C \in \mathcal{P}_c \cup \{\bot, \top\}$.*

Negative concept instance. Finally, we investigate the forms of tautologic DL-atoms with a query $\neg C(\boldsymbol{t})$.

Proposition 5. *A ground DL-atom a with the query $\neg C(\boldsymbol{t})$ is tautologic if and only if it has one of the following forms:*

c1. $\mathrm{DL}[\lambda,\ C \sqcap p, C \sqcup p;\ \neg C](\boldsymbol{t})$,

c2. $\mathrm{DL}[\lambda,\ C \sqcap p, D \uplus p, D \sqcup p;\ \neg C](\boldsymbol{t})$,

c3. $\mathrm{DL}[\lambda, C \sqcap p_0, C^0 \uplus p_0, C^0 \sqcap p'_0,$
$C^1 \uplus p_1, C^1 \sqcap p'_1, \ldots, C^n \uplus p_n, C^n \sqcap p'_n, C \sqcup p_{n+1};\ \neg C](\boldsymbol{t})$,

c4. $\mathrm{DL}[\lambda, C \sqcap p_0, C^0 \uplus p_0, C^0 \sqcap p'_0,$
$C^1 \uplus p_1, C^1 \sqcap p'_1, \ldots, C^n \uplus p_n, C^n \sqcap p'_n, D \uplus p_{n+1} D \sqcup p'_{n+1};\ \neg C](\boldsymbol{t})$,

where for every $i = 0, \ldots, n+1$, $p_i = p'_j$ for some $j < i$ or $p_i = p_0$, and $p'_{n+1} = p'_{i_j}$ for some $j \leq n$ or $p'_{n+1} = p_0$.

Informally, the lists of $(c3)$ and $(c4)$ include a "chain" $p = p_0 \subseteq p_{j_1} \subseteq p_{j_2} \subseteq p_{j_k} = p_{n+1}$ resp. $p = p_0 \subseteq p_{j'_1} \subseteq p_{j'_2} \subseteq p_{j'_{k'}} = p'_{n+1}$. The proof of this proposition is given in the extended paper [5]; likewise for subsequent results stated without proof.

Example 6 (cont'd). The DL-atom $a = \mathrm{DL}[EF \sqcap fr, S \sqcup fr, S \uplus fr;\ \neg EF](c)$ is an example of the tautologic form (c2). However, the DL-atom in the program of Example 1 is not of any form (c1)–(c4), and thus in general not tautologic.

Role queries A careful analysis reveals that the result for tautologic DL-atoms with concept instance queries carries over to the case when the query $Q(\boldsymbol{t})$ is a role instance query. The same holds for negative concept and role instance queries, when the concept names C, D are replaced with names R_1, R_2 (and the predicates p, q are binary). For the latter consider $a = \mathrm{DL}[\tau; \neg R](\boldsymbol{t})$ that is tautologic. Following the analysis in Proposition 5, which is generic in the arity of the tuple $\boldsymbol{t}$, necessarily the existence of roles R_1 and R_2 instead of C resp. D, and binary instead of unary input predicates p and q can be concluded, For example, the form (c3) above for the role query $\neg R_1$ results in $\mathrm{DL}[\gamma,\ R_1 \sqcap p, R_2 \sqcap q, R_2 \uplus p, R_2 \uplus q;\ \neg R_1](\boldsymbol{t})$, where $R_1,\ R_2$ are roles and $p,\ q$ are binary predicates. More formally, the following is obtained.

Proposition 6. *Propositions 2 and 5 hold if C and D are replaced by role names R_1 and R_2, respectively (and p and q are binary instead of unary).*

Thus, as an interesting consequence, there is no interference of concept and role names in tautologic DL-atoms.

Axiomatization Based on the results above, we obtain a calculus for the derivation of all tautologic DL-atoms as follows. The axioms are:

a0. $\mathrm{DL}[;\ Q]()$,
a1. $\mathrm{DL}[S \sqcap p, S \uplus p;\ \neg S](\boldsymbol{t})$,
a2. $\mathrm{DL}[S \sqcap p, S' \uplus p, S' \uplus p;\ \neg S](\boldsymbol{t})$,

where $Q = S \sqsubseteq S$, $Q = S \sqsubseteq \top$, or $Q = \top \not\sqsubseteq \bot$, S, S' are either distinct concepts or distinct roles, and p is a unary resp. binary predicate.

