Fundamenta Informaticae ??? (200?) 1001–1024
IOS Press

On the Relationship between Description Logic-based and F-Logic-based Ontologies[*]

Jos de Bruijn[†]
Faculty of Computer Science
Free University of Bozen-Bolzano, Piazza Domenicani 3, Bolzano, Italy
debruijn@inf.unibz.it

Stijn Heymans
Digital Enterprise Research Institute (DERI)
University of Innsbruck, Technikerstrasse 21a, Innsbruck, Austria
stijn.heymans@deri.org

Abstract. Many popular ontology languages are based on (subsets of) first-order predicate logic, with classes represented by unary predicates and properties by binary predicates. Specifically, the Semantic Web ontology language OWL DL is based on the Description Logic $\mathcal{SHOIN}$. F-Logic is an ontology language which is also based on first-order logic, but classes and properties are modeled as terms rather than predicates. Our goal is to enable interoperation between predicate-based and F-Logic-based ontology languages. To this end, we define an intuitive translation from predicate-based ontologies to F-Logic ontologies and show that this translation preserves entailment for large classes of ontology languages, including most of OWL DL. Specifically, we define the class of equality-safe ($\mathcal{E}$-safe) formulas, show that the Description Logic $\mathcal{SHIQ}$ is $\mathcal{E}$-safe, and show that the translation preserves validity of $\mathcal{E}$-safe formulas. We use these results to close the open problem of layering F-Logic programming on top of Description Logic Programs and we show that our results apply to HILOG, a syntactically higher-order, but semantically first-order language. Finally, we show that our results also apply to a meta-modeling extension for Description Logics (υ-semantics).

Keywords: frame logic, ontology translation, ontology meta-modeling, equality-safe formulas, semantic web language layering, f-logic programming

[*]This work was partly supported by the European Commission under the projects Knowledge Web (IST-2004-507482) and DIP (FP6-507483).
[†]Address for correspondence: Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100, Bolzano, Italy

1. Introduction

There have been several proposals for using F-Logic [21, 20] as the basis for an ontology language for the Semantic Web, as well as extending current Semantic Web languages with F-Logic-based rules [19, 10, 2, 6]. In F-Logic, classes and properties are interpreted as objects. This may hamper interoperation with Description Logic-based ontology languages (e.g. OWL DL [12]), in which classes and properties are interpreted as unary and binary predicates, respectively. We call ontologies in F-Logic "frame-based ontologies" and ontologies in Description Logics "predicate-based ontologies".

Statements such as "John is a Person" and "John has-father Jack" are represented in F-Logic using statements of the forms $John\!:\!Person$ and $John[has\text{-}father \twoheadrightarrow Jack]$, where : and $\twoheadrightarrow$ are F-Logic language constructs, and $John, Person, has\text{-}father, Jack$ are terms. In Description Logics, and, in general, predicate-based ontologies, these statements are represented as $Person(John)$ and $has\text{-}father(John, Jack)$, respectively, where $John$ and $Jack$ are terms, as before, but $Person$ and $has\text{-}father$ are predicate symbols. It is an open problem whether, and to what extent, the F-Logic representation of the statements can be seen as equivalent to the Description Logic representation, especially in the presence of equality statements, since equality of terms in F-Logic entails equality of their class extensions as well.

In this paper we define a straightforward translation from predicate-based ontologies to F-Logic. We show that when considering a sorted variant of F-Logic, the translation preserves entailment for arbitrary first-order theories. We then show that this is not the case in general when translating the ontology to an unsorted F-Logic language. However, for certain classes of first-order formulas, namely *cardinal* formulas [11], the translation preserves validity. A disadvantage of this class is that its definition is semantic, i.e. it is not possible to determine whether a given formula is cardinal based on its structure. There are known classes of cardinal formulas (e.g. equality-free formulas). However, these classes are not expressive enough to capture ontology languages with counting quantifiers or equality.

We define the novel class of equality-safe ($\mathcal{E}$-safe) formulas, show that axioms of the Description Logic $\mathcal{SHIQ}$ can be equivalently rewritten to $\mathcal{E}$-safe formulas, and show that $\mathcal{E}$-safe formulas are cardinal. Finally, $\mathcal{E}$-safe formulas are closed under negation, and thus entailment of $\mathcal{E}$-safe formulas can be reduced to checking validity. Using these results, we show that our translation preserves entailment for large classes of ontology languages which include equality, such as $\mathcal{SHIQ}$.

We use these results to close the open problem of F-Logic extensions of Description Logic Programs (DLP) [16]. We define F-Logic programming, with negation based on the stable model semantics [14], and show that this is indeed an extension of DLP.

We then show that our main result about $\mathcal{E}$-safe formulas also applies to HILOG [11], a syntactically higher-order, but semantically first-order logical language. Finally, we show that our results apply to a meta-modeling extension of Description Logics considered in [24], called v-entailment.

Structure of the paper In Section 2 we review predicate- and frame-based ontology modeling languages. In Section 3, we show that the translation of any predicate-based ontology to sorted F-Logic is faithful and that the translation of *cardinal* formulas to unsorted F-Logic is faithful; we identify the class of $\mathcal{E}$-safe formulas and demonstrate cardinality. We define the notion of F-Logic programming and show that the straightforward F-Logic extension of DLP preserves ground entailment in Section 4. We then show that our results apply to HILOG and Description Logics with meta-modeling in Section 5. Finally, we review related work and present conclusions in the Sections 6 and 7.

This paper extends [8] with a novel definition of F-Logic programming and a more in-depth treatment of HILOG and meta-modeling extensions of Description Logics.

2. Preliminaries

First-Order Logic and Description Logics The signature of a first-order language $\mathcal{L}$ is of the form $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$, where $\mathcal{F}$ and $\mathcal{P}$ are disjoint countable sets of function and predicate symbols and each function or predicate symbol has an associated arity n, which is a nonnegative integer. Function symbols with arity 0 are also called *constants*.

Given a signature Σ and a set of variable symbols $\mathcal{V}$, terms are either variables or constructed terms of the form $f(t_1, \dots, t_n)$ with $f \in \mathcal{F}$ an n-ary function symbol ($n \geq 0$) and $t_1, \dots, t_n$ terms. Atomic formulas are expressions of the forms $p(t_1, \dots, t_n)$ and $t_1 = t_2$, with $p \in \mathcal{P}$ an n-ary predicate symbol ($n \geq 0$) and $t_1, \dots, t_n$ terms. Formulas of a first-order language $\mathcal{L}$ are constructed as usual: every atomic formula is a formula of $\mathcal{L}$; compound formulas are constructed using atomic formulas, the logical connectives $\neg, \wedge, \vee, \supset$, the quantifiers $\exists, \forall$, and the auxiliary symbols ')','('. A sentence is a formula with no free occurrences of variables. A theory $\Phi \subseteq \mathcal{L}$ is a finite set of sentences.

An *interpretation* of a language $\mathcal{L}$ is a tuple $\mathcal{I} = \langle U, \cdot^I \rangle$, where U is a countable nonempty set (called *domain*) and $\cdot^I$ is a mapping which assigns: a function $f^I : U^n \to U$ to every n-ary function symbol $f \in \mathcal{F}$ and a relation $p^I \subseteq U^n$ to every n-ary predicate symbol $p \in \mathcal{P}$. A variable assignment B is a mapping which assigns an element $x^B \in U$ to every variable symbol x. A variable assignment B' is an x-variant of B if $y^B = y^{B'}$ for every variable $y \in \mathcal{V}$ with $y \neq x$. A *variable substitution* β is a mapping from variables to ground terms.

Given an interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$, a variable assignment B, and a term t, $t^{\mathcal{I},B}$ is defined as: $x^{\mathcal{I},B} = x^B$ for variable symbol x and $t^{\mathcal{I},B} = f^I(t_1^{\mathcal{I},B}, \dots, t_n^{\mathcal{I},B})$ for t of the form $f(t_1, \dots, t_n)$.

An interpretation $\mathcal{I}$ *satisfies* an atomic formula $p(t_1, \dots, t_n)$, given a variable assignment B, denoted $(\mathcal{I}, B) \models p(t_1, \dots, t_n)$, if $(t_1^{\mathcal{I},B}, \dots, t_n^{\mathcal{I},B}) \in p^I$. $(\mathcal{I}, B) \models t_1 = t_2$ iff $t_1^{\mathcal{I},B} = t_2^{\mathcal{I},B}$. This extends to arbitrary formulas as usual: $(\mathcal{I}, B) \models \neg\phi_1$ iff $(\mathcal{I}, B) \not\models \phi_1$; $(\mathcal{I}, B) \models \phi_1 \wedge \phi_2$ (resp. $(\mathcal{I}, B) \models \phi_1 \vee \phi_2$) iff $(\mathcal{I}, B) \models \phi_1$ and $(\mathcal{I}, B) \models \phi_2$ (resp. $(\mathcal{I}, B) \models \phi_1$ or $(\mathcal{I}, B) \models \phi_2$); $(\mathcal{I}, B) \models \forall x(\phi_1)$ (resp. $(\mathcal{I}, B) \models \exists x(\phi_1)$) iff for every (resp. for some) B' which is an x-variant of B, $(\mathcal{I}, B') \models \phi_1$.

An interpretation $\mathcal{I}$ is a *model* of ϕ, denoted $\mathcal{I} \models \phi$, if $(\mathcal{I}, B) \models \phi$ for every variable assignment B; ϕ is satisfiable if it has a model (unsatisfiable otherwise); ϕ is valid if every interpretation $\mathcal{I}$ is a model of ϕ. These definitions straightforwardly extend to the case of theories $\Phi \subseteq \mathcal{L}$.

A theory $\Phi \subseteq \mathcal{L}$ *entails* a formula $\phi \in \mathcal{L}$, denoted $\Phi \models \phi$, iff for every interpretation $\mathcal{I}$ of $\mathcal{L}$ such that $\mathcal{I} \models \Phi, \mathcal{I} \models \phi$.

A predicate-based ontology language is a first-order language in which unary predicates represent classes of objects and binary predicates represent properties (relations between objects). Description Logics [3] are such predicate-based ontology languages.[1] Of special interest is the Description Logic $\mathcal{SHOIQ}$, which is a slight generalization of the language underlying the Semantic Web ontology language OWL DL [25].

[1]We recognize that several nonmonotonic extensions of Description Logics have been defined in the literature. In the present paper we restrict ourselves here to classical Description Logics.

$\mathcal{SHOIQ}$ descriptions are formed as follows, with A a concept identifier, C, C' descriptions, R, R' role identifiers, $o_1, \ldots, o_n$ individual identifiers, and n a positive integer; the sets of concept, role, and individual identifiers are mutually disjoint.

$$C, C' \longrightarrow A \mid \top \mid \bot \mid C \sqcap C' \mid C \sqcup C' \mid \neg C \mid \{o_1 \ldots o_n\} \mid \exists R.C \mid \forall R.C \mid \geqslant nR.C \mid \leqslant nR.C$$

A $\mathcal{SHOIQ}$ ontology is a set of axioms, with an axiom S formed as follows.

$$\begin{aligned} S \longrightarrow\ & C \sqsubseteq C' \mid C \equiv C' \mid R \sqsubseteq R' \mid R \equiv R'^{-} \mid \\ & \mathsf{Trans}(R) \mid o_1 \in C \mid \langle o_1, o_2 \rangle \in R \mid o_1 = o_2 \mid o_1 \neq o_2 \end{aligned}$$

We say that a role R is a sub-role of a role R' if $R \sqsubseteq^{*} R'$, where $\sqsubseteq^{*}$ is the reflexive-transitive closure of $\sqsubseteq$. For every *number restriction* $\geqslant nR.C, \leqslant nR.C$, holds that R must be *simple*, i.e., R and its sub-roles may not be transitive (with transitivity indicated by $\mathsf{Trans}(R)$).

