Extending Consequence-Based Reasoning to $\mathcal{SHIQ}$

Andrew Bate, Boris Motik, Bernardo Cuenca Grau,
František Simančík, and Ian Horrocks

University of Oxford
Oxford, United Kingdom
first.last@cs.ox.ac.uk

1 Introduction

Description logics (DLs) [3] are a family of knowledge representation formalisms with numerous practical applications. $\mathcal{SHIQ}$ is a particularly important DL as it provides the formal underpinning for the Web Ontology Language (OWL). DLs model a domain of interest using *concepts* (i.e., unary predicate symbols) and *roles* (i.e., binary predicate symbols). DL applications often rely on *subsumption*—the problem of checking logical entailment between concepts—and so the development of practical subsumption procedures for DLs such as $\mathcal{SHIQ}$ has received a lot of attention.

Most DLs are fragments of the *guarded fragment* [1] of first-order logic; however, $\mathcal{SHIQ}$ provides a restricted form of counting that does not fall within the guarded fragment. Moreover, most DLs, including $\mathcal{SHIQ}$, can be captured using the *two-variable fragment of first-order logic with counting* ($\mathcal{C}^2$) [10], but this provides us with neither a practical nor a worst-case optimal reasoning procedure ($\mathcal{C}^2$ and $\mathcal{SHIQ}$ are NExpTime- and ExpTime-complete, respectively). Algorithms for more general logics thus do not satisfy the requirements of DL applications, and so numerous alternatives specific to DLs have been explored. Many DLs can be decided in the framework of resolution [17, 12], including $\mathcal{SHIQ}$ [13]. These procedures are usually worst-case optimal and can be practical, but, as we discuss in Section 3, in even very simple cases they can draw unnecessary inferences. Practically successful $\mathcal{SHIQ}$ reasoners, such as FaCT++ [25], HermiT [8], Pellet [24], and Racer [11], use variants of highly-optimised (hyper)tableau algorithms [5]—model-building algorithms that ensure termination by a variant of *blocking* [6]. Although worst-case optimal tableau algorithms are known [9], practical implementations are typically not worst-case optimal. While generally very effective, tableau algorithms still cannot process certain ontologies; for example, the GALEN ontology[1] has proved particularly challenging, mainly because tableau calculi tend to construct very large models.

A breakthrough in practical ontology reasoning came in the form of *consequence-based calculi*. Although not originally presented in the consequence-based framework, the algorithm for the DL $\mathcal{EL}$ [2] can be seen as the first such calculus. This algorithm was later reformulated and extended to Horn-$\mathcal{SHIQ}$ [14] and Horn-$\mathcal{SROIQ}$ [18]—DLs that support functional roles, but not disjunctive reasoning. Recently, consequence-based calculi were also developed for the DLs $\mathcal{ALCH}$ [23] and $\mathcal{ALCI}$ [22], which support disjunctive reasoning, but not counting. Consequence-based calculi can be seen as

[1] http://www.opengalen.org

a combination of resolution and hypertableau (see Section 3 for details). As in resolution, they describe ontology models by systematically deriving certain ontology consequences; and as in hypertableau, the ontology axioms can be used to guide the derivation process, and to avoid drawing unnecessary inferences. Moreover, consequence-based calculi are not just refutationally complete, but can classify an ontology in a single pass, which greatly reduces the overall work. These advantages allowed the CB system to be the first to classify all of GALEN [14].

Existing consequence-based algorithms can handle either disjunctions or counting, but not both. As we discuss in detail in Section 4, it is challenging to extend these algorithms to DLs such as $\mathcal{SHIQ}$ that combine both kinds of construct: counting quantifiers require equality reasoning, which together with disjunctions can impose complex constraints on ontology models. Unlike in existing consequence-based calculi, these constraints cannot be captured using DLs themselves; instead, a more expressive first-order fragment is needed, which makes the reasoning process much more involved.

In Section 5 we present a consequence-based calculus for $\mathcal{SHIQ}$. Borrowing ideas from resolution theorem proving, we encode the required consequences using a special kind of first-order clause; and to handle equality effectively, we base our calculus on *ordered paramodulation* [16]—a state of the art calculus for equational theorem proving used in modern systems such as E [21] and Vampire [19]. To make the calculus efficient on $\mathcal{EL}$, we have carefully constrained the rules so that, on $\mathcal{EL}$ ontologies, it mimics existing $\mathcal{EL}$ calculi. Thus, although a practical evaluation of our calculus is still pending, we believe that it is likely to perform well in practice on 'mostly-$\mathcal{EL}$' ontologies due to is close relationship with existing and well-proven calculi.

2 Preliminaries

First-Order Logic. To simplify matters technically, it is common practice in equational theorem proving to encode atoms as terms. An atomic formula $P(\vec{s})$ can be encoded as $P(\vec{s}) \approx \mathtt{t}$, where $\mathtt{t}$ is a new special constant, and P is considered as a function symbol rather than as a predicate symbol. Note however that, in order to avoid meaningless expressions in which predicate symbols occur at proper subterms, a multi-sorted type discipline on terms in the encoding is adopted. Thus, the set of symbols in the signature is partitioned into a set $\mathcal{P}$ of *predicate symbols* (which includes $\mathtt{t}$), and a set $\mathcal{F}$ of *function symbols*.

A *term* is constructed as usual using variables and the signature symbols. Terms containing predicate symbols as their outermost symbol are called $\mathcal{P}$-*terms*, while all other terms are $\mathcal{F}$-*terms*. For example, for P a predicate and f a function symbol, both $f(P(x))$ and $P(P(x))$ are malformed; $P(f(x))$ is a well-formed $\mathcal{P}$-term; and $f(x)$ is a well-formed $\mathcal{F}$-term. An *(in)equality* is an expression of the form $s \approx t$ ($s \not\approx t$) where s and t are both either $\mathcal{F}$- or $\mathcal{P}$-terms. We assume that $\approx$ and $\not\approx$ are implicitly symmetric—that is, $s \bowtie t$ and $t \bowtie s$ are one and the same expression, for $\bowtie \in \{\approx, \not\approx\}$. A *literal* is an equality or an inequality. An *atom* is an equality of the form $P(\vec{s}) \approx \mathtt{t}$, and we write it simply as $P(\vec{s})$ whenever it is clear from the context whether $P(\vec{s})$ is intended to be a $\mathcal{P}$-term or an atom. A *clause* is an expression of the form $\Gamma \to \Delta$ where Γ is a conjunction of atoms called the *body*, and Δ is a disjunction of literals called the

Table 1. Translating Normalised $\mathcal{SHIQ}$ Ontologies into DL-Clauses

$B_1 \sqsubseteq \geqslant n\, S.B_2$	$\rightsquigarrow$	$B_1(x) \rightarrow S(x, f_i(x))$ for $1 \leq i \leq n$ $B_1(x) \rightarrow B_2(f_i(x))$ for $1 \leq i \leq n$ $B_1(x) \rightarrow f_i(x) \not\approx f_j(x)$ for $1 \leq i < j \leq n$
$B_1 \sqsubseteq \leqslant n\, S.B_2$	$\rightsquigarrow$	$S(y,x) \wedge B_2(x) \rightarrow S_{B_2}(y,x)$ for fresh S_{B_2} $B_1(x) \wedge \bigwedge_{1 \leq i \leq n+1} S_{B_2}(x, z_i) \rightarrow \bigvee_{1 \leq i < j \leq n+1} z_i \approx z_j$
$B_1 \sqsubseteq \forall S.B_2$	$\rightsquigarrow$	$B_1(x) \wedge S(x, z_1) \rightarrow B_2(z_1)$
$\bigsqcap_{1 \leq i \leq n} B_i \sqsubseteq \bigsqcup_{1 \leq j \leq m} B_j$	$\rightsquigarrow$	$\bigwedge_{1 \leq i \leq n} B_i(x) \rightarrow \bigvee_{1 \leq i \leq m} B_j(x)$
$S_1 \sqsubseteq S_2$	$\rightsquigarrow$	$S_1(x, z_1) \rightarrow S_2(x, z_1)$
$S_1 \sqsubseteq S_2^-$	$\rightsquigarrow$	$S_1(x, z_1) \rightarrow S_2(z_1, x)$

head. We often treat conjunctions and disjunctions as sets; and we write the empty conjunction (disjunction) as $\top$ ($\bot$). We use the standard notion of subterm positions; then, $s|_p$ is the subterm of s at position p; moreover, $s[t]_p$ is the term obtained from s by replacing the subterm at position p with t; finally, position p is *proper* in t if $t|_p \neq t$.

Orders. A *term order* $\succ$ is a *strict* order on the set of all terms. The *multiset extension* $\succ_{mul}$ of $\succ$ compares multisets M and N on a universe U such that $M \succ_{mul} N$ if and only if $M \neq N$ and, for each $n \in N \setminus M$, some $m \in M \setminus N$ exists such that $m \succ n$, where $\setminus$ is the multiset difference. We extend $\succ$ to literals by identifying each $s \not\approx t$ with the multiset $\{s, s, t, t\}$ and each $s \approx t$ with the multiset $\{s, t\}$, and by comparing the result using $\succ_{mul}$. Given an order $\succ$, element $b \in U$, and subset $S \subset U$, the notation $S \succ b$ abbreviates $\exists a \in S : a \succ b$.

Description Logic $\mathcal{SHIQ}$. In this paper, a $\mathcal{SHIQ}$ ontology is represented as a set of *DL-clauses*, which we define next. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be countable sets of unary and binary predicate symbols, and let $\mathcal{F}$ be a countable set of unary function symbols. DL-clauses are constructed using the *central variable* x and variables z_i. A *DL-$\mathcal{F}$-term* has the form x, z_i, or $f(x)$ with $f \in \mathcal{F}$; a *DL-$\mathcal{P}$-term* has the form $B(z_i)$, $B(x)$, $B(f(x))$, $S(x, z_i)$, $S(z_i, x)$, $S(x, f(x))$, $S(f(x), x)$ with $B \in \mathcal{P}_1$ and $S \in \mathcal{P}_2$; and a *DL-term* is a DL-$\mathcal{F}$- or a DL-$\mathcal{P}$-term. A *DL-literal* has the form $A \approx \mathsf{t}$ with A a DL-$\mathcal{P}$-term (called a *DL-atom*), or $f(x) \bowtie g(x)$, $f(x) \bowtie z_i$, or $z_i \bowtie z_j$ with $\bowtie \in \{\approx, \not\approx\}$. A *DL-clause* contains only DL-atoms of the form $B(x)$, $S(x, z_i)$, and $S(z_i, x)$ in the body and only DL-literals in the head, and where each variable z_i occurring in the head also occurs in the body. An *ontology* $\mathcal{O}$ is a finite set of DL-clauses. A *query clause* is a DL-clause in which all atoms are of the form $B(x)$. Given an ontology $\mathcal{O}$ and a query clause $\Gamma \rightarrow \Delta$, the *query clause entailment* problem is to decide whether $\mathcal{O} \models \forall x.(\Gamma \rightarrow \Delta)$ holds; we often leave out $\forall x$ and write the latter as $\mathcal{O} \models \Gamma \rightarrow \Delta$.

$\mathcal{SHIQ}$ ontologies are commonly written using a DL-style syntax, but we can always transform such ontologies into DL-clauses without affecting the entailment of query clauses. Transitivity is encoded away as described in [20, 7], and the resulting axioms are *normalised* to the forms shown on the left-hand side of Table 1 as described in [14, 22]. The normalised axioms are translated to DL-clauses as shown in Table 1.

$$
\begin{aligned}
\mathcal{O}_1 = \{ \quad & B_i \sqsubseteq \exists S_j.B_{i+1} && \text{for } 0 \leq i < n \text{ and } 1 \leq j \leq 2 && (1)\\
& B_n \sqsubseteq C_n && && (2)\\
& C_{i+1} \sqsubseteq \forall S_j^-.C_i && \text{for } 0 \leq i < n \text{ and } 1 \leq j \leq 2 && (3) \ \}
\end{aligned}
$$

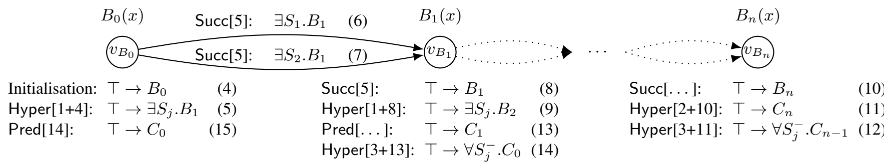

Fig. 1. Example Motivating Consequence-Based Calculi

3 Why Consequence-Based Calculi?

Consider the ontology $\mathcal{O}_1$ (written using DL notation) shown in Figure 1. Axiom (3) can be reformulated as $\exists S_j.C_{i+1} \sqsubseteq C_i$, and so $\mathcal{O}_1$ is in $\mathcal{EL}$. One can readily verify that $\mathcal{O} \models B_i \sqsubseteq C_i$ holds for each $1 \leq i \leq n$.

To prove, say, $\mathcal{O} \models B_0 \sqsubseteq C_0$, a (hyper)tableau calculus constructs in a forward-chaining manner a tree-shaped model of depth n and of fanout two, where nodes at depth i are labelled by B_i and C_i. Forward chaining ensures that reasoning is goal-oriented and thus amenable to practical implementation. However, all nodes labelled with B_i are of the same type and behave in the same way, which reveals a weakness of (hyper)tableau calculi: the constructed model can be large (exponential in our example) and highly redundant. Techniques such as *caching* [9] or *anywhere blocking* [15] can be used to constrain model construction, but their effectiveness often depends on the order of rule applications. Thus, model size has proved to be a key limiting factor for (hyper)tableau-based reasoners in practice [15].

In contrast, resolution describes models using universally quantified clauses that 'summarise' the model. This prevents redundancy and ensures worst-case optimality of many resolution decision procedures. Nevertheless, resolution can still derive unnecessary clauses. In our example, axioms (1) and (3) are translated into clauses (16) and (17), respectively, which can be used to derive all $2n^2$ clauses of the form (18).

$$
\begin{aligned}
B_i(x) \to S_j(x, f_{i,j}(x)) \quad & \text{for } i \in \{1, \ldots, n\} \text{ and } j \in \{1, 2\} && (16)\\
S_j(z_1, x) \wedge C_{k+1}(x) \to C_k(z_1) \quad & \text{for } k \in \{1, \ldots, n\} \text{ and } j \in \{1, 2\} && (17)\\
B_i(x) \wedge C_{k+1}(f_{i,j}(x)) \to C_k(x) \quad & \text{for } i, k \in \{1, \ldots, n\} \text{ and } j \in \{1, 2\} && (18)
\end{aligned}
$$

Of these $2n^2$ clauses, only those where $i = k$ are relevant to proving $\mathcal{O} \models B_0 \sqsubseteq C_0$. Moreover, if we extend $\mathcal{O}$ with additional clauses that contain B_i and C_i, each of the $2n^2$ clauses from (18) can participate in further inferences, which can give rise to many more irrelevant conclusions. This problem is exacerbated in satisfiable cases since all resolution consequences must then be computed in full.