The first rule of inference is reflecting the monotonicity of DL-atoms wrt. increasing input signatures:

Expansion
$$\frac{\mathrm{DL}[\lambda;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, \lambda';\ Q](\boldsymbol{t})}\ (e).$$

The following rules state that an update predicate p may be replaced by q, if the latter has in case of consistent update $\tau^I(a)$ a larger value than p:

Increase

$$\frac{\mathrm{DL}[\lambda, S \uplus q, S' \uplus p, S' \sqcap q;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \uplus p;\ Q](\boldsymbol{t})}\ (in_1), \qquad \frac{\mathrm{DL}[\lambda, S \sqcup q, S' \uplus p, S' \sqcap q;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \sqcup p;\ Q](\boldsymbol{t})}\ (in_1).$$

Let $\mathcal{K}_{taut}$ denote the respective calculus.

Lemma 1. *Every ground DL-atom a of form $(c1)$ – $(c4)$ is derivable from axioms* **a1** *and* **a2** *using the rules (e), $(in_{\uplus})$, and $(in_{\sqcup})$.*

Since also correctness of the rules is easily establish we have:

Theorem 1. *The calculus $\mathcal{K}_{taut}$ is sound and complete for the theory of tautologic ground DL-atoms.*

Proof. Soundness. It is easily seen that the rules (e), $(in_{\sqcup})$, and $(in_{\uplus})$ are sound. Indeed if a' results from a by rule (e), then $\tau^I(a') \supseteq \tau^I(a)$; if a' results from a by rule $(in_{\uplus})$ resp. $(in_{\sqcup})$, then either $\tau^I(a')$ is unsatisfiable or again $\tau^I(a') \supseteq \tau^I(a)$ (and in case of satisfiability $p^I \subseteq q^I$ must hold).

Completeness. The completeness of the theory follows, as regards concept queries, from Propositions 2–5, and Lemma 1, and as regards roles from Proposition 6. □

Notice that in fact $\mathcal{K}_{taut}$ is minimal, i.e., no axiom scheme or inference rule is redundant.

4 Independence under Inclusion

In the previous section, we considered the existence of contradictory and tautologic DL-atoms in DL-programs in the general case, assuming that the rules of the DL-program are arbitrary. However, by simple analysis or by assertions, we might have information about the relationship between rule predicates that must hold in any model or answer set.

For example, suppose that a DL-program contains the rule

$$q(X) \leftarrow p(X). \tag{3}$$

It imposes an inclusion constraint on the predicates p and q, i.e., for every model I of the program, $p^I \subseteq q^I$ must hold. If p and q are input predicates for DL-atoms, then the rule (3) might affect the independence behavior of a DL-atom in the program: relative to the inclusion constraint, it might be tautologic. Similar rules might state inclusions between binary input predicates, e.g.

$$q(X,Y) \leftarrow p(X,Y), \tag{4}$$

$$q(Y,X) \leftarrow p(X,Y); \tag{5}$$

also projections, e.g.

$$r(X) \leftarrow p(X,Y), \text{ or } r(Y) \leftarrow p(X,Y), \tag{6}$$

(of p on r) might occur. An interesting question is how the presence of such predicate constraints influences the independence behavior, which we address in this section.

We call any rule

$$q(Y_1,\ldots,Y_n) \leftarrow p(X_1,\ldots,X_m) \tag{7}$$

where $n \leq m$ and the Y_i are pairwise distinct variables from $X_1,\ldots,X_m$ an *inclusion constraint (IC)*; if $n = m$, we also write $p \subseteq q$ if $Y_i = X_i$ and $p \subseteq q^-$ if $Y_i = X_{n-i+1}$, for all $i = 1,\ldots,n$. Moreover, for the calculus for tautologic DL-atoms under inclusion constraints as developed in this section, we consider an extended language including p^- as a name, representing for every $p \in \mathcal{P}_o$ its inverse as defined above. By $Cl(\mathcal{C})$ we denote the closure of $\mathcal{C}$, i.e., the set of all ICs which are satisfied by every interpretation I such that $I \models \mathcal{C}$. In particular note that $p \subseteq q^- \models p^- \subseteq q$.