Tables 1 and 2 present the semantics of $\mathcal{SHOIQ}$ in the form of a translation to first-order logic (cf. [7]). Table 1 defines the mapping function π from $\mathcal{SHOIQ}$ descriptions to first-order logic formulas. X is a meta-variable; it is replaced with a variable or constant during the translation. Table 2 defines the translation from $\mathcal{SHOIQ}$ axioms to closed first-order logic formulas. The mapping π naturally extends to sets of $\mathcal{SHOIQ}$ axioms.

Mapping concepts to FOL		
$\pi(A, X)$	$=$	$A(X)$
$\pi(\top, X)$	$=$	$X = X$
$\pi(\bot, X)$	$=$	$\neg(X = X)$
$\pi(C \sqcap C', X)$	$=$	$\pi(C, X) \wedge \pi(C', X)$
$\pi(C \sqcup C', X)$	$=$	$\pi(C, X) \vee \pi(C', X)$
$\pi(\neg C, X)$	$=$	$\neg\pi(C, X)$
$\pi(\{o_1 \ldots o_n\}, X)$	$=$	$\bigvee_{1 \leqslant i \leqslant n} X = o_i$
$\pi(\exists R.C, X)$	$=$	$\exists y(R(X, y) \wedge \pi(C, y))$
$\pi(\forall R.C, X)$	$=$	$\forall y(R(X, y) \supset \pi(C, y))$
$\pi(\geqslant nR.C, X)$	$=$	$\exists y_1, \ldots, y_n(\bigwedge_{1 \leqslant i \leqslant n}(R(X, y_i) \wedge \pi(C, y_i)) \wedge \bigwedge_{i \neq j} \neg y_i = y_j)$
$\pi(\leqslant nR.C, X)$	$=$	$\forall y_1, \ldots, y_{n+1}(\bigwedge_{1 \leqslant i \leqslant n+1}(R(X, y_i) \wedge \bigwedge \pi(C, y_i)) \supset \bigvee_{i \neq j} y_i = y_j)$
		where $y, y_1, \ldots, y_{n+1}$ are new variables

Table 1. $\mathcal{SHOIQ}$ descriptions and their mappings to FOL

Individual identifiers used in class definitions (in enumerations $\{o_1 \ldots o_n\}$) are called *nominals*. The Description Logic $\mathcal{SHIQ}$ corresponds to $\mathcal{SHOIQ}$ without nominals, i.e. the enumeration construct $(\{o_1, \ldots, o_n\})$ may not be used in $\mathcal{SHIQ}$.

Description Logic Programs (DLP) [16] are an intersection of Description Logics and logic programming, which means that a Description Logic Program can be seen both as a Description Logic ontology and as a logic program, and can thus be processed both by Description Logic and logic programming

Mapping axioms to FOL		
$\pi(C \sqsubseteq C')$	$=$	$\forall x(\pi(C,x) \supset \pi(C',x))$
$\pi(C \equiv C')$	$=$	$\pi(C \sqsubseteq C') \wedge \pi(C' \sqsubseteq C)$
$\pi(R \sqsubseteq R')$	$=$	$\forall x,y(R(x,y) \supset R'(x,y))$
$\pi(R \equiv R'^{-})$	$=$	$\forall x,y(R(x,y) \supset R'(y,x)) \wedge \forall x,y(R'(y,x) \supset R(x,y))$
$\pi(\mathsf{Trans}(R))$	$=$	$\forall x,y,z(R(x,y) \wedge R(y,z) \supset R(x,z))$
$\pi(o_1 \in C)$	$=$	$\pi(C,o_1)$
$\pi(\langle o_1,o_2\rangle \in R)$	$=$	$R(o_1,o_2)$
$\pi(o_1 = o_2)$	$=$	$o_1 = o_2$
$\pi(o_1 \neq o_2)$	$=$	$\neg(o_1 = o_2)$

Table 2. $\mathcal{SHOIQ}$ axioms and their mappings to FOL

reasoners. The Description Logic $\mathcal{DHL}$ is a Horn logic subset of the Description Logic $\mathcal{SHIQ}$, which means that a $\mathcal{DHL}$ axiom is a $\mathcal{SHIQ}$ axiom which is equivalent to a (conjunction of) Horn logic formula(s). With $\mathcal{DHLO}$ we denote the extension of $\mathcal{DHL}$ with nominals, while staying in the Horn fragment. A Description Logic Program (DLP) Π_O is obtained from a $\mathcal{DHL}$ ontology O by rewriting the axioms in O to Horn formulas and interpreting the formulas using the standard minimal Herbrand model semantics (see e.g. [23]). By the standard results in Logic Programming [23], we know that O and Π_O agree on ground entailment.

$\mathcal{DHLO}$ descriptions are formed as follows, with C_L, C'_L (resp. C_R, C'_R) descriptions which are allowed only on the left-hand (resp. right-hand) side of the class inclusion symbol $\sqsubseteq$.

$$\begin{array}{rcl} C, C' & \longrightarrow & A \mid C \sqcap C' \mid \exists R.\{o_1\} \\ C_L, C'_L & \longrightarrow & C \mid C_L \sqcup C'_L \mid \exists R.C_L \mid \geqslant 1R \mid \{o_1 \ldots o_n\} \\ C_R & \longrightarrow & C \mid \forall R.C_R \end{array}$$

A $\mathcal{DHLO}$ ontology is a set of axioms, with an axiom S formed as follows.

$$\begin{array}{rcl} S & \longrightarrow & C_L \sqsubseteq C_R \mid C \equiv C' \mid R \sqsubseteq R' \mid R \equiv R'^{-} \mid \mathsf{Trans}(R) \mid \\ & & \top \sqsubseteq \forall R.C_R \mid \top \sqsubseteq \forall R^{-}.C_R \mid o_1 \in A \mid \langle o_1, o_2\rangle \in R \end{array}$$

$\mathcal{DHL}$ is $\mathcal{DHLO}$ without nominals. The following proposition, which slightly extends [16, Theorem 1], establishes the correspondence between $\mathcal{DHLO}$ ontologies and Horn formulas.

Proposition 2.1. Let O be a set of $\mathcal{DHLO}$ axioms, and $\pi(O)$ its FOL equivalent. Then, there is exists set of Horn logic formulas Φ which is equivalent to $\pi(O)$, i.e. Φ and $\pi(O)$ have the same models.

Example 2.1. Consider the $\mathcal{DHLO}$ ontology O:

$$\begin{array}{rcl}
\exists colorOf.Eye \sqcap Color & \sqsubseteq & EyeColor \\
\top & \sqsubseteq & \forall hasColor.Color \\
hasColor & \equiv & colorOf^{-} \\
a & \in & Eye \\
\langle a, green\rangle & \in & hasColor
\end{array}$$

O corresponds to the set of Horn formulas Φ:

$$\begin{array}{rcl}
\forall x,y(colorOf(x,y) \wedge Eye(y) \wedge Color(x) & \supset & EyeColor(x)) \\
\forall x,y(hasColor(x,y) & \supset & Color(y)) \\
\forall x,y(hasColor(x,y) & \supset & colorOf(y,x)) \\
\forall x,y(colorOf(x,y) & \supset & hasColor(y,x)) \\
 & & Eye(a) \\
 & & hasColor(a, green)
\end{array}$$

Notice that $\Phi \models EyeColor(green)$.

Frame Logic Frame Logic [19, 20] (F-Logic) is an extension of first-order logic which adds explicit support for object-oriented modeling. It is possible to explicitly specify methods, as well as generalization/specialization and class instantiation relationships. The syntax of F-Logic has some seemingly higher-order features, namely, the same identifier can be used for a class, an instance, and a method. However, the semantics of F-Logic is strictly first-order[2]. To simplify matters, we do not consider parameterized, functional (single-valued) and inheritable methods, and compound molecules. Additionally, we consider only finite F-Logic theories.

The signature of an F-language $\mathcal{L}^F$ is of the form $\Sigma = \langle \mathcal{F}, \mathcal{P}\rangle$ with $\mathcal{F}$ and $\mathcal{P}$ disjoint countable sets of function and predicate symbols, each with an associated arity $n \geq 0$. Let $\mathcal{V}$ be a set of variable symbols. Terms and atomic formulas are constructed as in first-order logic with equality.

A molecule in F-Logic is one of the following statements: (i) an *is-a* assertion of the form $C : D$, (ii) a *subclass-of* assertion of the form $C :: D$, or (iii) a data molecule of the form $C[D \twoheadrightarrow E]$, with C, D, E terms. An F-Logic molecule is *ground* if it does not contain variables.

Formulas of an F-language $\mathcal{L}^F$ are either atomic formulas, molecules, or compound formulas which are constructed in the usual way from atomic formulas, molecules, and the logical connectives $\neg, \wedge, \vee, \supset$, the quantifiers $\exists, \forall$ and the auxiliary symbols ')','('. We denote universal closure with $(\forall)$.

F-Logic Horn formulas are of the form $(\forall)B_1 \wedge \ldots \wedge B_n \supset H$, with $B_1, \ldots, B_n, H$ atomic formulas or molecules. F-Logic Datalog formulas are F-Logic Horn formulas without function symbols such that every variable in H occurs in some equality-free $B_1, \ldots, B_n$.

An *F-structure* is a tuple $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow}\rangle$. Here, U is a countable nonempty set, $\prec_U$ is an irreflexive partial order on the domain U and $\in_U$ is a binary relation over U. We write $a \preceq_U b$ when

[2]Note that F-Logic is also often used as an extension of nonmonotonic logic programming. We consider such nonmonotonic F-Logic programs in Section 4.

$a \prec_U b$ or $a = b$, for $a, b \in U$. For each F-structure holds that if $a \in_U b$ and $b \preceq_U c$ then $a \in_U c$. Thus, if $b \preceq_U c$, then $\{k \mid k \in_U b, k \in U\} \subseteq \{k \mid k \in_U c, k \in U\}$, but the converse does not necessarily hold. I.e., if $b \preceq_U c$, then the class extension of b is a subset of the class extension of c. However, the converse of this statement is not universally true.

An n-ary function symbol $f \in F$ is interpreted as a function over the domain U: $\mathbf{I}_F(f) : U^n \to U$. An n-ary predicate symbol $p \in P$ is interpreted as a relation over the domain U: $\mathbf{I}_P(p) \subseteq U^n$. $\mathbf{I}_{\twoheadrightarrow}$ associates a partial function $U \to 2^U$ with each element of U: $\mathbf{I}_{\twoheadrightarrow} : U \longrightarrow U \to 2^U$. Variable assignments are defined as in first-order logic.