Consequence-based calculi combine the goal-directed reasoning of (hyper)tableau calculi with the 'summarisation' of resolution. In [22], we presented a very general framework for $\mathcal{ALCI}$ ontologies that captures the key elements of consequence-based calculi such as [2, 14, 18, 23]. We use this framework as basis for our extension to

$\mathcal{SHIQ}$ so, before presenting our extension, we explain the main concepts on $\mathcal{O}_1$. Due to space restrictions we cannot reproduce in full the inference rules from [22]; however, these are similar in spirit to our inference rules for $\mathcal{SHIQ}$ presented in Table 2.

Our calculus constructs a *context structure* $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$—a graph whose vertices $\mathcal{V}$ are called *contexts* and whose directed edges are labelled with concepts of the form $\exists S.B$. Let I be a model of $\mathcal{O}$. Instead of representing each element of I individually as in (hyper)tableau calculi, we 'summarise' all elements of a certain kind using a single context v. Each context $v \in \mathcal{V}$ is associated with a (possibly empty) set core_v of *core* concepts that hold in all domain elements that v represents; thus, core_v determines the kind of context v. We use a set $\mathcal{S}_v$ of clauses to capture additional constraints that the elements represented by v must satisfy; in $\mathcal{ALCI}$, we can do so using clauses over DL concepts of the form $\bigsqcap B_i \sqsubseteq \bigsqcup B_j \sqcup \bigsqcup \exists S_k.B_k \sqcup \bigsqcup \forall S_\ell.B_\ell$. Thus, unlike in resolution where all consequences belong to a single set, we assign a consequence a particular set in order to reduce the number of inferences. Clauses in $\mathcal{S}_v$ are 'relative' to core_v: for each $\Gamma \sqsubseteq \Delta \in \mathcal{S}_v$, we have $\mathcal{O} \models \mathsf{core}_v \sqcap \Gamma \sqsubseteq \Delta$—that is, we choose not to include core_v in clause bodies since core_v always holds. Finally, $\succ$ provides each context $v \in \mathcal{V}$ with a concept order $\succ_v$ that restricts resolution inferences in the presence of disjunctions.

Consequence-based calculi are not just refutation-complete: they actually derive the required consequences. Figure 1 shows how this is achieved for $\mathcal{O}_1 \models B_0 \sqsubseteq C_0$; the core and the clauses are shown, respectively, above and below a context. To prove $B_0 \sqsubseteq C_0$, we introduce context v_{B_0} with $\mathsf{core}_{v_{B_0}} = \{B_0\}$ and clause (4) stating that B_0 holds in this context. Next, using the Hyper rule, we derive (5) from (1) and (4); this rule performs hyperresolution, but restricted to one context at a time.

Next, the Succ rule satisfies the existential quantifiers in (5). To this end, the rule uses a parameter called an *expansion strategy*. A strategy is given two sets of constraints that a successor of v_{B_0} must satisfy due to universal restrictions: K_1 contains constraints that must hold, and K_2 contains constraints that might hold. Given such K_1 and K_2, the strategy then decides whether to reuse an existing context or create a fresh one, and in the latter case it also determines how to initialise the new context's core. In our example, there are no universal restrictions and all information in v_{B_0} is deterministic, so $K_1 = K_2 = \{B_1\}$. For $\mathcal{EL}$, a reasonable strategy is to associate with each concept B_i a context v_{B_i} with $\mathsf{core}_{v_{B_i}} = \{B_i\}$, and to always to satisfy existential quantifiers of the form $\exists S.B_i$ using v_{B_i}; thus, in our example we introduce v_{B_1} and initialise it with (8). Note that (5) represents two existential quantifiers, both of which we satisfy (in separate applications of the Succ rule) using v_{B_1}. Different strategies may be used with more expressive DLs; please refer to [22, Section 3.4] for an in-depth discussion.

We construct contexts $v_{B_2}, \ldots, v_{B_n}$ in a similar way, finally deriving (11) by hyperresolving (2) and (10), and then (12) by hyperresolving (3) and (11). Clause (12) imposes a constraint on the predecessor context, which we propagate backwards using the Pred rule, obtaining (13) and (15). Since, however, clauses in $\mathcal{S}_{v_{B_0}}$ are 'relative' to $\mathsf{core}_{v_{B_0}}$, clause (15) actually represents our query clause $B_0 \sqsubseteq C_0$.

Thus, like resolution, consequence-based calculi 'summarise' models to prevent redundant computation, and, like (hyper)tableau calculi, they differentiate elements in a model of $\mathcal{O}$ to prevent the derivation of consequences such as (18).

$$\mathcal{O}_2 = \{ \quad B_0(x) \to S(f_1(x), x) \ (19) \quad B_1(x) \to S(x, f_i(x)) \text{ for } 2 \leq i \leq 3 \ (22)$$
$$B_0(x) \to B_1(f_1(x)) \ (20) \quad B_1(x) \to B_i(f_i(x)) \text{ for } 2 \leq i \leq 3 \ (23)$$
$$B_2(x) \wedge B_3(x) \to \bot \ (21) \quad B_i(x) \to B_4(x) \text{ for } 2 \leq i \leq 3 \ (24)$$
$$B_1(x) \wedge \bigwedge_{1 \leq j \leq 3} S(x, z_i) \to \bigvee_{1 \leq j < k \leq 3} z_j \approx z_k \quad (25) \ \}$$

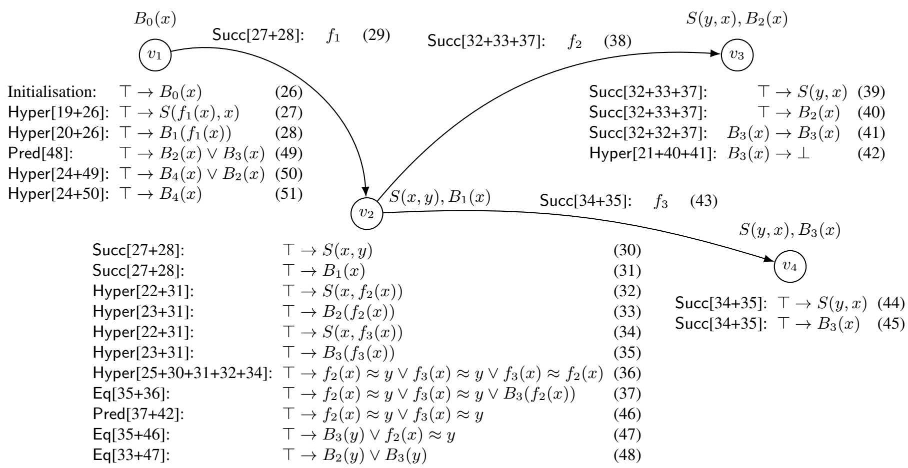

Fig. 2. Challenges in Extending the Consequence-Based Framework to $\mathcal{SHIQ}$

4 Extending the Framework to $\mathcal{SHIQ}$

We now present an example before formalising the calculus. Due to an interaction between counting quantifiers and inverse roles, a $\mathcal{SHIQ}$ ontology can impose more complex constraints on model elements than $\mathcal{ALCI}$. Let $\mathcal{O}_2$ be the $\mathcal{SHIQ}$ ontology shown in Figure 2; we argue that $\mathcal{O}_2 \models B_0(x) \to B_4(x)$ holds. To see why, consider an equality Herbrand interpretation I constructed from $B_0(a)$. Then, (19) and (20) derive $S(f_1(a), a)$ and $B_1(f_1(a))$; moreover, (22) and (23) derive $S(f_1(a), f_2(f_1(a)))$ and $B_2(f_2(f_1(a)))$, and $S(f_1(a), f_3(f_1(a)))$ and $B_3(f_3(f_1(a)))$. Due to (24) we derive $B_4(f_2(f_1(a)))$ and $B_4(f_3(f_1(a)))$. Finally, from (25) we derive (52).

$$f_2(f_1(a)) \approx a \vee f_3(f_1(a)) \approx a \vee f_3(f_1(a)) \approx f_2(f_1(a)) \tag{52}$$

We must satisfy at least one disjunct in (52). Disjunct $f_3(f_1(a)) \approx f_2(f_1(a))$ cannot be satisfied due to (21); but then, regardless of whether we satisfy $f_3(f_1(a)) \approx a$ or $f_2(f_1(a)) \approx a$, we derive $B_4(a)$; hence, the inference holds.

To prove this in our consequence-based framework, we must capture constraint (52) and its consequences. However, this cannot be done using standard description logic notation because DL concepts cannot identify specific successors and predecessors of $f_1(a)$—that is, they cannot say 'either the first or the second successor is equal to the predecessor'. Thus, our main challenges are to devise a method for representing all the relevant constraints that can be induced by $\mathcal{SHIQ}$ ontologies, and to ensure that such constraints are correctly propagated between adjacent contexts.

To address these challenges, we Skolemise existential quantifiers and transform axioms into DL-clauses. Skolemisation introduces function symbols that act as names for successors. Our clauses thus contain terms of the form x, $f_i(x)$, and y which have a special meaning in our setting: variable x represents the elements that a context stands for; $f_i(x)$ represents a successor of x; and y represents the predecessor of x. This allows us to represent constraint (52) as

$$f_2(x) \approx y \vee f_3(x) \approx y \vee f_3(x) \approx f_2(x). \tag{53}$$

Table 2 shows the inference rules of our calculus that are applicable to such a clausal representation. In each clause, literals are ordered from the smallest to the largest, and so the maximal literal is always on the right; moreover, clause numbers correspond to the order of clause derivation. In the rest of this section, we discuss the rules on our running example and show how they verify $\mathcal{O}_2 \models B_0(x) \rightarrow B_4(x)$; for brevity, we present only the inferences needed to produce the desired conclusion.

We first create context v_1 and initialise it with (26); this ensures that each interpretation represented by the context structure contains an element for which B_0 holds. Next, we derive (27) and (28) using hyperresolution. At this point, we could hyperresolve (22) and (28) to obtain $\top \rightarrow S(f_1(x), f_2(f_1(x)))$; however, such inferences could easily lead to nontermination of the calculus due to increased term nesting. Therefore, we require hyperresolution to map variable x in the DL-clauses to variable x in the context clauses; thus, in each context, hyperresolution derives only consequences about x.

Function symbol f_1 in clauses (27) and (28) is akin to an existential quantifier; consequently, the Succ rule introduces a fresh context v_2. Due to Skolemisation, edges in our context structure are labelled with function symbols, rather than concepts of the form $\exists S.B$ as in [22]. The rule uses an expansion strategy analogous to the $\mathcal{EL}$ strategy from Section 3. To determine which information to propagate to a successor, Definition 2 given below introduces a set $\mathsf{Su}(\mathcal{O})$ of *successor triggers*. In our example, DL-clause (25) contains atoms $B_1(x)$ and $S(x, z_i)$ in its body, where z_i can be mapped to a predecessor or a successor of x, so a context in which hyperresolution is applied to (25) will be interested in information about its predecessors; we reflect this by adding $B_1(x)$ and $S(x, y)$ to $\mathsf{Su}(\mathcal{O})$. Thus, the Succ rule introduces context v_2, sets its core to $B_1(x)$ and $S(x, y)$, and initialises the context with (30) and (31).

We next introduce (32)–(35) using hyperresolution, at which point we have sufficient information to apply hyperresolution to (25) to derive (36). Please note how the presence of (30) is crucial for this inference. We use paramodulation to deal with equality in clause (36). As is common in resolution-based theorem proving, we order the literals in a clause and apply inferences only to maximal literals; thus, we derive (37).

Clauses (32), (33), and (37) contain function symbol f_2, so we again apply the Succ rule and introduce context v_3. Due to clause (33), we know that $B_2(x)$ must always hold in v_2, so we add $B_2(x)$ to core_{v_2}. However, $B_3(f_2(x))$ occurs in clause (37) in a disjunction, so it holds only conditionally in v_2; we reflect this by including $B_3(x)$ in the body of clause (41). This allows us derive (42) using hyperresolution.

Clause (42) essentially says '$B_3(f_2(x))$ should not hold in the predecessor', so the Pred rule propagates this information to v_2. This produces clause (46); one can intuitively understand this inference as hyperresolution of (37) and (42), where we take into account that term $f_2(x)$ in context v_2 is represented as variable x in context v_3.

After two paramodulation steps, we derive clause (48), which essentially says 'the predecessor must satisfy $B_2(x)$ or $B_3(x)$'. The set $\mathsf{Pr}(\mathcal{O})$ of *predecessor triggers* from Definition 2 identifies this information as relevant to v_1: the DL-clauses in (24) contain $B_2(x)$ and $B_3(x)$ in their bodies, which are represented in v_2 as $B_2(y)$ and $B_3(y)$. Hence $\mathsf{Pr}(\mathcal{O})$ contains $B_2(y)$ and $B_3(y)$, which allows the Pred rule to derive (49).

After two more hyperresolution steps, we finally derive our target clause (51). Please note, however, that we cannot derive this if $B_4(x)$ were maximal in (50); thus, for completeness we require all atoms in the head of a query clause to be smallest. A similar observation applies to $\mathsf{Pr}(\mathcal{O})$: if $B_3(y)$ were maximal in (47), we would not derive (48) and propagate it to v_1; thus, we require all atoms in $\mathsf{Pr}(\mathcal{O})$ to be smallest too.

5 Formalising the Consequence-Based Algorithm for $\mathcal{SHIQ}$

Our calculus manipulates *context clauses*, which are constructed from *context terms* and *context literals* as described in Definition 1. Unlike in general resolution, we restrict context clauses to contain only variables x and y, which have a special meaning in our setting: variable x represents a point (i.e., a term) in the model, and y represents the predecessor of x; this naming convention is important for the rules of our calculus. This is in contrast to the DL-clauses of an ontology: these can contain variables x and z_i, and the latter can refer to either the predecessor or a successor of x.