Let us now consider the impact of a set $\mathcal{C}$ on independence. To this end, we consider independence *relative to* $\mathcal{C}$, i.e., the interpretations I, I' in Definition 1 must satisfy $\mathcal{C}$.

Example 7 (cont'd). Reconsider P in Example 1. We can include rule (2) (also written $ex \subseteq vi$) as an inclusion constraint to the set $\mathcal{C}$, and also rule (4). Moreover, as none of the fruits is known to be bitter in our context, we additionally include $ex \subseteq sw$ in $\mathcal{C}$. The closure $Cl(\mathcal{C})$ moreover contains the ICs $vi(X) \leftarrow so(X,Y)$ and $sw \leftarrow so(X,Y)$.

4.1 Contradictory DL-atoms

In what follows, we show that the presence of inclusion constraints $\mathcal{C}$ does not change the result regarding contradictory DL-atoms as obtained for the general case.

Proposition 7. *Let $a = \mathrm{DL}[\lambda;\, Q](t)$ be a ground DL-atom. Then a is contradictory relative to a set $\mathcal{C}$ of inclusion constraints iff $\lambda = \epsilon$ and $Q(t)$ is unsatisfiable.*

Proof. (If) Identical to the if-part of the proof of the Proposition 1.

(Only If) We use the same reasoning as in Proposition 1. If $\lambda \neq \epsilon$ then we can always find an interpretation I_0 such that $\lambda^{I_0}(a) \neq \emptyset$; indeed, we can use $I_0 = \emptyset$, if $\cap$ occurs in λ, and use $I_0 = \mathcal{HB}_{\Pi}$, i.e., the set of all ground atoms, otherwise. □

4.2 Tautologic DL-atoms

Next we investigate how the list of tautologies is modified when inclusion constraints are put on the predicates involved in them.

As we have noted above, the minimal forms of tautologic DL-atoms with concept (resp., role) queries involve only concepts and unary input predicates (resp., roles and binary input predicates).

An inclusion constraint of the form (6) in $\mathcal{C}$ (or the DL-program) will not allow us to get any further tautologic forms. E.g., consider the tautologic DL-atom $\mathrm{DL}[R \cap p, R \uplus p;\, \neg R](\boldsymbol{t})$ we intuitively should get that $\mathrm{DL}[R \cap r, R \uplus p;\, \neg R](\boldsymbol{t})$ is also tautologic. However, this is not a legal DL-atom, as the role R is extended by the unary predicate r.

Dependencies of the form (5) do not allow us to obtain new tautologic DL-atoms either. For example, consider a ground DL-atom $\mathrm{DL}[R \uplus p, R \cap p;\, \neg R](a, b)$, which has the form of axiom **a1**. If we replace the first occurrence of p by q, the resulting DL-atom $\mathrm{DL}[R \uplus q, R \cap p;\, \neg R](a, b)$ is not tautologic. However, for a constraint (4), it is tautologic; it also would be in the former case if the query argument is (a, a).

The following can be shown. For any DL-atom $a = \mathrm{DL}[\lambda;\, Q](\boldsymbol{t})$ and set $\mathcal{C}$ of ICs, let $inp_a(\mathcal{C})$ denote the set of all $q(\boldsymbol{Y}) \leftarrow p(\boldsymbol{X})$ in $\mathcal{C}$ such that p and q occur in λ. We call $\mathcal{C}$ *separable* for a, if every $ic \in inp_a(Cl(\mathcal{C}))$ involves predicates of the same arity.

Proposition 8. *Let $a = \mathrm{DL}[\lambda;\, Q](\boldsymbol{t})$ be a ground DL-atom and $\mathcal{C}$ a separable set of ICs for a. Then a is tautologic relative to $\mathcal{C}$ iff it is tautologic relative to $\mathcal{C}'$ which contains, depending on the type of $Q(\boldsymbol{t})$, the following constraints: (1) $\mathcal{C}' = \emptyset$, in case of a (negated) concept inclusion; (2) every $p \subseteq q$ in $inp_a(Cl(\mathcal{C}))$ where p, q are unary, in case of a (negated) concept instance; (3) every $p \subseteq q$ and $p \subseteq q^-$ in $inp_a(Cl(\mathcal{C}))$ where p, q are binary, in case of a (negated) role instance.*