Given an F-structure $\mathbf{I}$, a variable assignment B, and a term t of $\mathcal{L}^F$, $t^{\mathbf{I},B}$ is defined as: $x^{\mathbf{I},B} = x^B$ for x a variable symbol and $t^{\mathbf{I},B} = \mathbf{I}_F(f)(t_1^{\mathbf{I},B}, \ldots, t_n^{\mathbf{I},B})$ for t of the form $f(t_1, \ldots, t_n)$.

F-satisfaction of ϕ in $\mathbf{I}$, given the variable assignment B, denoted $\mathbf{I}, B \models_{\mathsf{f}} \phi$, is defined as:

- $\mathbf{I}, B \models_{\mathsf{f}} p(t_1, \ldots, t_n)$ iff $(t_1^{\mathbf{I},B}, \ldots, t_n^{\mathbf{I},B}) \in \mathbf{I}_P(p)$,
- $\mathbf{I}, B \models_{\mathsf{f}} t_1 : t_2$ iff $t_1^{\mathbf{I},B} \in_U t_2^{\mathbf{I},B}$,
- $\mathbf{I}, B \models_{\mathsf{f}} t_1 :: t_2$ iff $t_1^{\mathbf{I},B} \preceq_U t_2^{\mathbf{I},B}$,
- $\mathbf{I}, B \models_{\mathsf{f}} t_1[t_2 \twoheadrightarrow t_3]$ iff $\mathbf{I}_{\twoheadrightarrow}(t_2^{\mathbf{I},B})(t_1^{\mathbf{I},B})$ is defined and $t_3^{\mathbf{I},B} \in \mathbf{I}_{\twoheadrightarrow}(t_2^{\mathbf{I},B})(t_1^{\mathbf{I},B})$, and
- $\mathbf{I}, B \models_{\mathsf{f}} t_1 = t_2$ iff $t_1^{\mathbf{I},B} = t_2^{\mathbf{I},B}$.

Extension to satisfaction of compound formulas is as in first-order logic.

The notions of a model and of validity are defined analogous to first-order logic. A theory $\Phi \subseteq \mathcal{L}^F$ *F-entails* a formula $\phi \in \mathcal{L}^F$, denoted $\Phi \models_{\mathsf{f}} \phi$, iff for every F-structure $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \Phi$, $\mathbf{I} \models_{\mathsf{f}} \phi$.

Sorted F-Logic In predicate-based ontology languages, the sets of symbols used for concepts, roles and individuals are disjoint. This is not the case in F-Logic. This disjointness can be regained by using a *sorted* F-Logic language.

We consider a sorted F-Logic language with three sorts: individuals, concepts and roles. A sorted F-Logic language has a sorted signature $\Sigma = \langle \mathcal{F}_a, \mathcal{F}_c, \mathcal{F}_r, \mathcal{P} \rangle$, where $\mathcal{F}_a$ is a set of function symbols, as before, $\mathcal{F}_c$ is a set of concept symbols, $\mathcal{F}_r$ is a set of role symbols, and $\mathcal{P}$ is a set of predicate symbols, as before. $\mathcal{F}_a, \mathcal{F}_c, \mathcal{F}_r$, and $\mathcal{P}$ are pairwise disjoint. The usual restrictions on the use of symbols in formulas applies, namely only molecules of the form $a : c$, $c :: d$, $a[r \twoheadrightarrow b]$ are allowed, with a, b terms constructed from symbols in $\mathcal{F}_a \cup \mathcal{V}$, $c, d \in \mathcal{F}_c \cup \mathcal{V}$, and $r \in \mathcal{F}_r \cup \mathcal{V}$. Quantifiers need to be qualified with a, c, r to indicate over which domain (individual, concept, role) the variable quantifies.

A sorted F-structure has three pairwise disjoint domains: U_a, U_c, U_r for the individuals, concepts, and roles, respectively; $\prec_U$ is an irreflexive partial order over U_c; $\in_U$ is a relation between U_a and U_c: $\in_U \subseteq U_a \times U_c$. $\mathbf{I}_F$ interprets symbols in $\mathcal{F}_a$ as functions over U_a, symbols in $\mathcal{F}_c$ as elements in U_c, and symbols of $\mathcal{F}_r$ as elements in U_r. $\mathbf{I}_P$ interprets symbols in $\mathcal{P}$ as n-ary relations over U_a^n. Finally, $\mathbf{I}_{\twoheadrightarrow}$ associates a partial mapping $U_a \to 2^{U_a}$ with each element of U_r.

3. Translating Predicate-Based Ontologies to F-Logic

In this section we define a straightforward translation from predicate-based ontologies to F-Logic. We show that when considering sorted F-Logic, the translation preserves entailment for arbitrary first-order

Entity	**Translation**		
Class	$\delta(A(t))$	$=$	$t\!:\!A$
Property	$\delta(R(t_1,t_2))$	$=$	$t_1[R \twoheadrightarrow t_2]$
Equality	$\delta(t_1 = t_2)$	$=$	$t_1 = t_2$
n-ary predicate	$\delta(P(\vec{t}))$	$=$	$P(\vec{t})$
Universal	$\delta(\forall\vec{x}.\phi)$	$=$	$\forall\vec{x}(\delta(\phi))$
Existential	$\delta(\exists\vec{x}.\phi)$	$=$	$\exists\vec{x}(\delta(\phi))$
Conjunction	$\delta(\phi \wedge \psi)$	$=$	$(\delta(\phi) \wedge \delta(\psi))$
Disjunction	$\delta(\phi \vee \psi)$	$=$	$(\delta(\phi) \vee \delta(\psi))$
Implication	$\delta(\phi \supset \psi)$	$=$	$(\delta(\phi) \supset \delta(\psi))$
Negation	$\delta(\neg\phi)$	$=$	$\neg(\delta(\phi))$

Table 3. Translation of predicate-based to frame-based modeling

theories. We then show that this is not the case in general when translating an ontology to an unsorted F-Logic language. However, for certain classes of first-order formulas, namely those where entailment can be reduced to validity of a *cardinal* formula [11], the translation preserves entailment.

Table 3 defines the mapping δ from the predicate style of ontology modeling to the frame style. In the table, A is a unary predicate symbol, ϕ, ψ are formulas, R is a binary predicate symbol, P is an n-ary predicate symbol, with $n = 0$ or $n \geq 3$, x is a variable symbol, and t, t_1, t_2 are terms. The mapping δ extends to sets of formulas in the natural way.

Definition 3.1. Given a first-order language $\mathcal{L}$ with the signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$. Let $\mathcal{L}^F$ be the F-Logic language with the same signature Σ, we say that $\mathcal{L}^F$ *corresponds to* $\mathcal{L}$. Given a first-order theory $\Phi \subseteq \mathcal{L}$, then we say that $\delta(\Phi) \subseteq \mathcal{L}^F$ is the *corresponding* F-Logic theory.

Our translation preserves function-freeness, i.e. if no function symbol of arity > 0 was used in the original ontology, no function symbol of arity > 0 will occur in the translated ontology. This is not the case for the translation of Description Logics to F-Logic presented in [4].

Proposition 3.1. Let Φ be a first-order theory which does not contain function symbols of arity > 0. Then, $\delta(\Phi)$ does not contain functions symbols of arity > 0.

In the remainder of this section we will first show that the translation in Definition 3.1 is faithful (i.e. preserves entailment) when considering a sorted F-Logic language. We will then show that for a certain class of formulas, the class of *cardinal* formulas (see [11]), the translation is also faithful when considering an unsorted language. Besides the classes of cardinal formulas identified in [11], we identify the novel class of $\mathcal{E}$-safe formulas, show that reasoning in $\mathcal{SHIQ}$ can be reduced to checking validity of $\mathcal{E}$-safe formulas, and show that $\mathcal{E}$-safe formulas are cardinal. Note that the class of $\mathcal{E}$-safe formulas goes beyond $\mathcal{SHIQ}$, and can capture a wide class of description languages.

3.1. Translating to Sorted F-Logic

We first investigate a translation to sorted F-Logic. We augment the translation in Table 3 to ensure that variables only quantify over the domain of individuals U_a by replacing each universal quantifier $\forall$ in Table 3 with $\forall_a$ and each existential quantifier $\exists$ with $\exists_a$. We denote the thus obtained translation function with δ^s.

We now show equi-satisfiability of formulas in $\mathcal{L}$, and their F-Logic counterparts. If $\mathcal{L}$ is a predicate-based ontology language with signature $\Sigma = \langle \mathcal{F}, \mathcal{P} \rangle$, then the corresponding sorted F-Logic language $\mathcal{L}^F$ has the signature $\Sigma' = \langle \mathcal{F}_a, \mathcal{F}_c, \mathcal{F}_r, \mathcal{P}' \rangle$, where $\mathcal{F}_a = \mathcal{F}$, $\mathcal{F}_c = \{t \mid t \text{ is a unary predicate symbol in } \mathcal{P}\}$, $\mathcal{F}_r = \{t \mid t \text{ is a binary predicate symbol in } \mathcal{P}\}$, and $\mathcal{P}' = \mathcal{P} - (\mathcal{F}_c \cup \mathcal{F}_r)$.

Lemma 3.1. Let ϕ be a formula in $\mathcal{L}$ and let $\mathcal{L}^F$ be the corresponding sorted F-Logic language, then ϕ is satisfied in some interpretation of $\mathcal{L}$ if and only if $\delta^s(\phi)$ is satisfied in some sorted F-structure of $\mathcal{L}^F$.

Proof:
(Sketch) From any interpretation $\mathcal{I}$ of $\mathcal{L}$ such that $\mathcal{I} \models \phi$ one can straightforwardly construct a corresponding sorted F-structure $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \delta^s(\phi)$, and vice versa. □

Using Lemma 3.1 we now obtain correspondence with respect to entailment.

Theorem 3.1. Let $\Phi \subseteq \mathcal{L}$ be a finite first-order theory and let $\phi \in \mathcal{L}$ be a formula. Then

$$\Phi \models \phi \text{ iff } \delta^s(\Phi) \models_{\mathsf{f}} \delta^s(\phi).$$

Proof:
Follows immediately from Lemma 3.1 and the fact that checking the entailment $\Phi \models \phi$ can be reduced to checking unsatisfiability of the formula $(\bigwedge \Phi) \wedge \neg\phi$ (similar for entailment in F-Logic). □

3.2. Translating Cardinal Formulas

We now consider the translation function δ of Table 3 in its original form, and we consider unsorted languages and F-structures of the form $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\rightarrow\!\!\!\rightarrow} \rangle$.