Definition 1. *A* context $\mathcal{F}$-term *is a term of the form x, y, or $f(x)$ for $f \in \mathcal{F}$; a* context $\mathcal{P}$-term *is a term of the form $B(y)$, $B(x)$, $B(f(x))$, $S(x,y)$, $S(y,x)$, $S(x,f(x))$, or $S(f(x),x)$ for $B, R \in \mathcal{P}$ and $f \in \mathcal{F}$; and a* context term *is an $\mathcal{F}$-term or a $\mathcal{P}$-term. A* context literal *is a literal of the form $A \approx \mathsf{t}$ (called a* context atom*), $f(x) \bowtie g(x)$, or $f(x) \bowtie y$, for A a context $\mathcal{P}$-term and $\bowtie \in \{\approx, \not\approx\}$. A* context clause *is a clause with only function-free context atoms in the body, and only context literals in the head.*

Definition 2 introduces sets $\mathsf{Su}(\mathcal{O})$ and $\mathsf{Pr}(\mathcal{O})$, that identify the information that must be exchanged between adjacent contexts. Intuitively, $\mathsf{Su}(\mathcal{O})$ contains atoms that are of interest to a context's successor, and it guides the Succ rule whereas $\mathsf{Pr}(\mathcal{O})$ contains atoms that are of interest to a context's predecessor and it guides the Pred rule.

Definition 2. *Let $\mathcal{O}$ be an ontology. The set $\mathsf{Su}(\mathcal{O})$ of* successor triggers *of $\mathcal{O}$ is the smallest set of atoms such that, for each clause $\Gamma \to \Delta \in \mathcal{O}$, we have (i) $B(x) \in \Gamma$ implies $B(x) \in \mathsf{Su}(\mathcal{O})$, (ii) $S(x, z_i) \in \Gamma$ implies $S(x,y) \in \mathsf{Su}(\mathcal{O})$, and (iii) $S(z_i, x) \in \Gamma$ implies $S(y,x) \in \mathsf{Su}(\mathcal{O})$. The set $\mathsf{Pr}(\mathcal{O})$ of* predecessor triggers *is defined as*

$$\mathsf{Pr}(\mathcal{O}) = \{\, A\{x \mapsto y,\ y \mapsto x\} \mid A \in \mathsf{Su}(\mathcal{O}) \,\} \cup \{\, B(y) \mid B \text{ occurs in } \mathcal{O} \,\}.$$

Similarly to resolution, our calculus uses a term order $\succ$ to restrict its inferences. Definition 3 specifies the conditions that the order must satisfy. Conditions 1 and 2 ensure that $\mathcal{F}$-terms are compared uniformly across contexts; however, $\mathcal{P}$-terms can be compared in different ways in different contexts. Conditions 1 through 4 ensure that, when grounded with x and y mapping to a term its predecessor, respectively, the order is a *simplification order* [4]—a kind of term order commonly used in equational theorem proving. Finally, condition 5 ensures that atoms that might be propagated to a context's predecessor via the Pred rule are smallest, which is important for completeness.

Definition 3. *Let $\gtrdot$ be a total, well-founded order on function symbols. A* context term order $\succ$ *is an order on context terms satisfying the following conditions:*

1. *for each $f \in \mathcal{F}$, we have $f(x) \succ x \succ y$;*
2. *for all $f, g \in \mathcal{F}$ with $f \gtrdot g$, we have $f(x) \succ g(x)$;*
3. *for all terms s_1, s_2, and t and each position p in t, if $s_1 \succ s_2$, then $t[s_1]_p \succ t[s_2]_p$;*
4. *for each term s and each proper position p in s, we have $s \succ s|_p$; and*
5. *for each atom $A \approx \mathsf{t} \in \mathsf{Pr}(\mathcal{O})$ and each context term $s \notin \{x, y\}$, we have $A \not\succ s$.*

Order $\succ$ is extended to literals, also written $\succ$, as described in Section 2.

A lexicographic path order (LPO) [4] over context $\mathcal{F}$-terms and context $\mathcal{P}$-terms, in which x and y are treated as constants such that $x \succ y$, satisfies conditions 1 through 4. Furthermore, $\mathsf{Pr}(\mathcal{O})$ contains only atoms of the form $B(y)$, $S(x, y)$, and $S(y, x)$, which can always be smallest in the order; thus, condition 5 does not contradict the other conditions. Hence, an LPO that is relaxed for condition 5 satisfies Definition 3. In addition to orders, redundancy elimination techniques have proven crucial to the practical effectiveness of resolution calculi. We now define a criterion compatible with our setting.

Definition 4. *A set of clauses U* contains a clause $\Gamma \rightarrow \Delta$ up to redundancy, *written $\Gamma \rightarrow \Delta \,\hat{\in}\, U$, if*

1. *terms s and s' exist such that $s \approx s \in \Delta$ or $\{s \approx s',\ s \not\approx s'\} \subseteq \Delta$, or*
2. *a clause $\Gamma' \rightarrow \Delta' \in U$ exist such that $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$.*

Proposition 1. *For U a set of clauses and $C \in U$ a clause such that $C \,\hat{\in}\, U \setminus \{C\}$, for each clause C' with $C' \,\hat{\in}\, U$, we have $C' \,\hat{\in}\, U \setminus \{C\}$.*

Proposition 1 shows that our calculus is compatible with backward subsumption (which is captured in the Elim rule). Note that tautologies of the form $A \rightarrow A$ are *not necessarily* redundant since they can be used to initialise contexts. However, if our calculus were to derive both $A \rightarrow A$ and $A \rightarrow A \vee A'$ then the latter is always redundant.

We now formalise the notion of a context structure, and define soundness for a context structure. The latter captures the fact that core_v is not contained in the body of any context clause in $\mathcal{S}_v$.

Definition 5. *A* context structure *for an ontology $\mathcal{O}$ is a tuple $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$, where $\mathcal{V}$ is a finite set of* contexts, *$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{F}$ is a finite set of edges labelled with function symbols, function* core *assigns to each context v a conjunction core_v of atoms over the $\mathcal{P}$-terms from $\mathsf{Su}(\mathcal{O})$, function $\mathcal{S}$ assigns to each context v a finite set $\mathcal{S}_v$ of context clauses, and function $\succ$ assigns to each context v a context term order $\succ_v$. A context structure $\mathcal{D}$ is* sound *for $\mathcal{O}$ if the following conditions both hold:*

S1. $\mathcal{O} \models \mathsf{core}_v \wedge \Gamma \rightarrow \Delta$ *for each context $v \in \mathcal{V}$ and each clause $\Gamma \rightarrow \Delta \in \mathcal{S}_v$, and*
S2. $\mathcal{O} \models \mathsf{core}_u \rightarrow \mathsf{core}_v\{x \mapsto f(x), y \mapsto x\}$ *for each edge $\langle u, v, f \rangle \in \mathcal{E}$.*

Definition 6 introduces an expansion strategy—a parameter of our calculus that determines when and how to reuse contexts in order to satisfy existential restrictions. We have discussed the roles of expansion strategies in Section 3; moreover, in [22] we presented several expansion strategies for the DLs contained in $\mathcal{ALCI}$, and adapting these to $\mathcal{SHIQ}$ is straightforward.

Table 2. The rules of the consequence-based calculus for $\mathcal{SHIQ}$

Core	If	$A \in \mathsf{core}_v$,
		and $\top \rightarrow A \notin \mathcal{S}_v$,
	then	add $\top \rightarrow A$ to $\mathcal{S}_v$.
Hyper	If	$\bigwedge_{i=1}^{n} A_i \rightarrow \Delta \in \mathcal{O}$,
		σ is a substitution such that $\sigma(x) = x$,
		$\Gamma_i \rightarrow \Delta_i \vee A_i\sigma \in \mathcal{S}_v$ with $\Delta_i \not\succeq_v A_i\sigma$ for $i \in \{1, \ldots, n\}$,
		and $\bigwedge_{i=1}^{n} \Gamma_i \rightarrow \Delta\sigma \vee \bigvee_{i=1}^{n} \Delta_i \hat{\notin} \mathcal{S}_v$,
	then	add $\bigwedge_{i=1}^{n} \Gamma_i \rightarrow \Delta\sigma \vee \bigvee_{i=1}^{n} \Delta_i$ to $\mathcal{S}_v$.
Eq	If	$\Gamma_1 \rightarrow \Delta_1 \vee s_1 \approx t_1 \in \mathcal{S}_v$ with $s_1 \succ_v t_1$ and $\Delta_1 \not\succeq_v s_1 \approx t_1$,
		$\Gamma_2 \rightarrow \Delta_2 \vee s_2 \bowtie t_2 \in \mathcal{S}_v$ with $\bowtie \in \{\approx, \not\approx\}$ and $s_2 \succ_v t_2$ and $\Delta_2 \not\succeq_v s_2 \bowtie t_2$,
		$s_2\|_p = s_1$,
		and $\Gamma_1 \wedge \Gamma_2 \rightarrow \Delta_1 \vee \Delta_2 \vee s_2[t_1]_p \bowtie t_2 \hat{\notin} \mathcal{S}_v$,
	then	add $\Gamma_1 \wedge \Gamma_2 \rightarrow \Delta_1 \vee \Delta_2 \vee s_2[t_1]_p \bowtie t_2$ to $\mathcal{S}_v$.
Ineq	If	$\Gamma \rightarrow \Delta \vee t \not\approx t \in \mathcal{S}_v$ with $\Delta \not\succeq_v t \not\approx t$,
		and $\Gamma \rightarrow \Delta \hat{\notin} \mathcal{S}_v$,
	then	add $\Gamma \rightarrow \Delta$ to $\mathcal{S}_v$.
Factor	If	$\Gamma \rightarrow \Delta \vee s \approx t \vee s \approx t' \in \mathcal{S}_v$ with $\Delta \cup \{s \approx t\} \not\succeq_v s \approx t'$ and $s \succ_v t'$,
		and $\Gamma \rightarrow \Delta \vee t \not\approx t' \vee s \approx t' \hat{\notin} \mathcal{S}_v$,
	then	add $\Gamma \rightarrow \Delta \vee t \not\approx t' \vee s \approx t'$ to $\mathcal{S}_v$.
Elim	If	$\Gamma \rightarrow \Delta \in \mathcal{S}_v$ and
		$\Gamma \rightarrow \Delta \hat{\in} \mathcal{S}_v \setminus \{\Gamma \rightarrow \Delta\}$
	then	remove $\Gamma \rightarrow \Delta$ from $\mathcal{S}_v$.
Pred	If	$\langle u, v, f \rangle \in \mathcal{E}$,
		$\bigwedge_{i=1}^{m} A_i \rightarrow \bigvee_{i=m+1}^{m+n} A_i \in \mathcal{S}_v$,
		$\Gamma_i \rightarrow \Delta_i \vee A_i\sigma \in \mathcal{S}_u$ with $\Delta_i \not\succeq_u A_i\sigma$ for $1 \leq i \leq m$,
		$A_i \in \mathsf{Pr}(\mathcal{O})$ for each $m + 1 \leq i \leq m + n$,
		and $\bigwedge_{i=1}^{m} \Gamma_i \rightarrow \bigvee_{i=1}^{m} \Delta_i \vee \bigvee_{i=m+1}^{m+n} A_i\sigma \hat{\notin} \mathcal{S}_u$,
	then	add $\bigwedge_{i=1}^{m} \Gamma_i \rightarrow \bigvee_{i=1}^{m} \Delta_i \vee \bigvee_{i=m+1}^{m+n} A_i\sigma$ to $\mathcal{S}_u$;
	where	$\sigma = \{x \mapsto f(x), y \mapsto x\}$.
Succ	If	$\Gamma \rightarrow \Delta \vee A \in \mathcal{S}_u$ where $\Delta \not\succeq_u A$ and A contains $f(x)$, and
		no edge $\langle u, v, f \rangle \in \mathcal{E}$ exists such that $A \rightarrow A \hat{\in} \mathcal{S}_v$ for each $A \in K_2 \setminus \mathsf{core}_v$,
	then	let $\langle v, \mathsf{core}', \succ' \rangle := \mathsf{strategy}(K_1, \mathcal{D})$;
		if $v \in \mathcal{V}$, then let $\succ_v := \succ_v \cap \succ'$, and
		otherwise let $\mathcal{V} := \mathcal{V} \cup \{v\}$, $\mathsf{core}_v := \mathsf{core}'$, $\succ_v := \succ'$, and $\mathcal{S}_v := \emptyset$;
		add the edge $\langle u, v, f \rangle$ to $\mathcal{E}$; and
		add $A \rightarrow A$ to $\mathcal{S}_v$ for each $A \in K_2 \setminus \mathsf{core}_v$;
	where	$\sigma = \{x \mapsto f(x), y \mapsto x\}$,
		$K_1 = \{ A \in \mathsf{Su}(\mathcal{O}) \mid \top \rightarrow A\sigma \in \mathcal{S}_u \}$, and
		$K_2 = \{ A \in \mathsf{Su}(\mathcal{O}) \mid \Gamma' \rightarrow \Delta' \vee A\sigma \in \mathcal{S}_u$ and $\Delta' \not\succeq_u A\sigma \}$.

Definition 6. *An* expansion strategy *is a polynomial-time computable function* $\mathsf{strategy}$ *that takes a set of atoms* K *and a context structure* $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$*. The result of* $\mathsf{strategy}(K, \mathcal{D})$ *is a triple* $\langle v, \mathsf{core}', \succ' \rangle$ *where* core' *is a subset of* K*; either* $v \notin \mathcal{V}$ *is a fresh context, or* $v \in \mathcal{V}$ *is an existing context in* $\mathcal{D}$ *such that* $\mathsf{core}_v = \mathsf{core}'$*; and* $\succ'$ *is a context term order.*

We now present the main theorems. Full proofs of all technical results can be found in the appendix. Theorem 1 proves that all clauses derived by our calculus are indeed conclusions of the input ontology, and Theorem 2 is our statement of completeness.

Theorem 1. *For any strategy, applying a rule from Table 2 to an ontology* $\mathcal{O}$ *and a context structure* $\mathcal{D}$ *that is sound for* $\mathcal{O}$ *produces a context structure that is sound for* $\mathcal{O}$*.*

Theorem 2. *Let* $\mathcal{O}$ *be an ontology, and let* $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$ *be a context structure such that no inference rule from Table 2 is applicable to* $\mathcal{O}$ *and* $\mathcal{D}$*. Then, for each query clause* $\Gamma_Q \to \Delta_Q$ *and each context* $q \in \mathcal{V}$ *that satisfy all of the following conditions, we have* $\Gamma_Q \to \Delta_Q \hat{\in} \mathcal{S}_q$*.*

C1. $\mathcal{O} \models \Gamma_Q \to \Delta_Q$*.*
C2. *For each atom* $A \approx \mathsf{t} \in \Delta_Q$ *and each context term* $s \notin \{x, y\}$ *such that* $A \succ_q s$*, we have* $s \approx \mathsf{t} \in \Delta_Q \cup \mathsf{Pr}(\mathcal{O})$*.*
C3. *For each* $A \in \Gamma_Q$*, we have* $\Gamma_Q \to A \hat{\in} \mathcal{S}_q$*.*

Theorems 1 and 2 result in the following algorithm for deciding $\mathcal{O} \models \Gamma_Q \to \Delta_Q$.