Proof (Sketch). Every model I of $\mathcal{C}$ is a model of $\mathcal{C}'$. On the other hand, by the form of the ICs, every model I' of $\mathcal{C}'$ can be extended to an interpretation I such that $I \models \mathcal{C}$. In general, the intersection I of all models $I'' \supseteq I'$, which is given by the answer set of $\mathcal{C} \cup I'$, fulfills the claim. Indeed, a fact $a = q(\boldsymbol{c})$ can be in I iff it is provable from I' using a sequence $r_1, r_2, \ldots, r_k$ of rules from $\mathcal{C}$. As all rules are unary, a can be proved from some fact $a' = p(\boldsymbol{c'})$ in $\mathcal{C}$; unfolding the rules, we obtain a rule r of the form

$q(\boldsymbol{Y}) \leftarrow p(\boldsymbol{X})$, where $\boldsymbol{Y} = Y_1, \ldots, Y_m$ are distinct variables from $\boldsymbol{X} = X_1, \ldots, X_n$. As $\mathcal{C} \models r$ and $\mathcal{C}$ is separable for a, it follows that $m = n$ and thus $r \in \mathcal{C}'$, which implies $a' \in I'$. Consequently, I is an extension of I' as claimed. □

That is, for negative role queries we must in general take inverse predicate inclusions into account. To this end, we consider a language including for every $p \in \mathcal{P}_p$ a name p^- for its inverse (as defined in the paper). Such an inverse can be also effected by means of inclusions $q(Y, X) \leftarrow p(X, Y)$ and $p(Y, X) \leftarrow q(X, Y)$ in the set $\mathcal{C}$ of inclusion constraints (where q is then p^- and p is q^-).

Each rule $q(Y_1, Y_2) \leftarrow p(X_1, X_2)$ in $\mathcal{C}$ is then either an inclusion $p \subseteq q$ or an inclusion $p \subseteq q^-$. Note that $p \subseteq q$ iff $p^- \subseteq q^-$ and that for unary predicates, $p^- = p$ and is thus immaterial; furthermore, viewing $\cdot^-$ as an operator, $(p^-)^- = p$. We let $\mathcal{P}_p^{(-)} = \mathcal{P}_p \cup \{p^- \mid p \in \mathcal{P}_p\}$. To see some examples, consider the tautologic form (c1) in Proposition 5. Taking the inclusion constraint $p \subseteq q$ into account, we obtain the following new tautologic form:

- $\mathrm{DL}[\lambda, S \sqcap p, S' \sqcup q;\ \neg S](\boldsymbol{t})$.

The form (c2) yields

- $\mathrm{DL}[\lambda, S \sqcap p, S' \sqcup q, S' \uplus p;\ \neg S](\boldsymbol{t})$, where $p \neq q, S' \neq S$;
- $\mathrm{DL}[\lambda, S \sqcap p, S' \sqcup p, S' \uplus q;\ \neg S](\boldsymbol{t})$, where $p \neq q, S' \neq S$;
- $\mathrm{DL}[\lambda, S \sqcap p, S' \sqcup q, S' \uplus q;\ \neg S](\boldsymbol{t})$, where $p \neq q, S' \neq S$.

From the tautological DL-atom we get

- $\mathrm{DL}[\lambda, S \sqcap p, S' \uplus q, S' \sqcap r, S' \sqcup r;\ \neg S](\boldsymbol{t})$, where $S' \neq S, p \neq r, q \neq r, p \neq q$.

Tautologic forms emanating from (c4) are redundant here, because its modification for the considered case is already included above. For the cases when the DL-query has any of the forms $S(\mathbf{c})$, $C \sqsubseteq D$ or $C \not\sqsubseteq D$, where S is either a concept or a role and C, D are concepts, there are no new tautologies.

4.3 Axiomatization for Tautologies

The results presented above allow us to define rules of inference for deriving tautologies when inclusion constraints are put on the input predicates of a DL-atom.