It turns out that we lose the correspondence of entailment and, more specifically, validity. Consider, for example,

$$\phi = (\forall x, y(x = y)) \supset (q(a) \leftrightarrow r(a)). \tag{1}$$

The formula ϕ is trivially satisfied in any interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ which has more than one element in the domain ($|U| \geq 2$), since the antecedent will be trivially false, and thus the implication trivially true, in such an interpretation. If we consider an interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ with only one element k in U, then the antecedent is true, but the consequent is not necessarily true, because q and r may be interpreted differently (e.g. $q^I = \emptyset, r^I = \{k\}$). Thus, ϕ is not valid in first-order logic (FOL). Now consider the corresponding F-Logic formula

$$\delta(\phi) = (\forall x, y(x = y)) \supset (a\!:\!q \leftrightarrow a\!:\!r).$$

Where ϕ is not a valid FOL formula, $\delta(\phi)$ is a valid F-Logic formula, since q and r must be interpreted as the same class in every F-structure which has a domain consisting of exactly one element.

From the example we can see that the translation δ is not faithful (with respect to validity, and thus also entailment) for arbitrary predicate-based ontology languages. We will show that there is a class of theories for which the correspondence does hold. This is the class of theories for which entailment can be reduced to checking validity of a *cardinal* formula [11].

Definition 3.2. Let ϕ be a formula in $\mathcal{L}$ and let γ denote the number of symbols in $\mathcal{L}$. An interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ is cardinal if $|U| \geq \gamma$. ϕ is *cardinal* if the following holds:

> If ϕ is satisfied in every cardinal interpretation of $\mathcal{L}$, then ϕ is satisfied in every interpretation of $\mathcal{L}$.

Definition 3.2 naturally extends to sets of formulas.

Note that this condition does not hold for the formula ϕ in (1), because ϕ is true in every interpretation with a domain of at least 3 elements, but it is not true in every interpretation of $\mathcal{L}$. The following definition of cardinality is equivalent to Definition 3.2.

Proposition 3.2. Let ϕ be a formula in $\mathcal{L}$, then ϕ is *cardinal* if and only if

> ϕ is unsatisfied in some cardinal interpretation of $\mathcal{L}$ whenever ϕ is unsatisfied in some interpretation of $\mathcal{L}$.

Proof:
Follows immediately from the fact that the proposition is the contraposition of Definition 3.2. $\square$

We now formulate our main result for the case of unsorted F-Logic:

Lemma 3.2. Let $\phi \in \mathcal{L}$ be a formula. Then,

1. if ϕ is valid in first-order logic, then $\delta(\phi)$ is valid in F-Logic, and
2. if ϕ is cardinal and $\delta(\phi)$ is valid in F-Logic, then ϕ is valid in first-order logic.

Proof:
See the appendix. $\square$

Theorem 3.2. Let $\Phi \subseteq \mathcal{L}$ be a finite first-order theory and $\phi \in \mathcal{L}$ be a formula. Then,

$$\text{if} \quad \Phi \models \phi \quad \text{then} \quad \delta(\Phi) \models_{\mathsf{f}} \delta(\phi).$$

If $\neg(\bigwedge \Phi) \vee \phi$ is cardinal, then also

$$\Phi \models \phi \quad \text{iff} \quad \delta(\Phi) \models_{\mathsf{f}} \delta(\phi).$$

Proof:
Follows from Lemma 3.2 and the observation that checking the entailment $\Phi \models \phi$ can be reduced to checking validity of the formula $\neg(\bigwedge \Phi) \vee \phi$. $\square$

Results on cardinal formulas from [11] can be applied directly to our case. From [11] we know that equality-free sentences, as well as the negation of a conjunction of Horn clauses with no equality in the antecedent, are cardinal. This is, however, not sufficient for many ontology languages. Description Logics such as $\mathcal{SHIQ}$ allow explicit assertion of equality between individuals and the introduction of equality statements through maximal number restrictions (see the translation of $\leqslant nR.C$ in Table 1).

We define the class of $\mathcal{E}$-safe formulas ($\mathcal{E}$ stands for "equality") which allow only *safe* uses of equality. With "safe" we mean that the use of equality does not restrict the size of the domains of the models.

For the definition of $\mathcal{E}$-safe formulas we need the notion of a *variable graph* of a conjunction of atoms. Given a conjunction of atomic formulas $\chi = \alpha_1 \wedge \ldots \wedge \alpha_n$, the variable graph of χ is an undirected graph $\langle N, E\rangle$ where the set of nodes N is the smallest set such that if α_i contains a variable, then $\alpha_i \in N$, for $1 \leq i \leq n$, and the set of edges E is the smallest set such that if α_i and α_j contain a common variable, then $\langle \alpha_i, \alpha_j\rangle \in E$, for $1 \leq i < j \leq n$.

We first define the class of *limited* $\mathcal{E}$-safe ($l\mathcal{E}$-safe) formulas, denoted $l\mathcal{ESF}$:

$$l\mathcal{ESF} ::= A \mid \neg A \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \forall \vec{x}(\chi \supset \phi) \mid \exists \vec{x}(\chi \wedge \phi)$$

where A is an atomic formula either of the form $p(\vec{t})$ or $t_1 = t_2$ with t_1, t_2 either both ground or both non-ground terms; ϕ, ϕ_1, ϕ_2 are $l\mathcal{E}$-safe formulas, and χ is either an atom of the form $p(\vec{t})$ or a conjunction of atoms of the form $p(\vec{t})$ such that the variable graph of χ is connected. Finally, every free variable in ϕ must appear in χ. We now define the class of $\mathcal{E}$-safe formulas, denoted $\mathcal{ESF}$:

$$\mathcal{ESF} ::= \varphi \mid \forall x(\phi) \mid \exists x(\phi) \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2$$

where ψ_1, ψ_2 are $\mathcal{E}$-safe formulas, ϕ and φ are $l\mathcal{E}$-safe formulas, and x is the only free variable in ϕ. As usual, an $\mathcal{E}$-safe sentence is an $\mathcal{E}$-safe formula without free variables.

The structure of $\mathcal{E}$-safe formulas is similar to the structure of *guarded* formulas [1]. The major distinctions are the restrictions on the use of the equality symbol in $\mathcal{E}$-safe formulas and the fact that the guard in an $\mathcal{E}$-safe formula may be a conjunction of atoms, whereas in the guarded fragment the guard always consists of a single atom.

Observe that the formulas $\forall x(x = x \supset \phi)$ and $\exists x(x = x \wedge \phi)$ are equivalent to $\forall x(\phi)$ and $\exists x(\phi)$, respectively. Because of this equivalence, we consider formulas of the forms $\forall x(x = x \supset \phi)$ and $\exists x(x = x \wedge \phi)$, where ϕ is an $l\mathcal{E}$-safe formula with at most one free variable x, $\mathcal{E}$-safe formulas. It turns out that the negation of an $\mathcal{E}$-safe formula is equivalent to an $\mathcal{E}$-safe formula as well.

Proposition 3.3. Let ϕ be an $\mathcal{E}$-safe formula. Then, there exists an $\mathcal{E}$-safe formula ψ which is equivalent to $\neg\phi$.

Proof:
(Sketch) It can be easily verified that the negation normal form (NNF) of $\neg\phi$ is an $\mathcal{E}$-safe formula. □

Example 3.1. The following formulas are $\mathcal{E}$-safe:

$$\forall x(p(x) \supset q(x))$$
$$\forall x(s(x,y) \supset p(x))$$
$$\exists x,y(p(x) \wedge r(x,y) \wedge x = y)$$
$$\forall x,y(r(x,y) \supset x = y)$$
$$\forall x(\forall y_1, y_2(person(x) \wedge father(x,y_1) \wedge father(x,y_2) \supset y_1 = y_2))$$

As can be seen from the above formulas, every variable participating in an equality atom is *guarded* by a non-equality atom.

The following formulas are not $\mathcal{E}$-safe:

$$\begin{aligned}
&\forall x, y(x = y)\\
&\forall x, y(q(x) \wedge q(y) \supset x = y)\\
&\forall x, y(x = y \supset p(x, y))\\
&\forall x(q(x) \supset x = a)
\end{aligned}$$

Many expressive Description Logic languages can be considered $\mathcal{E}$-safe, including $\mathcal{SHIQ}$, as is demonstrated in the following proposition.

Proposition 3.4. Let S be a $\mathcal{SHIQ}$ axiom. Then, $\pi(S)$ (resp., $\neg\pi(S)$) can be rewritten to an $\mathcal{E}$-safe formula ϕ (resp. ψ) such that $\pi(S)$ and ϕ (resp. $\neg\pi(S)$ and ψ) are equivalent, i.e., share the same models.

Proof:
Assume S is a $\mathcal{SHIQ}$ axiom. In case S is a property or individual axiom, $\pi(S)$ is trivially $\mathcal{E}$-safe and $\phi = \pi(S)$.

Say S is a class axiom such that $\delta(S)$ is of the form $\forall x(\phi_0 \supset \phi_1)$. Given the form of $\pi(S)$ and the translation in Table 2, one can transform $\phi_0 \supset \phi_1$ to a conjunction φ of $l\mathcal{E}$-safe formulas by, e.g., removing disjunction from the antecedent induces a splitting of the original formula in a conjunction of formulas, such that $\phi \equiv \forall x(\varphi)$ is an $\mathcal{E}$-safe formula which is equivalent to $\pi(S)$.

Since, by Proposition 3.3, the negation of an $\mathcal{E}$-safe formula is equivalent to an $\mathcal{E}$-safe formula, we have that $\neg\pi(S)$ is equivalent to an $\mathcal{E}$-safe formula as well. In fact, if ψ is the negation normal form of $\neg\pi(S)$, then ψ is $\mathcal{E}$-safe. $\square$

The class of $\mathcal{E}$-safe formulas is highly expressive, even when considering only function symbols of arity 0. In fact, it is easy to see, with a slight modification of Proposition 3.4, that $\mathcal{SHIQ}$ knowledge bases extended with certain kinds of (function-free) Horn formulas (those which can be translated to $\mathcal{E}$-safe formulas) can be equivalently translated to sets of $\mathcal{E}$-safe formulas. As entailment in this class of formulas is undecidable in general [22, 18][3], entailment of $\mathcal{E}$-safe formulas is undecidable in general as well.

We now formulate our main result about cardinal formulas.

Lemma 3.3. The following classes of first-order formulas are cardinal.

1. Sets of equality-free sentences,
2. formulas of the form $\neg S$, where S is a conjunction of Horn clauses without equality in the head, and
3. the class of $\mathcal{E}$-safe sentences.

[3]The proofs of undecidability of combinations of Description Logics (contained in $\mathcal{SHIQ}$) with Horn formulas in [22, 18] rely only on Horn formulas which are $\mathcal{E}$-safe.

Proof:
See the appendix. $\square$

The following corollary follows immediately from Theorem 3.2, Proposition 3.4, and Lemma 3.3:

Corollary 3.1. Let Φ be a finite set of $\mathcal{SHIQ}$ axioms and ϕ a $\mathcal{SHIQ}$ axiom, then

$$\Phi \models \phi \quad \text{iff} \quad \delta(\pi(\Phi)) \models_{\mathsf{f}} \delta(\pi(\phi)).$$

Note that $\mathcal{SHOIQ}$ formulas are not $\mathcal{E}$-safe in general, because of the possibility of using nominals. Consider, for example, the $\mathcal{SHOIQ}$ knowledge base $\{\top \sqsubseteq \{a\}\}$ which is equivalent to the first-order sentence $\forall x(x = a)$, which is not $\mathcal{E}$-safe. We obtain the following negative result about $\mathcal{SHOIQ}$.