1. Create an empty context structure $\mathcal{D}$, introduce a context q, and, for each $A \in \Gamma_Q$, add $\Gamma_Q \to A$ to $\mathcal{S}_q$ in order to satisfy condition C3.
2. Initialise $\succ_q$ in a way that satisfies condition C2, and select an expansion strategy.
3. Saturate $\mathcal{D}$ and $\mathcal{O}$ using the inference rules from Table 2.
4. $\Gamma_Q \to \Delta_Q$ holds if and only if $\Gamma_Q \to \Delta_Q \hat{\in} \mathcal{S}_v$.

Proposition 2. *For each expansion strategy that introduces at most exponentially many contexts, the consequence-based calculus for* $\mathcal{SHIQ}$ *is worst-case optimal.*

Proof. The number $\wp$ of different context clauses that can be generated using the symbols in $\mathcal{O}$ is clearly at most exponential in the size of $\mathcal{O}$, and the number m of clauses participating in each inference is linear in the size of $\mathcal{O}$. Hence, with k contexts, the number of inferences is bounded by $(k \cdot \wp)^m$; if k is at most exponential in the size of $\mathcal{O}$, the number of inferences is exponential as well. Thus, if at most exponentially many contexts are introduced, our algorithm runs in exponential time, which is worst-case optimal for $\mathcal{SHIQ}$ [3]. □

6 Conclusion

We have presented the first consequence based calculus for $\mathcal{SHIQ}$, a DL that includes both disjunction and counting quantifiers. Our calculus combines ideas from state of the art resolution and (hyper)tableau calculi, including the use of ordered paramodulation for efficient equality reasoning. In spite of its increased complexity, the calculus mimics existing and well proven $\mathcal{EL}$ calculi on $\mathcal{EL}$ ontologies. Thus, although implementation and evaluation remain as future work, we believe that the calculus is likely to work well on 'mostly-$\mathcal{EL}$' ontologies—a type of ontology that occurs commonly in practice.

References

1. H. Andréka, J. van Benthem, and I. Németi. Modal Languages and Bounded Fragments of Predicate Logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998.
2. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ Envelope. In L. Pack Kaelbling and A. Saffiotti, editors, *Proc. of the 19th Int. Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh, UK, July 30–August 5 2005. Morgan Kaufmann Publishers.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, January 2003.
4. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
5. P. Baumgartner, U. Furbach, and I. Niemelä. Hyper Tableaux. In *Proc. of the European Workshop on Logics in Artificial Intelligence (JELIA '96)*, number 1126 in LNAI, pages 1–17, Évora, Portugal, September 30–October 3 1996. Springer.
6. P. Baumgartner and R. A. Schmidt. Blocking and Other Enhancements for Bottom-Up Model Generation Methods. In U. Furbach and N. Shankar, editors, *Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *LNCS*, pages 125–139, Seattle, WA, USA, August 17–20 2006. Springer.
7. S. Demri and H. de Nivelle. Deciding Regular Grammar Logics with Converse Through First-Order Logic. *Journal of Logic, Language and Information*, 14(3):289–329, 2005.
8. Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 Reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.
9. Rajeev Goré and Linh Anh Nguyen. EXPTIME Tableaux with Global Caching for Description Logics with Transitive Roles, Inverse Roles and Role Hierarchies. In Nicola Olivetti, editor, *Proc. of the 16th Int. Conf. on Automated Reasoning with Tableaux and Related Methods (TABLEAUX 2007)*, volume 4548 of *LNCS*, pages 133–148, Aix en Provence, France, July 3–6 2007. Springer.
10. E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *Proc. of the 12th IEEE Symposium on Logic in Computer Science (LICS '97)*, pages 306–317, Warsaw, Poland, June 29–July 2 1997. IEEE Computer Society.
11. V. Haarslev and R. Möller. RACER System Description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proc. of the 1st Int. Joint Conf. on Automated Reasoning (IJCAR 2001)*, volume 2083 of *LNAI*, pages 701–706, Siena, Italy, June 18–23 2001. Springer.
12. U. Hustadt and R.A. Schmidt. On the Relation of Resolution and Tableaux Proof Systems for Description Logics. In Thomas Dean, editor, *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI 1999)*, pages 110–117, Stockholm, Sweden, July 31 – August 6 1999. Morgan Kaufmann.
13. Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Deciding Expressive Description Logics in the Framework of Resolution. *Information & Computation*, 206(5):579–601, 2008.
14. Y. Kazakov. Consequence-Driven Reasoning for Horn SHIQ Ontologies. In Craig Boutilier, editor, *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, pages 2040–2045, Pasadena, CA, USA, July 11–17 2009.
15. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
16. R. Nieuwenhuis and A. Rubio. Theorem Proving with Ordering and Equality Constrained Clauses. *Journal of Symbolic Computation*, 19(4):312–351, 1995.
17. H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-Based Methods for Modal Logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.

18. M. Ortiz, S. Rudolph, and M. Simkus. Worst-Case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2. In F. Lin, U. Sattler, and M. Truszczynski, editors, *Proc. of the 12th Int. Conf. on Knowledge Representation and Reasoning (KR 2010)*, pages 269–279, Toronto, ON, Canada, May 9–13 2010. AAAI Press.
19. A. Riazanov and A. Voronkov. The design and implementation of VAMPIRE. *AI Communications*, 15(2–3):91–110, 2002.
20. R. A. Schmidt and U. Hustadt. A Principle for Incorporating Axioms into the First-Order Translation of Modal Formulae. In F. Baader, editor, *Proc. of the 19th Int. Conf. on Automated Deduction (CADE-19)*, volume 2741 of *LNAI*, pages 412–426, Miami Beach, FL, USA, July 28–August 2 2003. Springer.
21. S. Schulz. E—A Brainiac Theorem Prover. *AI Communications*, 15(2–3):111–126, 2002.
22. František Simančík, Boris Motik, and Ian Horrocks. Consequence-Based and Fixed-Parameter Tractable Reasoning in Description Logics. *Artificial Intelligence*, 209:29–77, 2014.
23. F. Simančík, Y. Kazakov, and I. Horrocks. Consequence-Based Reasoning beyond Horn Ontologies. In Toby Walsh, editor, *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pages 1093–1098, Barcelona, Spain, July 16–22 2011.
24. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
25. D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *LNAI*, pages 292–297, Seattle, WA, USA, August 17–20 2006. Springer.

A Proof of Theorem 1

In this chapter, we show that our calculus is sound, as stated in Theorem 1. The proof is analogous to the soundness proof of ordered superposition [16].

Theorem 1. *For any strategy, applying a rule from Table 2 to an ontology $\mathcal{O}$ and a context structure $\mathcal{D}$ that is sound for $\mathcal{O}$ produces a context structure that is sound for $\mathcal{O}$.*

Proof. Let $\mathcal{O}$ be an ontology, let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$ be a context structure sound for $\mathcal{O}$, and consider an application of an inference rule from Table 2 to $\mathcal{D}$ and $\mathcal{O}$. To prove the theorem, we show that each clause produced by the rule is a context clause and that it satisfies conditions S1 and S2 of Definition 5. Condition S1 holds obviously for the rules different from Hyper, Eq, and Pred. For condition S2, we rely on soundness of hyperresolution: for arbitrary formulas ω, ϕ_i, ψ_i, and γ_i, $1 \leq i \leq n$, we have

$$\left\{ \bigwedge_{j=1}^{n} \phi_j \rightarrow \omega \text{ and } \gamma_i \rightarrow \psi_i \vee \phi_i \right\}_{1 \leq i \leq n} \models \bigwedge_{i=1}^{n} \gamma_i \rightarrow \bigvee_{i=1}^{n} \psi_i \vee \omega. \tag{54}$$

To prove the claim, we consider each rule from Table 2 and assume that the rule is applied to clauses, contexts, and edges as shown in the table; then, we show that the clause produced by the rule satisfies condition S1 of Definition 5; and for the Succ rule, we also show that the edge introduced by the rule satisfies condition S2.

(Core) For each $A \in \mathsf{core}_v$, we clearly have $\mathcal{O} \models \mathsf{core}_v \rightarrow A$.

(Hyper) Since $\mathcal{D}$ is sound for $\mathcal{O}$, we have $\mathcal{O} \models \mathsf{core}_v \wedge \Gamma_i \rightarrow \Delta_i \vee A_i\sigma$ for each i with $1 \leq i \leq n$. By (54), we have $\mathcal{O} \models \mathsf{core}_v \wedge \bigwedge_{i=1}^{n} \Gamma_i \rightarrow \bigvee_{i=1}^{n} \Delta_i \vee \Delta\sigma$. Moreover, substitution σ satisfies $\sigma(x) = x$, all premises are context clauses, and $\mathcal{O}$ contains only DL-clauses; thus, the inference rule can match an atom $S(x, z_i)$ or $S(z_i, x)$ in an ontology clause to atoms $S(x, y)$ or $S(x, f(x))$ in the context clause, and so $\sigma(z_i)$ is either y or $f(x)$; thus, the result is a context clause.

(Eq) Since $\mathcal{D}$ is sound for $\mathcal{O}$, properties (55) and (56) hold. Moreover, clause in (57) is a logical consequence of the clauses in (55) and (56), so property (57) holds, as required.

$$\mathcal{O} \models \mathsf{core}_v \wedge \Gamma_1 \rightarrow \Delta_1 \vee s_1 \approx t_1 \tag{55}$$

$$\mathcal{O} \models \mathsf{core}_v \wedge \Gamma_2 \rightarrow \Delta_2 \vee s_2 \bowtie t_2 \tag{56}$$

$$\mathcal{O} \models \mathsf{core}_v \wedge \Gamma_1 \wedge \Gamma_2 \rightarrow \Delta_1 \vee \Delta_2 \vee s_2[t_1]_p \bowtie t_2 \tag{57}$$

Moreover, term s_1 is always of the form $g(f(x))$, term t_1 is of the form $h(f(x))$ or y, and term s_2 is of the form $g(f(x))$, $B(f(g(x)))$, $S(x, f(g(x)))$, or $S(f(g(x)), x)$; thus, $s_2[t_1]_p$ is a context term, and so the result is a context clause.

(Ineq) Since $\mathcal{D}$ is sound for $\mathcal{O}$, we have $\mathcal{O} \models \mathsf{core}_v \wedge \Gamma \rightarrow \Delta \vee t \not\approx t$; but then, we clearly have $\mathcal{O} \models \mathsf{core}_v \wedge \Gamma \rightarrow \Delta$, as required.

(Factor) Since $\mathcal{D}$ is sound for $\mathcal{O}$, property (58) holds. Moreover, clause in (59) is a logical consequence of the clause in (58), so property (59) holds, as required.

$$\mathcal{O} \models \mathsf{core}_v \wedge \Gamma \rightarrow \Delta \vee s \approx t \vee s \approx t' \tag{58}$$

$$\mathcal{O} \models \mathsf{core}_v \wedge \Gamma \rightarrow \Delta \vee t \not\approx t' \vee s \approx t' \tag{59}$$

(Elim) The resulting context structure contains a subset of the clauses than $\mathcal{D}$, so it is clearly sound for $\mathcal{O}$.

(Pred) Let $\sigma = \{x \mapsto f(x), y \mapsto x\}$. Since $\mathcal{D}$ is sound for $\mathcal{O}$, properties (60)–(62) hold. Now clause in (63) is an instance of the clause in (60), so property (63) holds. But then, by (54), properties (60) and (61) imply property (64). Finally, properties (62) and (64) imply property (65), as required.

$$\mathcal{O} \models \mathsf{core}_v \wedge \textstyle\bigwedge_{i=1}^{m} A_i \rightarrow \bigvee_{j=m+1}^{m+n} A_j \tag{60}$$

$$\mathcal{O} \models \mathsf{core}_u \wedge \Gamma_i \rightarrow \Delta_i \vee A_i\sigma \qquad \text{for } 1 \leq i \leq m \tag{61}$$

$$\mathcal{O} \models \mathsf{core}_u \rightarrow \mathsf{core}_v\sigma \tag{62}$$

$$\mathcal{O} \models \mathsf{core}_v\sigma \wedge \textstyle\bigwedge_{i=1}^{m} A_i\sigma \rightarrow \bigvee_{j=m+1}^{m+n} A_j\sigma \tag{63}$$

$$\mathcal{O} \models \mathsf{core}_v\sigma \wedge \mathsf{core}_v \wedge \textstyle\bigwedge_{i=1}^{m} \Gamma_i \rightarrow \bigvee_{j=m+1}^{m+n} A_j\sigma \tag{64}$$

$$\mathcal{O} \models \mathsf{core}_u \wedge \textstyle\bigwedge_{i=1}^{m} \Gamma_i \rightarrow \bigvee_{j=m+1}^{m+n} A_j\sigma \tag{65}$$

For each $m + 1 \leq i \leq m + n$, we have $A_i \in \mathsf{Pr}(\mathcal{O})$, so A_i is of the form $B(y)$, $S(x, y)$, or $S(y, x)$; but then, the definition of σ ensures that $A_i\sigma$ is a context atom, as required.

(Succ) Let $\sigma = \{x \mapsto f(x), y \mapsto x\}$. For each clause $A \rightarrow A$ added to $\mathcal{S}_v$, we clearly have $\mathcal{O} \models \mathsf{core}_v \wedge A \rightarrow A$, as required for condition S1 of Definition 5. Moreover, assume that the inference rule adds an edge $\langle u, v, f_k\rangle$ to $\mathcal{E}$; since $\mathcal{D}$ is sound for $\mathcal{O}$, we have (66); by Definition 6, we have $\mathsf{core}_v \subseteq K_1$.

$$\mathcal{O} \models \mathsf{core}_u \rightarrow A\sigma \qquad \text{for each } A \in K_1 \tag{66}$$

$$\mathcal{O} \models \mathsf{core}_u \rightarrow \mathsf{core}_v\sigma \tag{67}$$

But then, property (67) holds, as required for condition S2 of Definition 5. □

B Preliminaries: Rewrite Systems

In the proof of Theorem 2 we construct a model of an ontology, which, as is common in equational theorem proving, we represent using a ground *rewrite system*. We next recapitulate the definitions of rewrite systems, following the presentation by [4].