Inclusion

$$\frac{\mathrm{DL}[\lambda, S \sqcup p;\ Q](\boldsymbol{t}) \quad p \subseteq q}{\mathrm{DL}[\lambda, S \sqcup q;\ Q](\boldsymbol{t})}\ (i_1), \tag{8}$$

$$\frac{\mathrm{DL}[\lambda, S \uplus p;\ Q](\boldsymbol{t}) \quad p \subseteq q}{\mathrm{DL}[\lambda, S \uplus q;\ Q](\boldsymbol{t})}\ (i_2). \tag{9}$$

The "increase" rules are slightly adapted, in comparison to the general case by taking into account that $p \subseteq q$ iff $p^- \subseteq q^-$:

Increase

$$\frac{\mathrm{DL}[\lambda, S \uplus p;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \uplus q, S' \uplus p, S' \sqcap q;\ Q](\boldsymbol{t})}\ (in_{\uplus}), \quad \frac{\mathrm{DL}[\lambda, S \sqcup p;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \sqcup q, S' \uplus p, S' \sqcap q;\ Q](\boldsymbol{t})}\ (in_{\sqcup}),$$

$$\frac{\mathrm{DL}[\lambda, S \uplus p;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \uplus q, S' \uplus p^-, S' \cap q^-;\ Q](\boldsymbol{t})}\ (in^-_{\uplus}), \quad \frac{\mathrm{DL}[\lambda, S \uplus p;\ Q](\boldsymbol{t})}{\mathrm{DL}[\lambda, S \uplus q, S' \uplus p^-, S' \cap q^-;\ Q](\boldsymbol{t})}\ (in^-_{\cup}),$$

where $p, q \in \mathcal{P}_p^{(-)}$ are of the same arity.

We consider the following extended set of axioms compared to the case without inclusion constraints:

a0. $\mathrm{DL}[;\ Q]()$,
a1. $\mathrm{DL}[S \cap p, S \cup p;\ \neg S](\boldsymbol{t})$,
a2. $\mathrm{DL}[S \cap p, S' \uplus q, S' \cup q;\ \neg S](\boldsymbol{t})$, where $q \in \{p, p^-\}$,

and $Q = S \sqsubseteq S$, $Q = S \sqsubseteq \top$, or $Q = \top \not\sqsubseteq \bot$, S, S' are either distinct concepts or distinct roles; moreover, p is a unary or binary predicate.

The described axioms and rules together with the expansion rule defined above, form a calculus for the derivation of tautologic DL-atoms, which we denote by $\mathcal{K}^{\subseteq}_{taut}$. The main result of this section, following next, is its soundness and completeness.

Theorem 2. *The calculus $\mathcal{K}^{\subseteq}_{taut}$ is sound and complete for tautologic ground DL-atoms a relative to any closed set of inclusion constraints $\mathcal{C}$ (i.e., such that $\mathcal{C} = Cl(\mathcal{C})$) that is separable for a.*

We use our running example to illustrate the application of $\mathcal{K}^{\subseteq}_{taut}$.

Example 8 (cont'd). Reconsider the DL-program in Example 1, and recall that no ground instance of its DL-atom, in particular

$$a = \mathrm{DL}[H \uplus vi, H \cup sw, A \cap ex;\ \neg A](pineapple)$$

is tautologic. Now let us take the predicate constraints in P into account. Recall that essentially by the rules (2) and (3), we have that $\{ex \subseteq vi, ex \subseteq sw\} \subseteq Cl(\mathcal{C})$ (which is also separable for a). We thus can derive a in $\mathcal{K}^{\subseteq}_{taut}$ given $\mathcal{C}$ as follows:

$$\dfrac{\dfrac{\dfrac{\mathrm{DL}[H \uplus ex, H \cup ex, A \cap ex;\ \neg A](pineapple)}{\mathrm{DL}[H \uplus ex, H \cup ex, A \cap ex;\ \neg A](pineapple)}\ ex \subseteq vi\ (i_2)}{\mathrm{DL}[H \uplus vi, H \cup ex, A \cap ex;\ \neg A](pineapple)}\ ex \subseteq sw\ (i_1)}{\mathrm{DL}[H \uplus vi, H \cup sw, A \cap ex;\ \neg A](pineapple)}$$

The leaf of the proof tree is a DL-atom $\mathrm{DL}[H \uplus ex, H \cup ex, A \cap ex;\ \neg A](pineapple)$. It has the form of axiom **a2**. Hence the initial DL-atom a is, by virtue of Theorem 2, tautologic relative to $\mathcal{C}$.