Proposition 3.5. There are a finite $\mathcal{SHOIQ}$ theory O and a $\mathcal{SHOIQ}$ axiom S such that $\neg(\bigwedge \pi(O)) \vee \pi(S)$ is not cardinal and $\pi(O) \not\models \pi(S)$, but $\delta(\pi(O)) \models_{\mathsf{f}} \delta(\pi(S))$.

Proof:
Consider $O = \{\top \sqsubseteq \{a\}\}$ and $S = C \equiv D$. To show that $\neg(\bigwedge \pi(O)) \vee \pi(S)$ is not cardinal, consider an interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ such that $U = \{k\}$, $C^I = \emptyset$, and $D^I = \{k\}$. Clearly, $\mathcal{I} \not\models \neg(\bigwedge \pi(O)) \vee \pi(S)$. It is easy to verify that for any cardinal interpretation $\mathcal{I}^{\mathsf{c}}$, i.e. any interpretation with a domain of size ≥ 3, $\mathcal{I}^{\mathsf{c}} \models \neg(\bigwedge \pi(O)) \vee \pi(S)$. Therefore, $\neg(\bigwedge \pi(O)) \vee \pi(S)$ is not cardinal.

Clearly, $\pi(O) = \{\forall x(x = a)\} \not\models \pi(S) = \forall x(C(x) \equiv D(x))$. It is also easy to verify that $\delta(\pi(O)) = \{\forall x(x = a)\} \models \delta(\pi(S)) = \forall x(x\!:\!C \equiv x\!:\!D)$. $\square$

4. F-Logic Programming

F-Logic as we have considered so far in this paper is based on the standard semantics described in the original paper [20], which is a standard first-order semantics. However, all implementations of F-Logic we are aware of are based on nonmonotonic logic programming, which goes beyond the standard (monotonic) first-order semantics. In order to show properties of language layering involving languages based on logic programming we define F-Logic programs and give them a semantics based on the stable model semantics [14]. We show that F-Logic programs extend the Horn fragment of standard F-Logic, as defined in Section 2, and use this result, combined with the results in the previous section, to show that F-Logic programs generalize $\mathcal{DHLO}$.

A *normal F-Logic program* P consists of rules of the form

$$h \leftarrow b_1, \ldots, b_m, \textit{not}\ c_1, \ldots, \textit{not}\ c_n, \tag{2}$$

where $h, b_1, \ldots, b_m, c_1, \ldots, c_n$ are (equality-free) atoms or molecules. h is the *head atom* of r, $B^+(r) = \{b_1, \ldots, b_m\}$ is the set of *positive body atoms* of r, and $B^-(r) = \{c_1, \ldots, c_n\}$ is the set of *negative body atoms* of r. If $B^-(r) = \emptyset$, then r is *positive*. If every variable in r occurs in $B^+(r)$, then r is *safe*. If every rule $r \in P$ is positive (safe, respectively), then P is positive (safe, respectively). Additionally, every F-Logic program contains the following rules, which axiomatize the semantics of molecules:

$$x :: z \quad \leftarrow \quad x :: y, y :: z \tag{3}$$

$$x : z \quad \leftarrow \quad x : y, y :: z \tag{4}$$

$$x :: x \tag{5}$$

The first rule (3) axiomatizes transitivity of the subclass relation; the second rule (4) axiomatizes inheritance of class membership; the third rule (5) axiomatizes the fact that every class is a subclass of itself.[4]

The F-Logic signature Σ_P is a superset of the function and predicate symbols which occur in P. Let $\mathcal{L}_P^F$ denote the F-Logic language based on Σ_P. We assume that Σ_P contains at least one 0-ary function symbol or only 0-ary predicate symbols. The *Herbrand base*, B_H, of $\mathcal{L}_P^F$ is the set of ground atomic formulas and molecules of $\mathcal{L}_P^F$. Subsets of B_H are called *Herbrand interpretations*.

The *grounding* of a logic program P, denoted $gr(P)$, is the union of all possible ground instantiations of P, which are obtained by replacing each variable in a rule r with a ground term in Σ_P, for each rule $r \in P$.

Let P be a positive program. A Herbrand interpretation M of P is a *model* of P if, for every rule $r \in gr(P)$, $B^+(r) \subseteq M$ implies $h \in M$. A Herbrand model M is *minimal* iff for every model M' such that $M' \subseteq M$, $M' = M$.

Following [15], the *reduct* of a logic program P with respect to an interpretation M, denoted P^M, is the positive logic program obtained from $gr(P)$ by deleting (i) each rule r with $B^-(r) \cap M \neq \emptyset$, and (ii) $not\ c$ from the body of every remaining rule r with $c \in B^-(r)$. If M is a minimal Herbrand model of P^M, then M is a *stable model* of P.

If P is a positive logic program, then the corresponding Horn F-Logic theory Φ_P is obtained by, for every rule, replacing the arrow $\leftarrow$ with material implication $\supset$, and replacing every comma (,) in the body of the rule with $\wedge$.

Theorem 4.1. Let P be a positive logic program and Φ_P be the corresponding Horn F-Logic theory, then P has one stable model M and for every ground atom or molecule α, $\alpha \in M$ iff $\Phi_P \models \alpha$.

Proof:
Let Φ be an F-Logic theory and let Φ' be the FOL theory obtained from Φ by replacing every molecule of the form $t_1 : t_2$ with an atomic formula of the form $\_isa(t_1, t_2)$, molecules of the form $t_1 :: t_2$ with atomic formulas of the form $\_subclass(t_1, t_2)$, and molecules of the form $t_1[t_2 \twoheadrightarrow t_3]$ with atomic formulas of the form $\_att(t_1, t_2, t_3)$, with t_1, t_2, t_3 terms and $\_isa$, $\_subclass$, $\_att$ predicate symbols which do not occur in Φ, and adding the following formulas:

$$\begin{array}{lll} \forall & x, y, z & (\_subclass(x, y) \wedge \_subclass(y, z) \supset \_subclass(x, z)), \\ \forall & x, y, z & (\_isa(x, y) \wedge \_subclass(y, z) \supset \_isa(x, z)), \text{ and} \\ \forall & x & (\_subclass(x, x)). \end{array}$$

It is easy to verify that the F-Logic models of Φ and the FOL models of Φ' are isomorphic. The theorem follows immediately from the correspondence between minimal Herbrand models and first-order ground

[4]Note that this third rule is not safe, and thus may not be suitable for efficient implementation. However, the identity subclass relation is trivial and therefore often not required for query answering; alternatively, one could approximate the axiomatization as follows:

$$\begin{array}{lcl} x :: x & \leftarrow & x :: y \\ x :: x & \leftarrow & y :: x \\ x :: x & \leftarrow & y : x \end{array}$$

so the identity subclass relation is only returned in queries in case the term x is used as a class.

entailment [23], and the correspondence between stable models and minimal Herbrand models for positive programs [14]. □

F-Logic DLP There are several proposals for layering F-Logic programming on top of $\mathcal{DHLO}$ (e.g. [19, 10, 2, 6]). The following proposition shows that this layering is justified, in the sense that it preserves entailment.

Proposition 4.1. Let O be a $\mathcal{DHLO}$ ontology and let be α an equality-free ground atomic formula. Then,

$$O \models \alpha \quad \text{iff} \quad \delta(\pi(O)) \models_{\mathsf{f}} \delta(\alpha).$$

Proof:
Equivalence (with respect to entailment, modulo the transformation δ) between $\pi(O)$ and $\delta(\pi(O))$ follows from Theorem 3.2, Lemma 3.3 and the fact that $\pi(O)$ is equivalent to a set of Horn formulas without equality in the head. □

Given a $\mathcal{DHLO}$ ontology O, there is, by Proposition 2.1, a set of Horn formulas which is equivalent to O which we denote with O'. The F-Logic program corresponding to O, denoted P_O, corresponds to $\delta(O')$, interpreted under the Stable Model Semantics. The following corollary follows immediately from Theorem 4.1 and Proposition 4.1.

Corollary 4.1. Let O be a $\mathcal{DHLO}$ ontology, let P_O be the corresponding F-Logic program, and let α an equality-free ground atomic formula. Then, P_O has a single stable model M and

$$O \models \alpha \quad \text{iff} \quad \delta(\alpha) \in M.$$

A number of currently available F-Logic programming are based on the well-founded semantics [13], which is another semantics for negation in logic programs. In contrast to the stable model semantics, models in the well-founded semantics, called well-founded models, are three-valued, i.e. a ground atom is true, false, or *undefined*, and every logic program has exactly one well-founded model. With respect to entailment of ground atoms, and hence query answering, the well-founded semantics can be seen as an approximation to the stable model semantics, as shown by the following result from the literature.

Proposition 4.2. ([13])
Let P be a normal logic program and let M be the well-founded model of P. Then,

- if no atom in the Herbrand base is undefined in M, then M is the single stable model of P, and
- the set of ground atoms which is true in M is a subset of every stable model of P.

5. HILOG and Description Logics with Meta-Modeling

In this section we apply the results obtained in this paper to HILOG [11] and Description Logics with meta-modeling [24].

5.1. HILOG

HILOG [11] is a language which, like F-Logic, has syntactically higher-order features, but stays in a first-order semantic framework. However, in contrast to F-Logic, it does not have specific syntactical features for ontology modeling, but instead extends the syntax of first-order logic with the possibility to quantify over predicates and functions, as well as atomic formulas.

The alphabet of a HILOG language consists of a countable set $\mathcal{S}$ of parameter symbols. Let $\mathcal{V}$ be a countable sets of variable symbols, disjoint from $\mathcal{S}$. Then, terms are formed as follows: any $t \in \mathcal{S} \cup \mathcal{V}$ is a term and if $t, t_1, \ldots, t_n$ $(n \geq 0)$ are terms, then $t(t_1, \ldots, t_n)$ is a term. Any term and any equality atom $t_1 = t_2$ is an atomic formula. Complex formulas are built from atomic formulas in the usual way, using the usual logical connectives ($\wedge, \vee, \ldots, \supset$), quantifiers ($\forall, \exists$), and parentheses (')', '(').

A HILOG-structure is a quadruple $\mathbf{M} = \langle U, U_{true}, I, F \rangle$, where U is a countable nonempty set (the domain), $U_{true} \subseteq U$ specifies which of the elements in U correspond to true propositions, $I : \mathcal{S} \mapsto U$ maps parameter symbols to elements of the domain, and $F : U \mapsto [U^k \mapsto U]$, maps every $u \in U$ to a function $[U^k \mapsto U]$, also denoted $u_F^{(k)}$, for every positive integer $k \leq 1$.

Given a HILOG-structure $\mathbf{M}$ and a variable assignment $B : \mathcal{V} \mapsto U$, B recursively extends to terms as follows: $B(s) = I(s)$ for every $s \in \mathcal{S}$, and $B(t(t_1, \ldots, t_n)) = B(t)_F^{(n)}(B(t_1), \ldots, B(t_n))$.