Let $\mathcal{T}$ be the set of all ground terms constructed using a distinguished constant c (of sort $\mathcal{F}$), the function symbols from $\mathcal{F}$, and the predicate symbols from $\mathcal{P}$. A (ground) *rewrite system* R is a binary relation on $\mathcal{T}$. Each pair $(s, t) \in R$ is called a *rewrite rule* and is commonly written as $s \Rightarrow t$. The *rewrite relation* $\rightarrow_R$ for R is the smallest binary relation on $\mathcal{T}$ such that, for all terms $s_1, s_2, t \in \mathcal{T}$ and each (not necessarily proper) position p in t, if $s_1 \Rightarrow s_2 \in R$, then $t[s_1]_p \rightarrow_R t[s_2]_p$. Moreover, $\xrightarrow{*}_R$ is the reflexive–transitive closure of $\rightarrow_R$, and $\overset{*}{\leftrightarrow}_R$ is the reflexive–symmetric–transitive closure of $\rightarrow_R$. A term s is *irreducible by* R if no term t exists such that $s \rightarrow_R t$; and a literal, clause, or substitution α is *irreducible by* R if no term occurring in α is irreducible by R. Moreover, term t is a *normal form* of s w.r.t. R if $s \overset{*}{\leftrightarrow}_R t$ and t is irreducible by R. We consider the following properties of rewrite systems.

- R is *terminating* if no infinite sequence $s_1, s_2, \dots$ of terms exists such that, for each i, we have $s_i \rightarrow_R s_{i+1}$.
- R is *left-reduced* if, for each $s \Rightarrow t \in R$, the term s is irreducible by $R \setminus \{s \Rightarrow t\}$.
- R is *Church-Rosser* if, for all terms t_1 and t_2 such that $t_1 \overset{*}{\leftrightarrow}_R t_2$, a term z exists such that $t_1 \overset{*}{\rightarrow}_R z$ and $t_2 \overset{*}{\rightarrow}_R z$.

If R is terminating and left-reduced, then R is Church-Rosser [4, Theorem 2.1.5 and Exercise 6.7]. If R is Church-Rosser, then each term s has a unique normal form t such that $s \overset{*}{\rightarrow}_R t$ holds. The *Herbrand interpretation induced by* a Church-Rosser system R is the set R^* such that, for all $s, t \in \mathcal{T}$, we have $s \approx t \in R^*$ if and only if $s \overset{*}{\leftrightarrow}_R t$.

Term orders can be used to prove termination of rewrite systems. A term order $\succ$ is a *simplification order* if the following conditions hold:

- for all terms s_1, s_2, and t, all positions p in t, and all substitutions σ, we have that $s_1 \succ s_2$ implies $t[s_1\sigma]_p \succ t[s_2\sigma]_p$; and
- for each term s and each proper position p in s, we have $s \succ s|_p$.

If, for R a rewrite system, a simplification order $\succ$ exists such that $s \Rightarrow t \in R$ implies $s \succ t$, then R is terminating [4, Theorems 5.2.3 and 5.4.8], and $s \rightarrow_R t$ implies $s \succ t$.

C An Outline of the Proof of Theorem 2

To prove Theorem 2, we fix an ontology $\mathcal{O}$, a context structure $\mathcal{D}$, a query clause $\Gamma_Q \rightarrow \Delta_Q$, and a context q such that properties C2 and C3 of Theorem 2 are satisfied and $\Gamma_Q \rightarrow \Delta_Q \hat{\notin} \mathcal{S}_q$ holds, and we construct an interpretation that satisfies $\mathcal{O}$ but refutes $\Gamma_Q \rightarrow \Delta_Q$. We reuse techniques from equational theorem proving [16] and represent this interpretation by a *rewrite system* R—a finite set of rules of the form $l \Rightarrow r$. Intuitively, such a rule says that that any two terms of the form $f_1(\dots f_n(l) \dots)$ and $f_1(\dots f_n(r) \dots)$ with $n \geq 0$ are equal, and that we can prove this equality in one step by rewriting (i.e., replacing) l with r. Rewrite system R induces a Herbrand equality interpretation R^* that contains each $l \approx r$ for which the equality between l and r can be verified using a finite number of such rewrite steps. The universe of R^* consists of $\mathcal{F}$- and $\mathcal{P}$-terms constructed using the symbols in $\mathcal{F}$ and $\mathcal{P}$, and a special constant c; for convenience, let $\mathcal{T}$ be the set of all $\mathcal{F}$-terms from this universe.

We obtain R by unfolding the context structure $\mathcal{D}$ starting from context q: we map each $\mathcal{F}$-term $t \in \mathcal{T}$ to a context X_t in $\mathcal{D}$, and we use the clauses in $\mathcal{S}_{X_t}$ to construct a model fragment R_t—the part of R that satisfies the DL-clauses of $\mathcal{O}$ when x is mapped to t. The key issue is to ensure compatibility between adjacent model fragments: when moving from a *predecessor* term t' to a *successor* term $t = f(t')$, we must ensure that adding R_t to $R_{t'}$ does not affect the truth of the DL-clauses of $\mathcal{O}$ at term t'; in other words, the model fragment constructed at t must respect the choices made at t'. We represent these choices by a ground clause $\Gamma_t \rightarrow \Delta_t$: conjunction Γ_t contains atoms that are 'inherited' from t' and so must hold at t, and disjunction Δ_t contains atoms that must not hold at t because t' relies on their absence.

The model fragment construction takes as parameters a term t, a context $v = X_t$, and a clause $\Gamma_t \rightarrow \Delta_t$. Let N_t be the set of ground clauses obtained from $\mathcal{S}_v$ by mapping

x to t and y to the predecessor of t (if it exists), and whose body is contained in Γ_t. Moreover, let Su_t and Pr_t be obtained from $\mathsf{Su}(\mathcal{O})$ and $\mathsf{Pr}(\mathcal{O})$ by mapping x to t and y to the predecessor of t if one exists; thus, Su_t contains the ground atoms of interest to the successors of t, and Pr_t contains the ground atoms of interests to the predecessor of t. The model fragment for t can be constructed if the following properties hold:

L1. $\Gamma_t \rightarrow \Delta_t \hat{\not\in} N_t$.
L2. If $t = c$, then $\Delta_t = \Delta_Q$; and if $t \neq c$, then $\Delta_t \subseteq \mathsf{Pr}_t$.
L3. For each $A \in \Gamma_t$, we have $\Gamma_t \rightarrow A \,\hat{\in}\, N_t$.

The model fragment construction produces a rewrite system R_t such that

F1. $R_t^* \models N_t$, and
F2. $R_t^* \not\models \Gamma_t \rightarrow \Delta_t$—that is, all of Γ_t, but none of Δ_t hold in R_t^*, and so the model fragment at t is compatible with the 'inherited' constraints.

We construct R_t using the standard techniques from paramodulation-based theorem proving. First, we order all clauses in N_t into a sequence $C^i = \Gamma^i \rightarrow \Delta^i \vee L^i$, for $1 \leq n$, that is compatible with the context ordering $\succ_v$ in a particular way. Next, we initialise R_t to $\emptyset$, and then we examine each clause C^i in this sequence; if C^i does not hold in the model constructed thus far, we make the clause true by adding L^i to R_t. To prove condition F1, we assume for the sake of a contradiction that a clause C^i with smallest i exists such that $R_t^* \not\models C^i$, and we show that an application of the Eq, Ineq, or Factor rule to C^i necessarily produces a clause C^j such that $R_t^* \not\models C^j$ and $j < i$. Conditions L1 through L3 allow us to satisfy condition F2. Due to condition L2 and condition 5 of Definition 3, we can order the clauses in the sequence such that each clause C^i capable of producing an atom from Δ_t comes before any other clause in the sequence; and then we use condition L1 to show that no such clause actually exists. Moreover, condition L3 ensures that all atoms in Γ_t are actually produced in R_t^*.

To obtain R, we inductively unfold $\mathcal{D}$, and at each step we invoke the model fragment construction with suitably defined parameters. For the base case, we map constant c to context $X_c = q$, and we define $\Gamma_c = \Gamma_Q$ and $\Delta_c = \Delta_Q$; then, conditions L1 and L2 hold by definition, and condition L3 holds by property C3 of Theorem 2. For the induction step, we assume that we have already mapped some term t' to a context $u = X_{t'}$, and, for each function symbol $f \in \mathcal{F}$, we consider term $t = f(t')$.

- If t does not occur in an atom in $R_{t'}$, we let $R_t = \{t \Rightarrow c\}$ and thus make t equal to c. Term t is thus interpreted in exactly the same way as c, so we stop the unfolding.
- If $R_{t'}$ contains a rule $t \Rightarrow s$, then t and s are equal, and so we can interpret t exactly as s; consequently, we stop the unfolding.
- In all other cases, the Succ rule ensures that $\mathcal{D}$ contains an edge $\langle u, v, f \rangle$ such that v satisfies all preconditions of the rule, so we define $X_t = v$. Moreover, we let $\Gamma_t = R_{t'}^* \cap \mathsf{Su}_t$ be the set of atoms that hold at t' and are relevant to t , and we let $\Delta_t = \mathsf{Pr}_t \setminus R_{t'}^*$ be the set of atoms that do not hold at t' and are relevant to t. We finally show that such Γ_t and Δ_t satisfy condition L1: if that were not the case, then the Pred rule derives a clause in $N_{t'}$ that is not true in $R_{t'}^*$.

After processing all relevant terms, we let R be the union of all R_t from the above construction. To show that R^* satisfies $\mathcal{O}$, we consider a DL-clause $\Gamma \rightarrow \Delta \in \mathcal{O}$ and a substitution τ that makes the clause ground. W.l.o.g. we can assume that τ is irreducible by R—that is, it does not contain terms that can we rewritten using the rules in R. Since each model fragment satisfies condition F2, we can evaluate $\Gamma\tau \rightarrow \Delta\tau$ in $R^*_{\tau(x)}$ instead of R^*. Moreover, we show that $R^*_{\tau(x)} \models \Gamma\tau \rightarrow \Delta\tau$ holds: if that were not the case, the Hyper rule derives a clause in $N_{\tau(x)}$ that violates condition F1. Finally, we show that the same holds for the query clause $\Gamma_Q \rightarrow \Delta_Q$, which completes our proof.

D Proof of Theorem 2

Theorem 2. *Let $\mathcal{O}$ be an ontology, and let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$ be a context structure such that no inference rule from Table 2 is applicable to $\mathcal{O}$ and $\mathcal{D}$. Then, for each query clause $\Gamma_Q \rightarrow \Delta_Q$ and each context $q \in \mathcal{V}$ that satisfy all of the following conditions, we have $\Gamma_Q \rightarrow \Delta_Q \mathrel{\hat{\in}} \mathcal{S}_q$.*

- *C1.* $\mathcal{O} \models \Gamma_Q \rightarrow \Delta_Q$.
- *C2. For each atom $A \approx \mathsf{t} \in \Delta_Q$ and each context term $s \notin \{x, y\}$ such that $A \succ_q s$, we have $s \approx \mathsf{t} \in \Delta_Q \cup \mathsf{Pr}(\mathcal{O})$.*
- *C3. For each $A \in \Gamma_Q$, we have $\Gamma_Q \rightarrow A \mathrel{\hat{\in}} \mathcal{S}_q$.*

In this section, we fix an ontology $\mathcal{O}$, a context structure $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{S}, \mathsf{core}, \succ \rangle$, a context $q \in \mathcal{V}$, and a query clause $\Gamma_Q \rightarrow \Delta_Q$ such that conditions C3 and C2 of Theorem 2 are satisfied, and we show the contrapositive of condition C1: if $\Gamma_Q \rightarrow \Delta_Q \mathrel{\hat{\notin}} \mathcal{S}_q$, then $\mathcal{O} \not\models \Gamma_Q \rightarrow \Delta_Q$. To this end, we construct a rewrite system R such that the induced Herbrand model R^* satisfies all clauses in $\mathcal{O}$, but not $\Gamma_Q \rightarrow \Delta_Q$. We construct the model using a distinguished constant c, the unary function symbols from $\mathcal{F}$, and the unary and binary predicate symbols from $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Let t and s be terms with $t = f(s)$; then, s is the *predecessor* of t, and t is a *successor* of s; by these definitions, a constant has no predecessor. Let t be a term. The *$\mathcal{F}$-neighbourhood* of t is the set of $\mathcal{F}$-terms t, $f(t)$ with $f \in \mathcal{F}$, and the predecessor t' of t if one exists; the *$\mathcal{P}$-neighbourhood* of t contains $\mathcal{P}$-terms $B(t)$, $S(t, f(t))$, $S(f(t), t)$, $B(f(t))$, and, if t has the predecessor t', also $\mathcal{P}$-terms $S(t', t)$, $S(t, t')$, and $B(t')$, for all $B \in \mathcal{P}_1$ and $S \in \mathcal{P}_2$. Let σ_t be the substitution such that $\sigma_t(x) = t$ and, if t has the predecessor t', then $\sigma_t(y) = t'$. Finally, for each term t, we define sets of atoms Pr_t and Su_t as follows:

$$\mathsf{Su}_t = \{\, A\sigma_t \mid A \in \mathsf{Su}(\mathcal{O}) \text{ and } A\sigma_t \text{ is ground} \,\} \tag{68}$$

$$\mathsf{Pr}_t = \{\, A\sigma_t \mid A \in \mathsf{Pr}(\mathcal{O}) \text{ and } A\sigma_t \text{ is ground} \,\} \tag{69}$$

D.1 Constructing a Model Fragment

In this section, we show how, given a term t, we can generate a part of the model of $\mathcal{O}$ that covers the neighbourhood of t. In the rest of Appendix D.1, we fix the following parameters to the model fragment generation process:

- t is a ground $\mathcal{F}$-term,

- v is a context in $\mathcal{D}$,
- Γ_t is a conjunction of atoms, and
- Δ_t is a disjunction of atoms.

Let N_t be the set of ground clauses obtained from $\mathcal{S}_v$ as follows:

$$N_t = \{\Gamma\sigma_t \to \Delta\sigma_t \mid \Gamma \to \Delta \in \mathcal{S}_v,\ \text{both } \Gamma\sigma_t \text{ and } \Delta\sigma_t \text{ are ground, and } \Gamma\sigma_t \subseteq \Gamma_t\}$$

We assume that the following conditions hold.