The results of this section can be readily used for optimization or reasoning tasks on DL-programs that involve ground DL-atoms, e.g. in diagnosis and repair [16; 8]. They can moreover be exploited for dealing with non-ground DL-atoms. We may call a such a DL-atom $a = \mathrm{DL}[\lambda;\ Q](\boldsymbol{t})$ independent (resp. contradictory, tautologic), if each of its ground instances has this property. From the results above, we obtain that there are no contradictory nonground DL-atoms, and that to prove a tautologic, it is sufficient to consider a single instance a (particular constants do not matter, and for role queries $(\neg)R(t_1, t_2)$, consider different constants if possible).

Example 9. In our running example, e.g., the instance of a for $X = pineapple$ is tautologic relative to the constraints; hence a is tautologic and can be removed from rule (5).

5 Complexity

Let us now consider the complexity of determining whether a DL-atom a is independent. To determine whether a is contradictory is trivial, given the simple forms of unsatisfiable DL-queries. For determining whether a is tautologic, we can use the calculus $\mathcal{K}^{\subseteq}_{taut}$ established above, and aim at a derivation of a. In the search, we need an oracle for deciding whether $ic \in Cl(\mathcal{C})$, for a given IC ic and $\mathcal{C}$, to see whether a rule is applicable.

The complexity of this oracle is in fact the dominating factor for the search. Indeed, the inclusion rules of $\mathcal{K}^{\subseteq}_{taut}$ work strictly local, in the sense that they only replace one occurrence of an input predicate by another one, and few independent rule applications are needed to arrive at an axiom (see below).

The complexity of deciding, given an IC ic and a set $\mathcal{C}$ of ICs, whether $ic \in Cl(\mathcal{C})$, depends on the form of the ICs. In general, the problem is decidable in polynomial space, and it is NLogSpace-complete if the arities of the predicates in $\mathcal{C}$ are bounded by a constant k. In particular, for $k = 2$ deciding $ic \in Cl(\mathcal{C})$ if all predicates in ic have the same arity, is possible using the following inference rules:

$$\frac{X \subseteq Y \quad Y \subseteq Z}{X \subseteq Z} \qquad \frac{X \subseteq Y}{X^- \subseteq Y^-} \qquad \frac{X^- \subseteq Y^-}{X \subseteq Y} \tag{10}$$

where X, Y, Z are meta variables which denote unary (binary) predicates. On the other hand, the problem is NLogSpace-hard for every $k \geq 1$ as it subsumes graph reachability.

We have the following result.

Theorem 3. *Given a DL-atom a and a separable set $\mathcal{C}$ of ICs for a, deciding whether a is tautologic relative to $\mathcal{C}$ is (i)* NLogSpace*-complete and* NLogSpace*-hard even if $\mathcal{C} = \emptyset$, and is (ii) in* LogSpace*, and in fact expressible by a fixed first-order formula (hence in* AC^0*), if the DL query Q of a is not a negative concept resp. role query.*

Proof (Sketch). By the above results on $\mathcal{K}^{\subseteq}_{taut}$, we need an oracle for $ic \in Cl(\mathcal{C})$, where ic involves only unary resp. binary predicates. Due to the special form of ICs, $ic \in Cl(\mathcal{C})$ iff $ic \in Cl([\mathcal{C}]_2)$, where $[\mathcal{C}]_2$ is the set of all ICs in $\mathcal{C}$ that involve only unary and/or binary predicates. Thus, by the observation above, an NLogSpace oracle is sufficient.