A structure $\mathbf{M}$ *satisfies* an atomic formula ϕ, given a variable assignment B, if $B(\phi) \in U_{true}$. This extends to complex formulas in the usual way. As usual, $\mathbf{M}$ is a model of ϕ if $\mathbf{M}$ satisfies ϕ for every variable assignment B, and ϕ is HILOG-valid if every HILOG-structure is a model of ϕ. These notions extend to sets of formulas in the usual way.

We say that a set of formulas Φ *HILOG-entails* a formula ϕ if every HILOG-structure which is a model of Φ is also a model of ϕ.

The main result about cardinal formulas in HILOG is analogous to the result about cardinal formulas in F-Logic.

Proposition 5.1. ([11], Theorem 3.2)
Let $\Phi \subseteq \mathcal{L}$ be a finite first-order theory and $\phi \in \mathcal{L}$ a formula. Then,

$$\text{if} \quad \Phi \models \phi \quad \text{then} \quad \delta(\Phi) \ \text{ HILOG-entails } \ \delta(\phi).$$

If $\neg(\bigwedge \Phi) \vee \phi$ is cardinal, then, additionally,

$$\Phi \models \phi \quad \text{iff} \quad \Phi \ \text{ HILOG-entails } \ \phi.$$

The following corollary follows immediately from Lemma 3.3 and Proposition 5.1.

Corollary 5.1. Let $\phi \in \mathcal{L}$ be an $\mathcal{E}$-safe sentence, then

- ϕ is HILOG-valid if and only if ϕ is valid in first-order logic, and
- if $\Phi \subseteq \mathcal{L}$ is a finite set of $\mathcal{E}$-safe sentences, then Φ HILOG-entails ϕ if and only if $\Phi \models \phi$.

5.2. Description Logics with Meta-modeling

Two proposals for extending $\mathcal{SHOIQ}$ with meta-modeling support are presented in [24]. The proposals are based on the contextual predicate calculus and inspired by HILOG [11], respectively. The two proposals are called the π-semantics and the υ-semantics, respectively. We are concerned here with the υ-semantics, which features meta-modeling in the style of HILOG. However, there is a distinction: in HILOG, atomic formulas are interpreted as elements of a domain U, whereas this is not the case in the υ-semantics. In fact, in the υ-semantics, classes and properties are interpreted using class- and property-extension functions, in the spirit of the $\in_U$ and $\mathbf{I}_{\twoheadrightarrow}$ relation and function, in F-structures.

The definition of the syntax of $\mathcal{SHOIQ}$ (resp., $\mathcal{SHIQ}$) with meta-modeling is obtained from the definition of the syntax of $\mathcal{SHOIQ}$ (resp., $\mathcal{SHIQ}$) in Section 2 by omitting the requirement that the sets of concept, role, and individual identifiers are mutually disjoint; the sets of concept, role and individual identifiers comprise the set of parameter symbols $\mathcal{S}$.

A υ-interpretation is a tuple $I = \langle U, I_F, I_C, I_R\rangle$, where U is a countable non-empty set (the domain), I_F is a parameter interpretation function $I_F : \mathcal{S} \to U$, I_C is a class extension function $I_C : U \to 2^U$, and I_R is a role extension function $I_R : U \to 2^{(U\times U)}$. The class extension function I_C extends to descriptions as follows:

$$\begin{array}{rcl}
I_C(A) & = & A^I \subseteq U \\
I_C(\top) & = & U \\
I_C(\bot) & = & \emptyset \\
I_C(C \sqcap C') & = & I_C(C) \cap I_C(C') \\
I_C(C \sqcup C') & = & I_C(C) \cup I_C(C') \\
I_C(\neg C) & = & U \backslash I_C(C) \\
I_C(\{o_1 \dots o_n\}) & = & \{I_F(o_1), \dots, I_F(o_n)\} \\
I_C(\exists R.C) & = & \{x \mid (x,y) \in I_R(I_F(R)) \wedge y \in I_C(C)\} \\
I_C(\forall R.C) & = & \{x \mid (x,y) \in I_R(I_F(R)) \rightarrow y \in I_C(C)\} \\
I_C(\geq nR.C) & = & \{x \mid \|\{y \mid (x,y) \in I_R(I_F(R)) \wedge y \in I_C(C)\}\| \geq n\} \\
I_C(\leq nR.C) & = & \{x \mid \|\{y \mid (x,y) \in I_R(I_F(R)) \wedge y \in I_C(C)\}\| \leq n\}
\end{array}$$

Satisfiability of a formula ϕ in a υ-interpretation I, denoted $I \models_\upsilon \phi$, is defined as follows:

$$\begin{array}{lll}
I \models_\upsilon C \sqsubseteq C' & \text{iff} & I_C(C) \subseteq I_C(C'), \\
I \models_\upsilon C \equiv C' & \text{iff} & I_C(C) = I_C(C'), \\
I \models_\upsilon R \sqsubseteq R' & \text{iff} & I_R(I_F(R)) \subseteq I_R(I_F(R')), \\
I \models_\upsilon R \equiv R'^- & \text{iff} & I_R(I_F(R)) = I_R(I_F(R'))^-, \\
I \models_\upsilon \mathsf{Trans}(R) & \text{iff} & I_R(I_F(R))^+ \subseteq I_R(I_F(R)), \\
I \models_\upsilon o_1 \in C & \text{iff} & I_F(o_1) \in I_C(C), \\
I \models_\upsilon \langle o_1, o_2\rangle \in R & \text{iff} & \langle I_F(o_1), I_F(o_2)\rangle \in I_R(I_F(R)), \\
I \models_\upsilon o_1 = o_2 & \text{iff} & I_F(o_1) = I_F(o_2), \text{ and} \\
I \models_\upsilon o_1 \neq o_2 & \text{iff} & I_F(o_1) \neq I_F(o_2),
\end{array}$$

where R^+ denotes the transitive closure of the relation R, and R^- denotes the inverse of R.

The notions of a model and entailment are defined as usual. We denote entailment in the v-semantics with the symbol $\models_v$.

We extend the mapping π which was defined in Section 2 to map from $\mathcal{SHOIQ}$ with meta-modeling to contextual FOL; we can simply adopt the definitions of the mapping function from the Tables 1 and 2. We then use the mapping function δ to transform $\mathcal{SHOIQ}$ theories with meta-modeling to F-Logic. We obtain the following correspondence.

Theorem 5.1. Let Φ be a $\mathcal{SHOIQ}$ theory with meta-modeling. Then, there exists a v-interpretation I such that $I \models_v \Phi$ iff there exists an F-structure $\mathbf{I}$ such that $\mathbf{I} \models_{\mathsf{f}} \delta(\pi(\Phi))$.

Proof:
(Sketch) ($\Rightarrow$) Let $I = \langle U, I_F, I_C, I_R \rangle$ be a v-interpretation such that $I \models_v \Phi$. The corresponding F-structure $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$ is defined as follows: (i) the domains are the same, (ii) $\mathbf{I}_F(t) = I_F(t)$ for every $t \in \mathcal{S}$, (iii) $u_1 \in_U u_2$ if $u_1 \in I_C(u_2)$, for every pair $(u_1, u_2) \in U \times U$, and (iv) $\mathbf{I}_{\twoheadrightarrow}(u) = I_R(u)$ for every $u \in U$. It is easy to verify that $\mathbf{I} \models_{\mathsf{f}} \Phi$.
($\Leftarrow$) Analogous. $\Box$

Since $\mathcal{SHOIQ}$ with meta-modeling is closed under negation, correspondence with respect to validity and entailment follows immediately.

Using this correspondence, we can immediately apply the results obtained in this paper to $\mathcal{SHIQ}$ and $\mathcal{SHOIQ}$ with meta-modeling.

Proposition 5.2.

1. If Φ is a finite $\mathcal{SHIQ}$ theory and ϕ is a $\mathcal{SHIQ}$ axiom, then Φ v-entails ϕ iff $\Phi \models \phi$;
2. if Φ is a finite $\mathcal{SHOIQ}$ theory, ϕ is a $\mathcal{SHOIQ}$ axiom, and $\Phi \models \phi$, then Φ v-entails ϕ; and
3. there are a $\mathcal{SHOIQ}$ theory Φ' and a $\mathcal{SHOIQ}$ axiom ϕ' such that Φ' v-entails ϕ', but $\Phi' \not\models \phi'$.

Note that (2) in Proposition 5.2 was already implicit in [24].

6. Related Work

Balaban [4] proposes to use F-Logic as an underlying framework for Description Logics and uses the flexibility of F-Logic to extend Description Logics. DFL [5] uses F-Logic to reason about ontologies and rules. The major differences between the approach of Balaban and our approach are: (a) we do not need function symbols if the original language does not use function symbols; (b) we allow arbitrary predicate-based ontology languages, whereas Balaban's translation is restricted to Description Logics; and (c) Balaban uses a sorted F-Logic, whereas we do not need sorts for a large class of formulas.

F-OWL [27] uses FLORA [26], an F-Logic programming implementation, to reason over OWL. The authors capture the semantics of OWL using entailment rules over RDF triples. It is not clear exactly which part of the semantics of OWL is captured in F-OWL.

7. Conclusions

In predicate-based ontology representation languages (e.g. Description Logics), classes are modeled as unary predicates and properties as binary predicates, which are interpreted as sets and as binary relations, respectively. In F-Logic, classes and properties are both first interpreted as objects and then related to sets and relations, respectively.

In this paper we have introduced a translation from predicate-based ontologies to ontologies in F-Logic. We have shown that this translation preserves entailment for large classes of predicate-based ontology languages, including the class of *cardinal* formulas. Intuitively, cardinal formulas do not restrict the size of the domains of the models. We have defined the class of $\mathcal{E}$-safe formulas and shown that $\mathcal{E}$-safe formulas are cardinal. Finally, we have shown that the class of $\mathcal{E}$-safe formulas is a very expressive class of formulas which includes the Description Logic $\mathcal{SHIQ}$.

We have used the translation to close the open problem of the F-Logic extension of Description Logic Programs [16]. Furthermore, we have shown that our results also apply to HILOG [11] and the v-semantics for Description Logics with meta-modeling [24].

The results obtained in this paper can be used for, for example, F-Logic based reasoning with, and extension of, classes of predicate-based ontology languages. Another application of the results is the use of F-Logic as a vehicle for the extension of RDF(S) [17]. Initial results on the F-Logic-based reasoning with, and extension of, RDF(S) are reported in [9].

Appendix. Proofs of Lemmas 3.2 and 3.3

Proof of Lemma 3.2:
To prove *1.* we define, for a given F-Logic structure $\mathbf{I}$, the corresponding FOL interpretation $(\mathbf{I})^{FOL}$. We then proceed by contradiction, and show that whenever there is an F-Logic structure $\mathbf{I}$ which does not satisfy the formula $\delta(\phi)$, the corresponding FOL interpretation $(\mathbf{I})^{FOL}$ does not satisfy ϕ, contradicting the assumption that ϕ is valid in FOL:

1. Given an F-structure $\mathbf{I} = \langle U, \prec_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow} \rangle$ for the language $\mathcal{L}^F$, the corresponding FOL interpretation $\mathcal{I} = (\mathbf{I})^{FOL} = \langle U, \cdot^I \rangle$ for $\mathcal{L}$ is defined as follows: (i) $\forall\, f \in \mathcal{F}$: $f^I = \mathbf{I}_F(f)$, (ii) $\forall$ unary $c \in \mathcal{P}$: $c^I = \{k \mid k \in_U \mathbf{I}_F(c), k \in U\}$, (iii) $\forall$ binary $r \in \mathcal{P}$: $r^I = \{\langle k_1, k_2 \rangle \mid k_2 \in \mathbf{I}_{\twoheadrightarrow}(\mathbf{I}_F(r))(k_1)$, for $k_1, k_2 \in U\}$, and (iv) $\forall$ non-unary and non-binary $p \in \mathcal{P}$: $p^I = \mathbf{I}_P(p)$.