L1. $\Gamma_t \to \Delta_t \mathrel{\tilde{\notin}} N_t$.
L2. If $t = c$, then $\Delta_t = \Delta_Q$; and if $t \neq c$, then $\Delta_t \subseteq \mathsf{Pr}_t$.
L3. For each $A \in \Gamma_t$, we have $\Gamma_t \to A \mathrel{\hat{\in}} N_t$.

We next construct a rewrite system R_t that satisfies $R_t^* \models N_t$ and $R_t^* \not\models \Gamma_t \to \Delta_t$. Throughout Appendix D.1, we treat the terms in the $\mathcal{F}$-neighbourhood of t is if they were constants. Thus, even though the rewrite system R will contain terms t and $f(t)$, we will not consider terms with further nesting.

D.1.1 Grounding the Context Order

To construct R_t, we need an order on the terms in the neighbourhood of t that is compatible with $\succ_v$. To this end, let $>_t$ be a total, strict, simplification order on the set of ground terms constructed using the $\mathcal{F}$-neighbourhood of t and the predicate symbols in $\mathcal{P}$ that satisfies the following conditions for all context terms s_1 and s_2 such that $s_1\sigma_t$ and $s_2\sigma_t$ are both ground, and for t' the predecessor of t (if one exists).

O1. $s_1 \succ_v s_2$ implies $s_1\sigma_t >_t s_2\sigma_t$.
O2. $s_1\sigma_t \approx \mathsf{t} \in \Delta_t$ and $s_1\sigma_t >_t s_2\sigma_t$ and $s_2\sigma_t \notin \{t, t'\}$ imply $s_2\sigma_t \approx \mathsf{t} \in \Delta_t$.

Condition C2 of Theorem 2 and condition 5 of Definition 3 ensure that the order $\succ_v$ on (nonground) context terms can be grounded in a way compatible with condition L2. Moreover, since in this section we treat all $\mathcal{F}$-terms as constants, we can make the $\mathcal{P}$-terms of the form $B(t')$, $S(t', t)$, and $S(t, t')$ smaller than other $\mathcal{F}$- and $\mathcal{P}$-terms (i.e.,, we need not worry about defining the order on the predecessor of t' or on the ancestors of $f(t)$). Thus, at least one such order exists, so in the rest of this section we fix an arbitrary such order $>_t$. We extend $>_t$ to ground literals (also written $>_t$) by identifying each $s \not\approx t$ with the multiset $\{s, s, t, t\}$ and each $s \approx t$ with the multiset $\{s, t\}$, and then comparing the result using the multiset extension (as defined in Section 2). Finally, we further extend $>_t$ to disjunctions of ground literals (also written $>_t$) by identifying each disjunction $\bigvee_{i=1}^n L_i$ with the multiset $\{L_1, \ldots, L_n\}$ and then comparing the result using the multiset extension.

D.1.2 Constructing the Rewrite System R_t

We arrange all clauses in N_t into a sequence $C^1, \ldots, C^n$. Since the body of each C^i is a subset of Γ_t, no C^i can contain $\bot$ in its head as that would contradict condition L1; thus, we can assume that each C^i is of the form $C^i = \Gamma^i \to \Delta^i \vee L^i$ where $L^i >_t \Delta^i$, literal L^i is of the form $L^i = l^i \bowtie r^i$ with $\bowtie \in \{\approx, \not\approx\}$, and $l^i \geq_t r^i$. For the rest of Appendix D.1, we reserve C^i, Γ^i, Δ^i, L^i, l^i, and r^i for referring to the (parts of) the clauses in this sequence. Finally, we assume that, for all $1 \leq i < j \leq n$, we have $\Delta^j \vee L^j \geq_t \Delta^i \vee L^i$.

We next define the sequence $R_t^0, \ldots, R_t^n$ of rewrite systems by setting $R_t^0 = \emptyset$ and defining each R_t^i with $1 \leq i \leq n$ inductively as follows:

- $R_t^i = R_t^{i-1} \cup \{l^i \Rightarrow r^i\}$ if L^i is of the form $l^i \approx r^i$ such that
  R1. $(R_t^{i-1})^* \not\models \Delta^i \vee l^i \approx r^i$,
  R2. $l^i >_t r^i$,
  R3. l^i is irreducible by R_t^{i-1}, and
  R4. $s \approx r^i \notin (R_t^{i-1})^*$ for each $l^i \approx s \in \Delta^i$;
- $R_t^i = R_t^{i-1}$ in all other cases.

Finally, let $R_t = R_t^n$; we call R_t the *model fragment for* t, v, Γ_t, and Δ_t. Each clause $C^i = \Gamma^i \rightarrow \Delta^i \vee l^i \approx r^i$ that satisfies the first condition in the above construction is called *generative*, and the clause is said to *generate* the rule $l^i \Rightarrow r^i$ in R_t.

D.1.3 The Properties of the Model Fragment R_t

Lemma 1. *The rewrite system R_t is Church-Rosser.*

Proof. To see that R_t is terminating, simply note that, for each rule $l \Rightarrow r \in R_t$, condition R2 ensures $l >_t r$, and that $>_t$ is a simplification order.

To see that R_t is left-reduced, consider an arbitrary rule $l \Rightarrow r \in R_t$ that is added to R_t in step i of the clause sequence. By condition R3, $l \Rightarrow r$ is irreducible by R_t^i. Now consider an arbitrary rule $l' \Rightarrow r' \in R_t$ that is added to R_t at any step j of the construction where $j > i$. The definition of the clause order implies $l' \approx r' >_t l \approx r$; since $l' >_t r'$ and $l >_t r$ by condition R2, by the definition of the literal order we have $l' \geq_t l$. Since $l \Rightarrow r \in R_t^{j-1}$, condition R3 ensures $l \neq l'$, and so we have $l' >_t l$; consequently, l' is not a subterm of l, and thus l is irreducible by R_t^j. □

Lemma 2. *For each clause $\Gamma \rightarrow \Delta$ such that $\Gamma \rightarrow \Delta \,\hat{\in}\, \mathcal{S}_v$ and $\Gamma\sigma_t \subseteq \Gamma_t$ hold, we have $\Gamma\sigma_t \rightarrow \Delta\sigma_t \,\hat{\in}\, N_t$.*

Proof. Assume that $\Gamma \rightarrow \Delta \,\hat{\in}\, \mathcal{S}_v$ holds. If $\Gamma \rightarrow \Delta$ satisfies condition 1 of Definition 4, then terms s and s' exist such that $s \approx s \in \Delta$ or $\{s \approx s',\ s \not\approx s'\} \subseteq \Delta$; but then, $s\sigma_t \approx s'\sigma_t \in \Delta\sigma_t$ or $\{s\sigma_t \approx s'\sigma_t,\ s\sigma_t \not\approx s'\sigma_t\} \subseteq \Delta\sigma_t$, so we have $\Gamma\sigma_t \rightarrow \Delta\sigma_t \,\hat{\in}\, N_t$. Furthermore, if $\Gamma \rightarrow \Delta$ satisfies condition 2 of Definition 4, then clause $\Gamma' \rightarrow \Delta' \in \mathcal{S}_v$ exists such that $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$; but then, due to $\Gamma'\sigma_t \subseteq \Gamma\sigma_t \subseteq \Gamma_t$, we have that $\Gamma'\sigma_t \rightarrow \Delta'\sigma_t \in N_t$ holds, and therefore $\Gamma\sigma_t \rightarrow \Delta\sigma_t \,\hat{\in}\, N_t$ holds as well. □

Lemma 3. *For each $1 \leq i \leq n$ and each $l \not\approx r \in \Delta^i \vee L^i$, we have $(R_t^{i-1})^* \models l \approx r$ if and only if $R_t^* \models l \approx r$.*

Proof. Consider an arbitrary clause $C^i = \Gamma^i \rightarrow \Delta^i \vee L^i$ and an arbitrary inequality $l \not\approx r \in \Delta^i \vee L^i$. If $l \approx r \in (R_t^{i-1})^*$, then $R_t^{i-1} \subseteq R_t$ implies $l \approx r \in R_t^*$, and so we have $R_t^* \models l \approx r$, as required. Now assume that $l \approx r \notin (R_t^{i-1})^*$. Let l' and r' be the normal forms of l and r, respectively, w.r.t. R_t^{i-1}. Now consider an arbitrary j with $i \leq j \leq n$ such that $l^j \Rightarrow r^j$ is generated by C^j. We then have $l^j \approx r^j >_t l \not\approx r$, which by the definition of literal order implies $l^j >_t l \geq_t l'$ and $l^j >_t r \geq_t r'$; since $>_t$ is a simplification order, l^j is a subterm of neither l' nor r'. Thus, l' and r' are the normal forms of l and r, respectively, w.r.t. R_t^j, and so we have $l' \approx r' \notin (R_t^j)^*$; but then, we have $l \approx r \notin (R_t^j)^*$, as required. □

Lemma 4. *For each generative clause $\Gamma^i \rightarrow \Delta^i \vee l^i \approx r^i$, we have $R_t^* \not\models \Delta^i$.*

Proof. Consider a generative clause $C^i = \Gamma^i \rightarrow \Delta^i \vee l^i \approx r^i$ and a literal $L \in \Delta^i$; condition R1 ensures that $(R_t^{i-1})^* \not\models L$. We next show that $(R_t^{i-1})^* \not\models L$.

Assume that L is of the form $l \not\approx r$. Since $l \not\approx r \in \Delta^i \vee l^i \approx r^i$, by Lemma 3 we have $R_t^* \not\models L$, as required.

Assume that L is of the form $l \approx r$ with $l >_t r$. We show by induction that, for each j with $i \leq j \leq n$, we have $(R_t^j)^* \not\models L$. To this end, we assume that $(R_t^{j-1})^* \not\models L$. If C^j is not generational, then $R_t^j = R_t^{j-1}$, and so $(R_t^j)^* \not\models L$. Thus, assume that C^j is generational. We consider the following two cases.

- $l^j = l$. We have the following two subcases.
  - $j = i$. Condition R4 then ensures $r \approx r^i \notin (R_t^{i-1})^*$. Let r' and r'' be the normal forms of r and r^i, respectively, w.r.t. R_t^{i-1}; we have $r' \approx r'' \notin (R_t^{i-1})^*$. Moreover, $l >_t r \geq_t r'$ and $l >_t r^i \geq_t r''$ hold; since $>_t$ is a simplification order, l is a subterm of neither r' nor r''; therefore, r' and r'' are the normal forms of r and r^i, respectively, w.r.t. R_t^i, and therefore $r' \approx r'' \notin (R_t^i)^*$. Finally, since $l \Rightarrow r^i \in R_t^i$, term r'' is the normal form of l w.r.t. R_t^i, and so $l \approx r \notin (R_t^i)^*$.
  - $j > i$. But then, $l^j \approx r^j >_t l^i \approx r^i >_t l \approx r$ implies $l^j = l^i = l$. Furthermore, C^i is generational, so we have $l^i \Rightarrow r^i \in R_t^{j-1}$. But then, l^j is not irreducible by R_t^{j-1}, which contradicts condition R3.
- $l^j >_t l$. Let l' and r' be the normal forms of l and r, respectively, w.r.t. R_t^{j-1}. Then, we have $l^j >_t l \geq_t l'$ and $l^j >_t r \geq_t r'$; since $>_t$ is a simplification order, l^j is a subterm of neither l' nor r'. Thus, l' and r' are the normal forms of l and r, respectively, w.r.t. R_t^j, and so $l' \approx r' \notin (R_t^j)^*$; hence, $l \approx r \notin (R_t^j)^*$ holds. □

Lemma 5. *Let $\Gamma \rightarrow \Delta$ be a clause with $\Gamma \rightarrow \Delta \hat{\in} N_t$. Then $R_t^* \models \Delta$ holds whenever some i with $1 \leq i \leq n+1$ exists such that*

1. *for each $1 \leq j < i$, we have $R_t^* \models \Delta^j \vee L^j$, and*
2. *$i \leq n$ (i.e., i is an index of a clause from N_t) implies $\Delta^i \vee L^i >_t \Delta$.*

Proof. Assume that $\Gamma \rightarrow \Delta \hat{\in} N_t$ holds. If $\Gamma \rightarrow \Delta$ satisfies condition 1 of Definition 4, then we clearly have $R_t^* \models \Delta$. Assume that $\Gamma \rightarrow \Delta$ satisfies condition 2 of Definition 4 due to some clause $\Gamma^j \rightarrow \Delta^j \vee L^j \in N_t$ such that $\Gamma^j \subseteq \Gamma$ and $\Delta^j \cup \{L^j\} \subseteq \Delta$ hold; the latter clearly implies $\Delta \geq_t \Delta^j \vee L^j$. Let i be an integer satisfying this lemma's assumption. If $i = n+1$, then we clearly have $j < i$; otherwise, $\Delta^i \vee L^i >_t \Delta$ implies $\Delta^i \vee L^i >_t \Delta^j \vee L^j$, and so we also $j < i$. But then, by the lemma assumption we have $R_t^* \models \Delta^j \vee L^j$, which implies $R_t^* \models \Delta$, as required. □

Lemma 6. *For each $\Gamma \to \Delta \in N_t$, we have $R_t^* \models \Delta$.*

Proof. For the sake of a contraction, choose $C^i = \Gamma^i \to \Delta^i \vee L^i$ as the clause in the sequence of clauses from Appendix D.1.2 with the smallest i such that $R_t^* \not\models \Delta^i \vee L^i$; please recall that $L^i >_t \Delta^i$ and that $L^i = l^i \bowtie r^i$ with $\bowtie \in \{\approx, \not\approx\}$. The way we choose i ensures that i satisfies condition 1 of Lemma 5. Now by the definition of N_t, a clause $\Gamma \to \Delta \vee L \in \mathcal{S}_v$ exists such that

$$\Gamma\sigma_t = \Gamma^i \subseteq \Gamma_t, \quad \Delta\sigma_t = \Delta^i, \quad L\sigma_t = L^i, \quad \text{and} \quad \Delta \not\succeq_v L. \tag{70}$$

We next prove the claim of this lemma by considering the possible forms of L^i.

Assume $L^i = l^i \approx r^i$ with $l^i = r^i$. But then, we have $R_t^* \models L^i$, which contradicts our assumption that $R_t^* \not\models \Delta^i \vee L^i$.