To prove that $a = \mathrm{DL}[\lambda; Q](\boldsymbol{t})$ is tautologic, we can guess an instance of an axiom $\mathbf{ai}$ from which we want to arrive at a by application of rules in $\mathcal{K}^{\subseteq}_{taut}$. Checking that $Q(\boldsymbol{t})$ matches the query of $\mathbf{ai}$ is easy, and we can check in case of **a1**, **a2** that $S \uplus p$ occurs in λ; we then can check whether $S \uplus p$ resp. $S' \uplus p^{(-)}$, $S' \uplus p^{(-)}$ occur in λ, and if not, in case of **a1** build nondeterministically a "chain" $q_0 (= p) \subseteq q_1 \subseteq \cdots \subseteq q_k$ such that $S' \uplus q_k \in \lambda$ and in case of **a2** also a "chain" $r_0 (= p) \subseteq r_1 \subseteq \cdots \subseteq r_{k'}$ such that $S' \uplus q_{k'} \in \lambda$, where for every q_i, we have that either $q_{i-1} \subseteq q_i$ (which can be checked with the oracle), or some pair $S'' \uplus q_{i-1}^{(-)}, S'' \uplus q_i^{(-)}$ occurs in λ and similarly, for every r_j we have that either $r_{j-1} \subseteq r_j$ (an oracle check), or some pair $S'' \uplus r_{j-1}^{(-)}, S'' \uplus r_j^{(-)}$ occurs in λ; building a chain stops as soon as $S' \uplus q_i \in \lambda$ resp. $S' \uplus r_i \in \lambda$ is found (it may else stop after a certain number of steps, but this is irrelevant here).

A simple analysis reveals that this overall algorithm is feasible, relative to the oracle, in logarithmic space (one can cycle through the few guesses with constantly many

variables, and building chains as above is feasible in nondeterministic logarithmic space, as we just need to memorize q_i, p_0, p_{n+1} resp. p'_{n+1}, and S'). It follows that in general, the problem is in NLogSpace.

The problem is shown to be NLogSpace-hard via a reduction from the canonical graph reachability problem. Let $G = (V, E)$ be a directed graph and let $s, t \in V$ be nodes. We view each node $v \in V$ as a unary predicate, and define the DL-atom $a = \mathrm{DL}[C \sqcap s, \lambda, C \sqcup t; \neg C](a)$ where λ contains for each edge $(v, w) \in E$ the elements $C^{(v,w)} \uplus v, C^{(v,w)} \sqcap w$, where $C^{(v,w)}$ does not occur elsewhere. Then it holds that a is tautologic (wrt. $\mathcal{C} = \emptyset$) iff t is reachable from s in G. Indeed, note that by its form, a must be derived from an instance $\mathrm{DL}[C \sqcap s, C \sqcup t; \neg C](a)$ of **a1**, and that for this a chain $q_0 = s \subseteq q_1 \subseteq \cdots \subseteq q_k = t$ must be built to obtain $C \sqcup t$, and only the rule $(in_{\sqcup})$ is applicable. This chain corresponds to a path in G from s to t. Conversely from any path $s = v_0, v_1, \ldots v_k = t$ in G, we can build a corresponding chain with elements $C^{(v,w)} \uplus v, C^{(v,w)} \sqcap w$ in λ using the rule $(in_{\sqcup})$.

Finally, if Q is not a negative concept resp. role query, then for a to be tautologic it must be an instance of **a0**, which is checkable in logarithmic space and also expressible by a FOL formula ϕ over a relational structure (roughly, a plain SQL query over a database) that stores in suitable relations: all triples $S_i\ op_i\ p_i$ in λ, using S_i, op_i, and p_i as constants; the query $Q(\boldsymbol{t})$; and all inclusions $p \subseteq q^{(-)}$ from $Cl(\mathcal{C})$. In fact, ϕ can be fixed, and the relations are easily assembled from a and $\mathcal{C}$. As evaluating a fixed FOL formula over relational structures is in AC^0, we obtain the result. □

6 Conclusion and Future Work

To the best of our knowledge, the notion of independent DL-atom has not been considered before, which is of use in optimization and for reasoning tasks on DL-programs. We investigated the forms of tautologic and contradictory ground DL-atoms in the general case, as well as in the case when inclusion constraints on the input predicates are known. We showed that contradictory DL-atoms have a simple form, and we presented a sound and complete calculus for determining tautologic DL-atoms. Based on it, we determined the complexity of deciding this problem, and showed that the problem is very efficiently solvable in general, as well as relative to the predicate constraints. Furthermore, the results for ground DL-atoms can be easily lifted to deal with nonground DL-atoms, and an implementation of the calculus using logic programming is rather straightforward.

Outlook. Several issues remain for further investigation. A possible extension is to consider DL queries which allow for non-atomic concepts, respectively roles. Some of our results can be readily extended to such queries (e.g., to conjunctive concept/role queries), but to get a clear picture further work is needed.

As an alternative, or in addition to ICs, further information about the DL-program might be available relative to which independence of a DL-atom can be established.