Assume that ϕ is valid in first-order logic, but $\delta(\phi)$ is not valid in F-Logic. Then, there must be some F-structure $\mathbf{I}$ such that $\mathbf{I} \not\models_{\mathrm{f}} \delta(\phi)$. It is easy to verify that $\mathcal{I} = (\mathbf{I})^{FOL}$ does not satisfy ϕ, contradicting the assumption that ϕ is valid in first-order logic.

The proof of *2.* is similar: for a given cardinal FOL interpretation $\mathcal{I}$, we define the corresponding F-Logic structure $(\mathcal{I})^{FL}$. We then proceed by contradiction. We show that whenever there is a cardinal interpretation $\mathcal{I}$ which does not satisfy the formula ϕ, $(\mathcal{I})^{FL}$ does not satisfy $\delta(\phi)$. By Proposition 3.2 we know that whenever a cardinal formula ϕ is not valid in FOL, there must be some cardinal interpretation $\mathcal{I}$ such that $\mathcal{I} \not\models \phi$. Therefore, if a cardinal formulas ϕ is not valid, $\delta(\phi)$ is not valid, contradicting the assumption that $\delta(\phi)$ is valid in F-Logic.

2. Given a cardinal interpretation $\mathcal{I} = \langle U, \cdot^I \rangle$ of $\mathcal{L}$. Since $|U|$ is greater than or equal to the number of symbols in $\mathcal{L}$, we may assume that for each $q \in \mathcal{P}$ there is a unique individual $k_q \in U$. $\mathbf{I} = (\mathcal{I})^{FL} =$

$\langle U, \preceq_U, \in_U, \mathbf{I}_F, \mathbf{I}_P, \mathbf{I}_{\twoheadrightarrow}\rangle$ is the corresponding F-Logic structure, which is defined as follows: (i) $\forall\, f \in \mathcal{F}$: $\mathbf{I}_F(f) = f^I$, (ii) $\forall$ unary $c \in \mathcal{P}$: $\mathbf{I}_F(c) = k_c$, (iii) $\forall$ binary $r \in \mathcal{P}$: $\mathbf{I}_F(r) = k_r$, (iv) $\forall$ unary $c \in \mathcal{P}$ and every individual $k \in U$, if $k \in c^I$ then $k \in_U \mathbf{I}_F(c)$, (v) $\forall$ unary $c_1, c_2 \in \mathcal{P}$: $\mathbf{I}_F(c_1) \preceq_U \mathbf{I}_F(c_2)$ if $c_1^I \subseteq c_2^I$, (vi) $\forall$ binary $r \in \mathcal{P}$ and $\forall\, k_1, k_2 \in U$, if $\langle k_1, k_2\rangle \in r^I$ then $k_2 \in \mathbf{I}_{\twoheadrightarrow}(\mathbf{I}_F(r))(k_1)$, and (vii) $\forall$ non-unary and non-binary $p \in \mathcal{P}$: $\mathbf{I}_P(p) = p^I$.

Let ϕ be a cardinal formula. Assume that $\delta(\phi)$ is valid in F-Logic, but ϕ is not valid in first-order logic. Then, by Proposition 3.2, there must be some cardinal interpretation $\mathcal{I} = \langle U, \cdot^I\rangle$ such that $\mathcal{I} \not\models \phi$, and thus $(\mathcal{I}, B) \not\models \phi$ for some variable assignment B. Since $\mathcal{I}$ is cardinal, $\mathbf{I} = (\mathcal{I})^{FL}$ is defined. To prove the lemma, it is sufficient to show that $\mathbf{I}, B \not\models_{\mathrm{f}} \delta(\phi)$, contradicting the assumption that $\delta(\phi)$ is valid in F-Logic. We proceed by induction over the structure of the formula ϕ.

Consider $\phi = A(t)$. $(\mathcal{I}, B) \not\models \phi$ iff $t^{\mathcal{I},B} \notin A^I$ iff $t^{\mathbf{I},B} \notin_U \mathbf{I}_F(A)$. The 'only if' direction follows from (v) in the translation above. The 'if' direction follows from the fact that, by construction of $\mathbf{I}$, $\mathbf{I}_F(C) \neq k$ for any $k = \mathbf{I}_F(D)$, with $D \neq C$ a concept identifier. Similar for formulas of the form $R(t_1, t_2)$.

Consider $\phi = (t_1 = t_2)$. $(\mathcal{I}, B) \models \phi$ iff $t_1^{\mathcal{I},B} = t_2^{\mathcal{I},B}$ iff $t_1^{\mathbf{I},B} = t_2^{\mathbf{I},B}$. The last 'iff' follows trivially from the construction of $\mathbf{I}$.

Consider $\phi = \forall x(\psi)$. $(\mathcal{I}, B) \not\models \phi$ iff for some x-variant B' of B, $(\mathcal{I}, B') \not\models \psi$ iff $(\mathbf{I}, B') \not\models \delta(\psi)$. The last 'iff' follows by induction and from the observation that the domains of $\mathcal{I}$ and $\mathbf{I}$ are the same. Similar for $\phi = \exists x(\psi)$. This can be trivially extended to formulas of the forms $\neg\psi$, $\psi_1 \wedge \psi_2$, and $\psi_1 \vee \psi_2$. □

Proof of Lemma 3.3:
Cardinality of the first and second class is shown in [11]. The proof of cardinality of $\mathcal{E}$-safe sentences proceeds along the following lines. For each $\mathcal{E}$-safe φ sentence we show that whenever φ is not satisfied in some interpretation $\mathcal{I}$, then φ is not satisfied in a corresponding cardinal interpretation $\mathcal{I}^{\mathsf{c}}$. From Proposition 3.2 follows that φ is cardinal.

We first define, given an interpretation $\mathcal{I}$, the corresponding cardinal interpretation $\mathcal{I}^{\mathsf{c}}$. We need the following auxiliary notion. Given an interpretation $\mathcal{I} = \langle U, \cdot^I\rangle$ of a language $\mathcal{L}$ with signature $\Sigma_{\mathcal{L}} = \langle \mathcal{F}, \mathcal{P}\rangle$, $k \in U$ is *unused* in $\mathcal{I}$ if: (a) k does not occur in the domain or the range of the function $f^I : U^n \rightarrow U$ for any $f \in \mathcal{F}$, and (b) k does not occur in any tuple of the relation $p^I \subseteq U^n$ for any $p \in \mathcal{P}$.

If $\mathcal{I}$ is cardinal, then $\mathcal{I}^{\mathsf{c}} = \mathcal{I}$. Otherwise, let $\mathcal{I}^{\mathsf{c}} = \langle U^{\mathsf{c}}, \cdot^{I^{\mathsf{c}}}\rangle$ be the cardinal interpretation obtained from $\mathcal{I}$ by adding a set K of individuals to the domain: $U^{\mathsf{c}} = U \cup K$, such that $|U^{\mathsf{c}}| = \gamma$, where γ is the number of symbols in the language $\mathcal{L}$, and defining $\cdot^{I^{\mathsf{c}}}$ in the following way:

1. for every function symbol $f \in \mathcal{F}$, $f^{I^{\mathsf{c}}} = f^I$,

2. for every n-ary predicate symbol $p \in \mathcal{P}$, if $\vec{k} \in p^I$, then $\vec{k} \in p^{I^{\mathsf{c}}}$, and for every $i \in \{1, \ldots, n\}$ and every $k \in K$, if there is some tuple $\langle a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n\rangle \in p^{I^{\mathsf{c}}}$, then $\langle a_1, \ldots, a_{i-1}, k, a_{i+1}, \ldots, a_n\rangle \in p^{I^{\mathsf{c}}}$.

There are five kinds of $\mathcal{E}$-safe sentences: (1) $l\mathcal{ESF}$ sentences, (2) universal and (3) existential $\mathcal{E}$-safe sentences, and (4) conjunctions and (5) disjunctions of $\mathcal{E}$-safe sentences. Any $l\mathcal{ESF}$ sentence ϕ is equivalent to a universal sentence $\forall x(\phi)$. We proceed to show, for each of the types of formulas (2–5), that whenever a sentence φ is not satisfied in an interpretation $\mathcal{I}$, φ is not satisfied in $\mathcal{I}^{\mathsf{c}}$, i.e. if $\mathcal{I} \not\models \varphi$, then $\mathcal{I}^{\mathsf{c}} \not\models \varphi$. It follows from Proposition 3.2 that φ is cardinal.

(2) $\varphi = \forall x(\phi)$:

Since $\mathcal{I} \not\models \forall x(\phi)$, there must be some variable assignment B of $\mathcal{I}$ such that $(\mathcal{I}, B) \not\models \phi$. We show that $(\mathcal{I}^{\mathsf{c}}, B) \not\models \phi$ by induction over the structure of ϕ. W.l.o.g. we assume that $\exists x, \forall x$ do not occur in ϕ.

- $\phi = p(t_1, \ldots, t_n)$: since $(\mathcal{I}, B) \not\models \phi$, $\langle t_1^{\mathcal{I},B}, \ldots, t_n^{\mathcal{I},B} \rangle \notin p^I$. By construction of $\mathcal{I}^{\mathsf{c}}$, we have that also $\langle t_1^{\mathcal{I}^{\mathsf{c}},B}, \ldots, t_n^{\mathcal{I}^{\mathsf{c}},B} \rangle \notin p^{I^c}$, and thus $(\mathcal{I}^{\mathsf{c}}, B) \not\models \phi$. Similar for $\phi = \neg p(t_1, \ldots, t_n)$.

- $\phi = t_1 = t_2$: follows immediately from the construction of $\mathcal{I}^{\mathsf{c}}$. Similar for $\phi = \neg t_1 = t_2$.

- $\phi = \psi_1 \vee \psi_2$: by induction we obtain $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_1$ and $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_2$.

- $\phi = \psi_1 \wedge \psi_2$: if $(\mathcal{I}, B) \not\models \psi_1$ (resp., $(\mathcal{I}, B) \not\models \psi_2$), we obtain $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_1$ (resp., $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_2$) by induction.

- $\phi = \forall \vec{y}(\chi \supset \psi)$: let B' be a $\vec{y}$-variant of B such that $(\mathcal{I}, B') \not\models \chi \supset \psi$; observe that $x^{B'} = x^B$. Since $(\mathcal{I}, B) \not\models \phi$, such a B' must exist. We have that $(\mathcal{I}, B') \models \chi$, and thus $(\mathcal{I}^{\mathsf{c}}, B') \models \chi$, by construction of $\mathcal{I}^{\mathsf{c}}$. We obtain $(\mathcal{I}^{\mathsf{c}}, B') \not\models \psi$ by induction.