Assume $L^i = l^i \approx r^i$ with $l^i >_t r^i$. By the definition of $>_t$, we have $l \succ_v r$. We first show that $(R_t^{i-1})^* \not\models \Delta^i \vee L^i$ holds; towards this goal, note that, for each equality $l \approx r \in \Delta^i \vee L^i$, properties $R_t^* \not\models l \approx r$ and $R_t^{i-1} \subseteq R_t$ imply $(R_t^{i-1})^* \not\models l \approx r$; and for each inequality $l \not\approx r \in \Delta^i$, Lemma 3 and $R_t^* \not\models l \not\approx r$ imply $(R_t^{i-1})^* \not\models l \not\approx r$. Thus, clause C^i satisfies conditions R1 and R2; however, since $R_t^* \not\models l^i \approx r^i$, clause C^i is not generational and thus either condition R3 or condition R4 are not satisfied. We next consider each of these two possibilities.

- Condition R3 does not hold—that is, l^i is reducible by R_t^{i-1}. By the definition of reducibility, a position p and a generative clause $C^j = \Gamma^j \to \Delta^j \vee l^j \approx r^j$ exist such that $j < i$ and $l^i|_p = l^j$. Due to $j < i$, we have $l^i \approx r^i \geq_t l^j \approx r^j$; together with $l^j \approx r^j >_t \Delta^j$, we have $l^i \approx r^i >_t \Delta^j$. Lemma 4 ensures $R_t^* \not\models \Delta^j$, and the definition of N_t ensures that a clause $\Gamma' \to \Delta' \vee l' \approx r' \in \mathcal{S}_v$ exists such that

$$\begin{array}{llll} \Gamma'\sigma_t = \Gamma^j \subseteq \Gamma_t, & \Delta'\sigma_t = \Delta^j, & l'\sigma_t = l^j, & r'\sigma_t = r^j, \\ \Delta' \not\succeq_v l' \approx r' & \text{and} \quad l' \succ_v r'. & & \end{array} \tag{71}$$

  By the assumption of Theorem 2, the Eq rule is not applicable to (70) and (71), and so $\Gamma \wedge \Gamma' \to \Delta \vee \Delta' \vee l[r']_p \approx r \mathrel{\hat{\in}} \mathcal{S}_v$. Let $\Delta'' = \Delta^i \vee \Delta^j \vee l^i[r^j]_p \approx r^i$. Then clearly $\Gamma\sigma_t \cup \Gamma'\sigma_t \subseteq \Gamma_t$, so Lemma 2 ensures that $\Gamma^i \wedge \Gamma^j \to \Delta'' \mathrel{\hat{\in}} N_t$ holds. Set R_t^* is a congruence, so $l^i[r^j]_p \approx r^i \notin R_t^*$ holds, and therefore $R_t^* \not\models \Delta''$ holds. Finally, $>_t$ is a simplification order, which ensures $l^i \approx r^i >_t l^i[r^j]_p \approx r^i$; together with $l^i \approx r^i >_t \Delta^i$ and $l^i \approx r^i >_t \Delta^j$, we have $l^i \approx r^i >_t \Delta''$. But then, Lemma 5 implies $R_t^* \models \Delta''$, which is a contradiction.
- Condition R4 does not hold. Then, some term s exists such that $l^i \approx s \in \Delta^i$ and $s \approx r^i \in (R_t^{i-1})^*$. Due to $R_t^{i-1} \subseteq R_t$, we have $s \approx r^i \in R_t^*$, and so $R_t^* \not\models s \not\approx r^i$. Furthermore, $\Delta \vee L$ is of the form $\Delta' \vee l \approx r \vee l' \approx r'$ such that

$$l\sigma_t = l^i, \quad r\sigma_t = s, \quad l'\sigma_t = l^i, \quad \text{and} \quad r'\sigma_t = r^i. \tag{72}$$

  But then, we clearly have $l' = l$. By the assumption of Theorem 2, the Factor rule is not applicable to $\Gamma \to \Delta \vee L$, and so we have $\Gamma \to \Delta' \vee r \not\approx r' \vee l' \approx r' \mathrel{\hat{\in}} \mathcal{S}_v$. Let $\Delta'' = \Delta'\sigma_t \vee s \not\approx r^i \vee l^i \approx r^i$. But then, $\Gamma\sigma_t \subseteq \Gamma_t$ and Lemma 2 ensure that $\Gamma^i \to \Delta'' \mathrel{\hat{\in}} N_t$ holds. By all the previous observations, we have $R_t^* \not\models \Delta''$. Moreover, $l^i >_t r^i$ and $l^i >_t s$ imply $l^i \approx r^i >_t s \approx r^i$; thus, $\Delta^i \vee l^i \approx r^i >_t \Delta''$ holds. But then, Lemma 5 implies $R_t^* \models \Delta''$, which is a contradiction.

Assume $L^i = l^i \not\approx r^i$ with $l^i = r^i$. Then, literal L is of the form $l \not\approx r$ such that $l\sigma_t \not\approx r\sigma_t = l^i \not\approx r^i$. But then, $l^i = r^i$ implies $l = r$. By the assumption of Theorem 2, the Ineq rule is not applicable to clause $\Gamma \rightarrow \Delta \vee L$, and so we have $\Gamma \rightarrow \Delta \,\hat{\in}\, \mathcal{S}_v$. Since $\Gamma\sigma_t \subseteq \Gamma_t$, by Lemma 2 we have $\Gamma^i \rightarrow \Delta^i \,\hat{\in}\, N_t$. Clearly, $\Delta^i \vee l^i \not\approx r^i >_t \Delta^i$, and so Lemma 5 implies $R_t^* \models \Delta^i$, which is a contradiction.

Assume $L^i = l^i \not\approx r^i$ with $l^i >_t r^i$. Lemma 3 ensures $(R_t^{i-1})^* \not\models l^i \not\approx r^i$; hence, l^i is reducible by R_t^{i-1} so, by the definition of reducibility, a position p and a generative clause $C^j = \Gamma^j \rightarrow \Delta^j \vee l^j \approx r^j$ exist such that $j < i$ and $l^i|_p = l^j$. Due to $j < i$, we have $l^i \not\approx r^i >_t l^j \approx r^j >_t \Delta^j$. Lemma 4 ensures $R_t^* \not\models \Delta^j$, and the definition of N_t ensures that a clause $\Gamma' \rightarrow \Delta' \vee l' \approx r' \in \mathcal{S}_v$ exists satisfying (71), as in the first case. By the assumption of Theorem 2, the Eq rule is not applicable to clauses (70) and (71), and so $\Gamma \wedge \Gamma' \rightarrow \Delta \vee \Delta' \vee l[r']_p \not\approx r \,\hat{\in}\, \mathcal{S}_v$ holds. Let $\Delta'' = \Delta^i \vee \Delta^j \vee l^i[r^j]_p \not\approx r^i$. We clearly have $\Gamma\sigma_t \cup \Gamma'\sigma_t \subseteq \Gamma_t$, so by Lemma 2 we have $\Gamma^i \wedge \Gamma^j \rightarrow \Delta'' \,\hat{\in}\, N_t$. Since R_t^* is a congruence, we have $R_t^* \not\models l^i[l^j]_p \not\approx r^i$, and therefore $R_t^* \not\models \Delta''$ holds. Finally, $>_t$ is a simplification order, so $l^i \not\approx r^i >_t l^i[l^j]_p$; together with $l^i \approx r^i >_t \Delta^i$ and $l^i \approx r^i >_t \Delta^j$, we have $l^i \approx r^i >_t \Delta''$. But then, Lemma 5 implies $R_t^* \models \Delta''$, which is a contradiction. □

Lemma 7. *For each clause $\Gamma \rightarrow \Delta$ with $\Gamma \rightarrow \Delta \,\hat{\in}\, N_t$, we have $R_t^* \models \Delta$.*

Proof. Apply Lemma 5 for $i = n + 1$ and Lemma 6. □

Lemma 8. $R_t^* \not\models \Gamma_t \rightarrow \Delta_t$.

Proof. For $R_t^* \models \Gamma_t$, note that condition L2 ensures $\Gamma_t \rightarrow A \,\hat{\in}\, N_t$, and so Lemma 7 ensures $R_t^* \models A$ for each atom $A \in \Gamma_t$.

For $R_t^* \not\models \Delta_t$, assume for the sake of a contradiction that an atom $A \in \Delta_t$ exists such that $R_t^* \models A$. Then, a generative clause $C^i = \Gamma^i \rightarrow \Delta^i \vee l^i \approx r^i \in N_t$ and a position p exist such that $A|_p = l^i$; let $\Delta = \Delta^i \vee l^i \approx r^i$. Since $>_t$ is a simplification order and $l^i >_t r^i$, we have $A \geq_t l^i \approx r^i$; but then, since $l^i \approx r^i >_t \Delta^i$, we have $A \geq_t \Delta$. We next consider an arbitrary literal $l \bowtie r \in \Delta$ with $\bowtie \in \{\approx, \not\approx\}$ and $l \geq_t r$. Literal $l \bowtie r$ is obtained by grounding a context literal by σ_t, and so l cannot be t or t'. Moreover, $A \geq_t l \bowtie r$ implies $A \geq_t l$; but then, by condition O2 we have $l \approx \mathsf{t} \in \Delta_t$ and $r = \mathsf{t}$. We thus have $\Delta \subseteq \Delta_t$; but then, $\Gamma^i \subseteq \Gamma_t$ implies that $\Gamma_t \rightarrow \Delta_t \,\hat{\in}\, N_t$ holds, which contradicts condition L1. □

D.2 Interpreting the Ontology $\mathcal{O}$

We now show how to combine the rewrite systems R_t from Appendix D.1 into a single rewrite system R such that $R^* \models \mathcal{O}$ and $R^* \not\models \Gamma_Q \rightarrow \Delta_Q$.

D.2.1 Unfolding the Context Structure

We construct R by a partial induction over the terms in $\mathcal{T}$. We define several partial functions: function X maps a term t to a context $X_t \in \mathcal{V}$; functions Γ and Δ assign to a term t a conjunction Γ_t and a disjunction Δ_t, respectively, of atoms; and function R maps each term into a model fragment R_t for t, X_t, Γ_t, and Δ_t.

M1. For the base case, we consider the constant c.

$$X_c = q \tag{73}$$
$$\Gamma_c = \Gamma_Q \sigma_c \tag{74}$$
$$\Delta_c = \Delta_Q \sigma_c \tag{75}$$
$$R_c = \text{the model fragment for } c, q, \Gamma_c, \text{ and } \Delta_c \tag{76}$$

M2. For the inductive step, assume that $X_{t'}$ has already been defined, and consider an arbitrary function symbol $f \in \mathcal{F}$ such that $f(t')$ is $R_{t'}$-irreducible. Let $u = X_{t'}$ and $t = f(t')$. We have two possibilities.

M2.a. Term t occurs in $R_{t'}$. Then, term $t = f(t')$ was generated in $R_{t'}$ by some ground clause $C = \Gamma \rightarrow \Delta \vee L \in N_{t'}$ such that $L >_t \Delta$ and $f(t')$ occurs in L. By the definition of N_t, then a clause $C' = \Gamma' \rightarrow \Delta' \vee L' \in \mathcal{S}_u$ exists such that $C = C'\sigma_{t'}$ and L' contains $f(x)$; moreover, $L >_{t'} \Delta$ implies $\Delta' \not\succeq_u L'$. The Succ and Core rules are not applicable to $\mathcal{D}$, so we can choose a context $v \in \mathcal{V}$ such that $\langle u, v, f \rangle \in \mathcal{E}$ and $A \rightarrow A \mathrel{\hat{\in}} \mathcal{S}_v$ for each $A \in K_2$, where K_2 is as in the Succ rule. We define the following:

$$X_t = v \tag{77}$$
$$\Gamma_t = R^*_{t'} \cap \mathsf{Su}_t \tag{78}$$
$$\Delta_t = \mathsf{Pr}_t \setminus R^*_{t'} \tag{79}$$
$$R_t = \text{the model fragment for } t, v, \Gamma_t, \text{ and } \Delta_t \tag{80}$$

M2.b. Term t does not occur in $R_{t'}$. Then, let $R_t = \{t \Rightarrow c\}$, and we do not define any other functions for t.

Finally, let R be the rewrite system defined by $R = \bigcup R_t$.

Lemma 9. *The model fragments R_c and R_t constructed in lines* (76) *and* (80) *satisfy conditions L1 through L3 in Appendix D.1.*

Proof. The proof is by induction on the structure of terms $t \in \mathsf{dom}(X)$. For $t = c$, conditions L1 through L3 hold directly from conditions C1 through C3 of Theorem 2. We next assume that the lemma holds for some term $t' \in \mathsf{dom}(X)$, and we consider an arbitrary term t of the form $t = f(t')$; let $u = X_t$ and $v = X_{t'}$. Condition L2 holds because $>_t$ is obtained by grounding a context order $\succ_t$ that satisfies condition 5 of Definition 3. Condition L3 holds by the way u is chosen in condition M2 and the fact that $K_2\sigma_t \subseteq \mathsf{Su}_t$. We next show that condition L1 holds.