Regarding predicate constraints, one issue is non-separable sets of inclusion constraints., i.e., to permit projections among input predicates of DL-atoms, for which the presented calculus is sound but not complete. One can also imagine more general inclusion constraints, by relaxing the conditions to allow e.g. repetition of arguments, or inclusion of intersections. Other possibilities are to consider exclusion constraints, or

(non-)emptiness constraints on predicates. Adopting a technical view, we could consider arbitrary sets of constraints that describe an envelope of the set of answer sets of the underlying DL-program. The study of different forms of constraints remains to be done.

Orthogonal to rules, one may exploit information about the ontology. So far, information in the ontology Φ about the concepts (roles) that occur in the DL-atoms considered has been ignored. However, such information may lead to further independent DL-atoms. For example, knowing that $\Phi \models C \sqsubseteq D$ and that $\mathrm{DL}[\lambda, C \uplus p;\ Q](\boldsymbol{t})$ is tautologic, we can infer that $\mathrm{DL}[\lambda, D \uplus p;\ Q](\boldsymbol{t})$ is also tautologic. Incorporating such information and other information into the calculus remains for future work.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. CUP (2003)
2. Dao-Tran, M., Eiter, T., Krennwallner, T.: Realizing default logic over description logic knowledge bases. In: Proc. ECSQARU'09. LNCS 5590, pp. 602–613. Springer (2009)
3. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R.: Well-founded semantics for description logic programs in the Semantic Web. ACM Trans. Comput. Log. 12(2), 11 (2011)
4. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the Semantic Web. AIJ 172, 1495-1539 (2008)
5. Eiter, T., Fink, M., Stepanova, D.: Semantic independence in DL-programs. Tech. Rep. INFSYS RR-1843-12-07 (2012)
6. Eiter, T., Ianni, G., Krennwallner, T., Schindlauer, R.: Exploiting conjunctive queries in description logic programs. Ann. Math. Artif. Intell. 53(1-4), 115–152 (2008)
7. Eiter, T., Ianni, G., Schindlauer, R., Tompits, H.: Nonmonotonic description logic programs: Implementation and experiments. In: Proc. LPAR'04. LNCS 3452, pp. 511–527. (2004)
8. Fink, M., El Ghali, A., Chniti, A., Korf, R., Schwichtenberg, A., Lévy, F., Pührer, J., Eiter, T.: D2.6 Consistency maintenance. Tech. Rep. 2.6, ONTORULE ICT-2009-231875 Project (2011), http://ontorule-project.eu/outcomes?func=fileinfo&id=92
9. Fink, M., Pearce, D.: A logical semantics for description logic programs. In: JELIA'10. LNCS 6341, pp. 156–168. Springer (2010)
10. Heymans, S., Korf, R., Erdmann, M., Pührer, J., Eiter, T.: Loosely coupling F-logic rules and ontologies. In: Int'l Conf. on Web Intelligence (WI 2010), pp. 248-255. IEEE CS (2010)
11. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. Journal of Web Semantics 1(4), 345–357 (2004)
12. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a Web ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
13. Lukasiewicz, T.: A novel combination of answer set programming with description logics for the semantic web. IEEE Trans. Knowl. Data Eng. 22(11), 1577–1592 (2010)
14. Motik, B., Patel-Schneider, P.F., Parsia, B. (eds.): OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (2008), w3C Working Draft April 2009
15. Motik, B., Rosati, R.: Reconciling Description Logics and Rules. JACM 57(5), 1–62 (2010)
16. Pührer, J., Heymans, S., Eiter, T.: Dealing with inconsistency when combining ontologies and rules using dl-programs. In: Proc. ESWC'10 (1). LNCS 6088, pp. 183–197. Springer (2010)
17. Shen, Y.D.: Well-supported semantics for description logic programs. In: Proc. IJCAI'11. pp. 1081–1086. AAAI Press (2011)
18. Wang, K., Billington, D., Blee, J., Antoniou, G.: Combining description logic and defeasible logic for the semantic web. In: RuleML'04. LNCS 3323, pp. 170–181. Springer (2004)
19. Wang, Y., You, J.H., Yuan, L.Y., Shen, Y.D.: Loop formulas for description logic programs. Theory and Practice of Logic Progamming 10(4-6), 531–545 (2010)