- $\phi = \exists \vec{y}(\chi \wedge \psi)$: by assumption, we have that for every variable assignment B' of $\mathcal{I}$ which is a $\vec{y}$-variant of B, $(\mathcal{I}, B') \not\models \chi \wedge \psi$. Notice that, by construction of ϕ, every free variable in ψ occurs in χ. Now, if for every free variable y in χ holds that $y^{B'} \in U$, $(\mathcal{I}^{\mathsf{c}}, B') \not\models \chi \wedge \psi$ immediately follows by induction, since χ, ψ are both $l\mathcal{E}$-safe.

  Now, assume that for some free variable y in χ holds that $y^{B'} = k \in K$ and $(\mathcal{I}^{\mathsf{c}}, B') \models \chi \wedge \psi$. It is easy to verify, by construction of $\mathcal{I}^{\mathsf{c}}$, that there must be some variable assignment B'' which is a $\vec{y}$-variant of B such that $(\mathcal{I}, B'') \models \chi \wedge \psi$, and thus $(\mathcal{I}, B) \models \phi$, contradicting the assumption $(\mathcal{I}, B) \not\models \phi$.

(3) $\varphi = \exists x(\phi)$:

Since $\mathcal{I} \not\models \exists x(\phi)$, there is no variable assignment B' of $\mathcal{I}$ such that $(\mathcal{I}, B') \models \phi$. Let B be a variable assignment of $\mathcal{I}^{\mathsf{c}}$. We show that $(\mathcal{I}^{\mathsf{c}}, B) \not\models \phi$ by induction over the structure of ϕ.

- $\phi = p(t_1, \ldots, t_n)$: if for every t_i holds that $t_i^{\mathcal{I},B} \notin K$, then clearly $(\mathcal{I}^{\mathsf{c}}, B) \not\models \phi$. Now assume $t_i^{\mathcal{I},B} = k \in K$ and $(\mathcal{I}^{\mathsf{c}}, B) \models p(t_1, \ldots, t_n)$ for some t_i. This means that there is some tuple $\langle a_1, \ldots, a_{i-1}, k, a_{i+1}, \ldots, a_n \rangle \in p^{I^{\mathsf{C}}}$, and thus, by construction of $\mathcal{I}^{\mathsf{c}}$, there must be some tuple $\langle a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n \rangle \in p^I$, thus $(\mathcal{I}, B') \models \exists x(\phi)$ for some variable assignment B' such that $x^{B'} = a_i$, contradicting the assumption that $\mathcal{I} \not\models \exists x(\phi)$. Similar for $\phi = \neg p(\vec{t})$.

- $\phi = t_1 = t_2$: since $\mathcal{I} \not\models \exists x(\phi)$, t_1 and t_2 must be different terms. If t_i is a ground term or a constructed term, then clearly $t_i^{\mathcal{I},B} \notin K$; $(\mathcal{I}^{\mathsf{c}}, B) \not\models \phi$ follows straightforwardly from $\mathcal{I} \not\models \exists(\phi)$. Assume that t_1 and t_2 are both variables and $t_1^B = t_2^B = k$ for some $k \in K$, then it is straightforwardly construct a variable assignment B' of $\mathcal{I}$ such that $t_1^B = t_2^B = k$ for some $k \in U$, and thus $(\mathcal{I}, B') \models \phi$, contradicting the assumption that $\mathcal{I} \not\models \exists x(\phi)$. Similar for $\phi = \neg t_1 = t_2$.

- $\phi = \psi_1 \vee \psi_2$: by induction we obtain $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_1$ and $(\mathcal{I}^{\mathsf{c}}, B) \not\models \psi_2$.

- $\phi = \psi_1 \wedge \psi_2$: let B' be a variable assignment of $\mathcal{I}$ such that $y^{B'} = y^B$ for all variables $y \neq x$, then if $(\mathcal{I}, B') \not\models \psi_1$ (resp., $(\mathcal{I}, B') \not\models \psi_2$), we obtain $(\mathcal{I}^c, B) \not\models \psi_1$ (resp., $(\mathcal{I}^c, B) \not\models \psi_2$) by induction.
- $\phi = \forall \vec{y}(\chi \supset \psi)$: let B' be a $\vec{y}$-variant of B such that $(\mathcal{I}, B') \not\models \chi \supset \psi$. Since $\mathcal{I} \not\models \exists x(\phi)$, such a B' must exist. We have that $(\mathcal{I}, B') \models \chi$, and thus $(\mathcal{I}^c, B') \models \chi$, by construction of $\mathcal{I}^c$. We obtain $(\mathcal{I}^c, B') \not\models \psi$ by induction.
- $\phi = \exists \vec{y}(\chi \wedge \psi)$: by assumption, we have that for every $\vec{y}$-variant B' of B, $(\mathcal{I}, B') \not\models \chi \wedge \psi$. Assume that $y^{B'} \notin K$ for all variables $y \neq x$. Since χ, ψ are both $l\mathcal{E}$-safe, we obtain $(\mathcal{I}^c, B') \not\models \chi \wedge \psi$ by induction.

  Now assume that for some free variable y in χ holds that $y^{B'} = k \in K$ and $(\mathcal{I}^c, B') \models \chi \wedge \psi$. It is easy to verify that, by construction of $\mathcal{I}^c$, there must be some variable assignment B'' which is a y-variant of B such that $(\mathcal{I}, B'') \models \chi \wedge \psi$, contradicting the assumption.

(4) $\varphi = \psi_1 \wedge \psi_2$:
Assume that $\mathcal{I} \not\models \psi_1 \wedge \psi_2$, and thus $\mathcal{I} \not\models \psi_1$ or $\mathcal{I} \not\models \psi_2$. By induction we immediately obtain $\mathcal{I}^c \not\models \psi_1$ or $\mathcal{I}^c \not\models \psi_2$, and thus $\mathcal{I}^c \not\models \psi_1 \wedge \psi_2$

(5) $\varphi = \psi_1 \wedge \psi_2$:
Assume that $\mathcal{I} \not\models \psi_1 \vee \psi_2$, and thus $\mathcal{I} \not\models \psi_1$ and $\mathcal{I} \not\models \psi_2$. By induction we immediately obtain $\mathcal{I}^c \not\models \psi_1$ and $\mathcal{I}^c \not\models \psi_2$, and thus $\mathcal{I}^c \not\models \psi_1 \vee \psi_2$. $\square$

Acknowledgement

The authors would like to thank Uwe Keller, Michael Kifer, and three anonymous reviewers for valuable comments on earlier versions of the paper.

References

[1] Andréka, H., van Benthem, J., Németi, I.: Modal languages and bounded fragments of predicate logic, *Journal of Philosophical Logic*, **27**, 1998, 217–274.

[2] Angele, J., Boley, H., Bruijn, J. de, Fensel, D., Hitzler, P., Kifer, M., Krummenacher, R., Lausen, H., Polleres, A., Studer, R.: Web Rule Language (WRL), W3C Member Submission 09 September 2005.

[3] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F., Eds.: *The Description Logic Handbook*, Cambridge University Press, 2003.

[4] Balaban, M.: The F-logic Approach for Description Languages, *Annals of Mathematics and Artificial Intelligence*, **15**(1), 1995, 19–60.

[5] Balaban, M., Eyal, A.: DFL - a dialog based integration of concept and rule reasoners, *Data & Knowledge Engineering*, **38**(3), 2001, 301–334.

[6] Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuinness, D., Su, J., Tabet, S.: Semantic Web Services Language (SWSL), W3C Member Submission 09 September 2005.

[7] Borgida, A.: On the Relative Expressiveness of Description Logics and Predicate Logics, *Artificial Intelligence*, **82**(1–2), 1996, 353–367.

[8] Bruijn, J. de, Heymans, S.: Translating Ontologies from Predicate-based to Frame-based Languages, *Proceedings of the 2nd International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML2006)*, Athens, Georgia, USA, November 10-11 2006.

[9] Bruijn, J. de, Heymans, S.: Logical Foundations of (e)RDF(S): Complexity, Reasoning, and Extension, *Proceedings of the 6th International Semantic Web Conference (ISWC2007)*, Busan, Korea, November 2007.

[10] Bruijn, J. de, Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language: An Overview, *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, Budva, Montenegro, June 2006.

[11] Chen, W., Kifer, M., Warren, D. S.: HILOG: A Foundation for Higher-Order Logic Programming, *Journal of Logic Programming*, **15**(3), 1993, 187–230.

[12] Dean, M., Schreiber, G., Eds.: *OWL Web Ontology Language Reference*, W3C Recommendation 10 February 2004.

[13] Gelder, A. V., Ross, K., Schlipf, J. S.: The Well-Founded Semantics for General Logic Programs, *Journal of the ACM*, **38**(3), 1991, 620–650.

[14] Gelfond, M., Lifschitz, V.: The Stable Model Semantics for Logic Programming, *Proceedings of the Fifth International Conference on Logic Programming* (R. A. Kowalski, K. Bowen, Eds.), The MIT Press, Cambridge, Massachusetts, 1988.

[15] Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing*, **9**(3/4), 1991, 365–386.

[16] Grosof, B. N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic, *Proceedings of the 12th International Conference on the World Wide Web (WWW2003)*, Budapest, Hungary, 2003.

[17] Hayes, P., Ed.: *RDF Semantics*, W3C Recommendation 10 February 2004.

[18] Horrocks, I., Patel-Schneider, P. F.: A Proposal for an OWL Rules Language, *Proceedings of the 13th International World Wide Web Conference (WWW2004)*, New York City, USA, 2004.

[19] Kifer, M.: Rules and Ontologies in F-Logic, *Reasoning Web, First International Summer School, Tutorial Lectures*, Msida, Malta, July 2005.

[20] Kifer, M., Lausen, G., Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages, *JACM*, **42**(4), 1995, 741–843.

[21] Kifer, M., Wu, J.: A Logic for Object-Oriented Logic Programming (Maier's O-Logic Revisited), *Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1989.

[22] Levy, A. Y., Rousset, M.-C.: Combining Horn rules and description logics in CARIN, *Artificial Intelligence*, **104**, 1998, 165 – 209.

[23] Lloyd, J. W.: *Foundations of Logic Programming (2nd edition)*, Springer-Verlag, 1987.

[24] Motik, B.: On the Properties of Metamodeling in OWL, *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, Springer, Galway, Ireland, 2005.

[25] Patel-Schneider, P. F., Hayes, P., Horrocks, I.: *OWL Web Ontology Language Semantics and Abstract Syntax*, W3C Recommendation 10 February 2004.

[26] Yang, G., Kifer, M., Zhao, C.: FLORA-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web, *Proceedings of the Second International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)*, Catania, Sicily, Italy, 2003.

[27] Zou, Y., Finin, T., Chen, H.: F-OWL: an Inference Engine for the Semantic Web, *Formal Approaches to Agent-Based Systems, Third International Workshop, FAABS 2004, Revised Selected Papers*, LNCS, Springer, Greenbelt, USA, 2004.