For the sake of a contradiction, assume that $\Gamma_t \rightarrow \Delta_t \mathrel{\hat{\in}} N_t$ holds. Set Su_t contains only atoms and we have $\Gamma_t \subseteq \mathsf{Su}_t$ due to (79); therefore, $\Gamma_t \rightarrow \Delta_t \mathrel{\hat{\in}} N_t$ necessarily holds due to condition 2 of Definition 4, and therefore set N_t contains a clause

$$\bigwedge_{i=1}^{m} A_i \rightarrow \bigvee_{i=m+1}^{m+n} A_i \quad \text{with } \{ A_i \mid 1 \leq i \leq m \} \subseteq \Gamma_t \text{ and } \{ A_i \mid m+1 \leq i \leq m+n \} \subseteq \Delta_t \subseteq \mathsf{Pr}_t. \tag{81}$$

By the definition of N_t, set $\mathcal{S}_v$ contains a clause

$$\bigwedge_{i=1}^{m} A'_i \rightarrow \bigvee_{i=m+1}^{m+n} A'_i \quad \text{where} \quad A_i = A'_i\sigma_t \text{ and } A'_i \in \mathsf{Pr}(\mathcal{O}) \text{ for each } i. \tag{82}$$

Consider an arbitrary atom A_i with $1 \leq i \leq m$. By condition M2, terms t and t' are irreducible by $R_{t'}$; but then, since $\Gamma_t \subseteq R^*_{t'}$ holds by (78), atom A_i was generated in $R_{t'}$ by a generative clause (where subscript i does not necessarily indicate the position of the clause in sequence of clauses from Appendix D.1.2)

$$\Gamma_i \to \Delta_i \vee A_i \quad \text{with} \quad A_i >_t \Delta_i. \tag{83}$$

By the definition of $N_{t'}$, set $\mathcal{S}_u$ contains a clause

$$\Gamma'_i \to \Delta'_i \vee A'_i \quad \text{where } \Gamma_i = \Gamma'_i\sigma_t,\ \Delta_i = \Delta'_i\sigma_t, \text{ and } \Delta'_i \not\succeq vA'_i \text{ for } 1 \leq i \leq m. \tag{84}$$

The Pred rule is not applicable to (82) and (84) so (85) holds; together with Lemma 2, this ensures ensures (86).

$$\bigwedge_{i=1}^{m} \Gamma'_i \to \bigvee_{i=1}^{m} \Delta'_i \vee \bigvee_{i=m+1}^{m+n} A'_i\sigma \mathrel{\hat{\in}} \mathcal{S}_u \quad \text{for } \sigma = \{x \mapsto f(x),\ y \mapsto x\} \tag{85}$$

$$\bigwedge_{i=1}^{m} \Gamma_i \to \bigvee_{i=1}^{m} \Delta_i \vee \bigvee_{i=m+1}^{m+n} A_i \mathrel{\hat{\in}} N_{t'} \tag{86}$$

By Lemma 4, we have $R^*_{t'} \not\models \Delta_i$; and (79) ensures that $R^*_{t'} \not\models \Delta_t$, and so $R^*_{t'} \not\models A_i$ for each $m+1 \leq i \leq m+n$; however, this contradicts (86) and Lemma 7. □

D.2.2 Termination, Confluence, and Compatibility

Lemma 10. *The rewrite system R is Church-Rosser.*

Proof. We show that R is terminating and left-reduced, and thus Church-Rosser. In the proof of the former, we use a total simplification order $\rhd$ on all ground $\mathcal{F}$- and $\mathcal{P}$-terms defined as follows. We extend the precedence $\gtrdot$ from Definition 3 to all $\mathcal{F}$- and $\mathcal{P}$-symbols in an arbitrary way, but ensuring that constant t is smallest in the order; then, let $\rhd$ be a *lexicographic path order* [4] over such $\gtrdot$. It is well known that such $\rhd$ is a simplification order, and that it satisfies the following properties for each $\mathcal{F}$-term t with predecessor t' (if one exists), all function symbols $f, g \in \mathcal{F}$, and each $\mathcal{P}$-term A:

- $f(t) \rhd t \rhd t'$,
- $f \gtrdot g$ implies $f(t) \rhd g(t)$, and
- $A \rhd \mathsf{t}$.

Thus, conditions 1 and 2 of Definition 3 and the manner in which context orders are grounded in Appendix D.1.1 clearly ensure that, for each $\mathcal{F}$-term $t \in \mathsf{dom}(X)$ and for all terms s_1 and s_2 from the $\mathcal{F}$-neighbourhood of t with $s_1 >_t s_2$, we have $s_1 \rhd s_2$.

We next show that R is terminating by arguing that each rule in R is embedded in $\rhd$. To this end, consider an arbitrary rule $l \Rightarrow r \in R$. Clearly, a term $t \in \mathsf{dom}(R)$ exists such that $l \Rightarrow r \in R_t$. This rule is obtained from a head $l \approx r$ of a clause in N_t, and condition R2 of the definition of R_t ensures that $l >_t r$. Moreover, $l \approx r$ is obtained by grounding a context literal with σ_t, so we have the following possible forms of $l \approx r$.

- Terms l and r are both from the $\mathcal{F}$-neighbourhood of t. Then, $l >_t r$ implies $l \rhd r$.
- Terms $l \approx r = A \approx \mathsf{t}$ for A a $\mathcal{P}$-term. Then, $A \rhd \mathsf{t}$ since t is smallest in $\gtrdot$.

We next show that R is left-reduced. For the sake of a contradiction, assume that a rule $l \Rightarrow r \in R$ exists such that l is reducible by $R' = R \setminus \{l \Rightarrow r\}$. Let p be the 'deepest' position at which some rule in R' reduces l (i.e., no rule in R' reduces l at position below p), and let $l' \Rightarrow r' \in R'$ be the rule that reduces l at position p; thus, $l' = l|_p$. By the definition of R, we have $l' \Rightarrow r' \in R_t$ where t can be as follows.

- Term t is handled in condition M2.a. Then $l' \Rightarrow r'$ is generated by an equality $l' \approx r'$ in the head of a generative clause, and so l' is of the form $f(t)$. Thus, $f(t)$ is reducible by R_t, which contradicts condition M2 from the construction of R.
- Term t is handled in condition M2.b. Then $l' = t$; moreover, R' does not contain t by the construction of R, which contradicts the assumption that $l' \Rightarrow r' \in R'$. □

Lemma 11. *For each term t, each $f \in \mathcal{F}$, and each atom $A \in \mathsf{Su}_t \cup \mathsf{Pr}_{f(t)}$ such that $A \in R^*$ and all $\mathcal{F}$-terms in A are irreducible by R, we have $A \in R^*_t$.*

Proof. Let t be a term, let $f \in \mathcal{F}$ be a function symbol, and let $A \in \mathsf{Su}_t \cup \mathsf{Pr}_{f(t)}$ be an atom such that all $\mathcal{F}$-terms in A are irreducible by R; the latter ensures $A \Rightarrow \mathsf{t} \in R$. We next consider the possible forms of A.

Assume $A \in \mathsf{Su}_t$. By the definition of Su_t in (68) and the fact that $\mathsf{Su}(\mathcal{O})$ contains only atoms of the form $A(x)$, $S(x,y)$, and $S(y,x)$, atom A can be of the form $A(t)$, $S(t,t')$, or $S(t',t)$, for t' the predecessor of t (if one exists). By the form of the generative clauses, we clearly have $A \in R^*_t$ or $A \in R^*_{t'}$. Now assume $A \in R^*_{t'}$. Due to $A \in \mathsf{Su}_t$ and the definition of Γ_t in (78), we have $A \in \Gamma_t$. Lemma 8 ensures that $R^*_t \not\models \Gamma_t \rightarrow \Delta_t$. But then, we have $A \in R^*_t$, as required.

Assume $A \in \mathsf{Pr}_{f(t)}$. By the definition of $\mathsf{Pr}_{f(t)}$ in (69) and the fact that $\mathsf{Pr}(\mathcal{O})$ contains only atoms of the form $A(y)$, $S(y,x)$, and $S(x,y)$, atom A can be of the form $A(t)$, $S(t,f(t))$, or $S(f(t),t)$. By the form of the generative clauses, we clearly have $A \in R^*_t$ or $A \in R^*_{f(t)}$. Assume for the sake of a contradiction that $A \notin R^*_t$, but $A \in R^*_{f(t)}$. Due to $A \in \mathsf{Pr}_{f(t)}$ and the definition of $\Delta_{f(t)}$ in (79), we have $A \in \Delta_{f(t)}$; due to Lemma 8, we have $R^*_{f(t)} \not\models \Gamma_{f(t)} \rightarrow \Delta_{f(t)}$; therefore, we have $A \notin R^*_{f(t)}$, which is a contradiction. □

Lemma 12. *Let s_1 and s_2 be DL-terms, and let τ be a substitution irreducible by R such that $s_1\tau$ and $s_2\tau$ are ground and each $\tau(z_i)$ (if defined) is in the $\mathcal{F}$-neighbourhood of $\tau(x)$. Then, for $\bowtie \in \{\approx, \not\approx\}$, if $R^*_{\tau(x)} \models s_1\tau \bowtie s_2\tau$, then $R^* \models s_1\tau \bowtie s_2\tau$.*

Proof. Let s_1 and s_2 and τ be as stated above, let $t = \tau(x)$, and let t' be the predecessor of t (if one exists). Since t is irreducible by R, rewrite system R_t has been defined in Appendix D.2.1. We next consider the possible forms of $\bowtie$.

- Assume $\bowtie = \approx$. But then, $R_t \subseteq R$ and $R^*_t \models s_1\tau \approx s_2\tau$ imply $R^* \models s_1\tau \approx s_2\tau$.
- Assume $\bowtie = \not\approx$. Let s'_1 and s'_2 be the normal forms of $s_1\tau$ and $s_2\tau$, respectively, w.r.t. R_t. Due to the shape of DL-literals, s_1 and s_2 can be of the form $f(x)$ or z_i; therefore, $s_1\tau$ and $s_2\tau$ are of the form $f(t)$ or t'. Term t is irreducible by R, and therefore t' is irreducible by R as well. Furthermore, due to the shape of context terms, the only rewrite system where $f(t)$ could occur on the left-hand side of a rewrite rule is R_t. Consequently, $f(t)$ is irreducible by R as well. But then, s'_1 and s'_2 are the normal forms of $s_1\tau$ and $s_2\tau$, respectively, w.r.t. R; thus, $R^* \models s'_1 \not\approx s'_2$, and therefore we have $R^* \models s_1\tau \not\approx s_2\tau$, as required. □

D.2.3 The Completeness Claim

Lemma 13. *For each DL-clause $\Gamma \rightarrow \Delta \in \mathcal{O}$, we have $R^* \models \Gamma \rightarrow \Delta$.*

Proof. Consider an arbitrary DL-clause $\Gamma \rightarrow \Delta \in \mathcal{O}$ of the following form:

$$\textstyle\bigwedge_{i=1}^{n} A_i \rightarrow \Delta \tag{87}$$

Let τ' be an arbitrary substitution such that $\Gamma\tau' \rightarrow \Delta\tau'$ is ground, and let τ be the substitution obtained from τ' by replacing each ground term with its normal form w.r.t. R. Since R^* is a congruence, we have $R^* \models \Gamma\tau' \rightarrow \Delta\tau'$ if and only if $R^* \models \Gamma\tau \rightarrow \Delta\tau$. We next assume that $R^* \models \Gamma\tau$, and we show that $R^* \models \Delta\tau$ holds as well.

Consider an arbitrary atom $A_i \in \Gamma$. By the definition of DL-clauses, A_i is of the form $B(x)$, $S(x, z_j)$, or $S(z_j, x)$. Substitution τ is irreducible by R, and so all $\mathcal{F}$-terms in $A\tau$ are irreducible by R; but then, $A_i\tau \in R^*$ clearly implies $A_i\tau \Rightarrow \mathsf{t} \in R$. Each such rule is obtained from a generative clause so $A_i\tau$ is of the form $B(t)$, $S(t, f(t))$, $S(f(t), t)$, $S(t, t')$, or $S(t', t)$, where $t = \tau(x)$ and t' is the predecessor of t (if it exists). We next prove that $A_i\tau \in \mathsf{Su}_t \cup \mathsf{Pr}_{f(t)}$ holds by considering the possible forms of A_i.

- $A_i = B(x)$, so $A_i\tau = B(t)$. Thus, $B(x) \in \mathsf{Su}(\mathcal{O})$ and so $B(t) \in \mathsf{Su}_t$.
- $A_i = S(x, z_j)$, so $A_i\tau$ is of the form $S(t, t')$ or $S(t, f(t))$. Thus, $S(x, y) \in \mathsf{Su}(\mathcal{O})$ and so $S(t, t') \in \mathsf{Su}_t$, and $R(y, x) \in \mathsf{Pr}(\mathcal{O})$ and so $S(t, f(t)) \in \mathsf{Pr}_{f(t)}$.
- $A_i = S(z_j, x)$, so $A_i\tau$ is of the form $S(t', t)$ or $S(f(t), t)$. Thus, $S(y, x) \in \mathsf{Su}(\mathcal{O})$ and so $S(t', t) \in \mathsf{Su}_t$, and $R(x, y) \in \mathsf{Pr}(\mathcal{O})$ and so $S(f(t), t) \in \mathsf{Pr}_{f(t)}$.

Lemma 11 then implies $A_i\tau \in R_t$, and so N_t contains a generative clause of the form (88). Now let $v = X_t$; by the definition of N_t, set $\mathcal{S}_v$ contains a clause of the form (89).

$$\Gamma_i \rightarrow \Delta_i \vee A_i \text{ with } A_i >_t \Delta_i \text{ and } \Gamma_i \subseteq \Gamma_t \tag{88}$$

$$\Gamma_i' \rightarrow \Delta_i' \vee A_i' \text{ with } \Delta_i' \not\succeq vA_i' \text{ and } \Gamma_i'\sigma_t = \Gamma_i,\ \Delta_i'\sigma_t = \Delta_i, \text{ and } A_i'\sigma_t = A_i \tag{89}$$

The Hyper rule is not applicable to (87) and (89), and therefore (90) holds, where σ is the substitution obtained from τ by replacing each occurrence of t (possibly nested in another term) with x. Finally, Lemma 2 ensures that (91) holds as well.

$$\textstyle\bigwedge_{i=1}^{n} \Gamma_i' \rightarrow \Delta\sigma \vee \bigvee_{i=1}^{n} \Delta_i' \,\hat{\in}\, \mathcal{S}_v \tag{90}$$

$$\textstyle\bigwedge_{i=1}^{n} \Gamma_i \rightarrow \Delta\tau \vee \bigvee_{i=1}^{n} \Delta_i \,\hat{\in}\, N_t \tag{91}$$

Now (91) and Lemma 7 imply $R_t^* \models \Delta\tau \vee \bigvee_{i=1}^{n} \Delta_i$, but Lemma 4 implies $R_t^* \not\models \Delta_i$; therefore, we have $R_t^* \models \Delta\tau$. Finally, Lemma 12 ensures $R^* \models \Delta\tau$, as required. □

Lemma 14. $R^* \not\models \Gamma_Q \rightarrow \Delta_Q$.

Proof. The claim clearly follows from $R^* \not\models \Gamma_c \rightarrow \Delta_c$. Note that Lemma 8 ensures $R_c^* \not\models \Gamma_c \rightarrow \Delta_c$; thus, $R_c^* \models \Gamma_c$ and $R_c^* \not\models \Delta_c$. The former observation and Lemma 12 ensure that $R^* \models \Gamma_c$ holds. Moreover, for each atom $B(x) \in \Delta_Q$, Definition 2 ensures $B(y) \in \mathsf{Pr}(\mathcal{O})$; thus, for each $f \in \mathcal{F}$, we have $B(c) \in \mathsf{Pr}_{f(c)}$, and so the contrapositive of Lemma 11 ensures $R^* \not\models B(c)$. Thus, $R^* \not\models \Delta_c$ holds, as required. □