Embedding non-ground logic programs into autoepistemic logic for knowledge base combination

Title	Embedding non-ground logic programs into autoepistemic logic for knowledge base combination
Author(s)	Polleres, Axel
Publication Date	2011

Embedding Non-Ground Logic Programs into Autoepistemic Logic for Knowledge Base Combination

JOS DE BRUIJN and THOMAS EITER
Technische Universität Wien
and
AXEL POLLERES
Digital Enterprise Research Institute, National University of Ireland, Galway
and
HANS TOMPITS
Technische Universität Wien

In the context of the Semantic Web, several approaches to the combination of ontologies, given in terms of theories of classical first-order logic and rule bases, have been proposed. They either cast rules into classical logic or limit the interaction between rules and ontologies. Autoepistemic logic (AEL) is an attractive formalism which allows to overcome these limitations, by serving as a uniform host language to embed ontologies and nonmonotonic logic programs into it. For the latter, so far only the propositional setting has been considered. In this paper, we present three embeddings of normal and three embeddings of disjunctive non-ground logic programs under the stable model semantics into first-order AEL. While all embeddings correspond with respect to objective ground atoms, differences arise when considering non-atomic formulas and combinations with first-order theories. We compare the embeddings with respect to stable expansions and autoepistemic consequences, considering the embeddings by themselves, as well as combinations with classical theories. Our results reveal differences and correspondences of the embeddings and provide useful guidance in the choice of a particular embedding for knowledge combination.

Categories and Subject Descriptors: I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation Languages*; *Modal Logic*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Computational Logic*

A preliminary version of this article appeared in: *Proc. 20th International Joint Conference on Artificial Intelligence* (*IJCAI 2007*), AAAI Press, pp. 304–309.

Author's address: J. de Bruijn, T. Eiter, and H. Tompits, Institut für Informationssysteme 184/3, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria; email: `{bruijn,eiter,tompits}@kr.tuwien.ac.at`; A. Polleres, Digital Enterprise Research Institute, National University of Ireland, Galway, IDA Business Park, Lower Dangan Galway, Ireland; email: `axel.polleres@deri.org`.

This work was partially funded by the European Commission projects Knowledge Web (IST 507482), DIP (IST 507483), REWERSE (IST 506779), and ONTORULE (IST 231875), by the Austrian Science Fund (FWF) projects P17212-N04, P21698, and P21840, by the CICyT of Spain project TIC-2003-9001, and by Science Foundation Ireland under the Líon (SFI/02/CE1/I131) and Líon-2 (SFI/08/CE/I1380) grants.

General Terms: Theory, Languages
Additional Key Words and Phrases: First-order autoepistemic logic, knowledge combination, ontologies, rules, stable model semantics

---

1. INTRODUCTION

In the last years, significant effort has been devoted to bring the vision of a Semantic Web closer to reality. Adopting a layered architecture, a number of building blocks have been proposed that serve different purposes, from low-level data encoding to high-level semantic representation. In this architecture, the building blocks for ontologies, rules, and query languages play a prominent role. Furthermore, to ensure interoperability and wide applicability, standard representation formalisms are propagated by the World Wide Web Consortium (W3C), including the Resource Description Framework (RDF) [RDF Concepts 2004; RDF Semantics 2004], the Web Ontology Language (OWL) [OWL Semantics 2004; OWL 2 2009], and the recent Rule Interchange Format Basic Logic Dialect (RIF BLD) [RIF BLD 2009]. In addition, the RIF logical framework [Kifer 2008] lays the foundation for Web rule languages extending RIF BLD with nonmonotonic negation.

Each of these formalisms has a formal semantics, which is either expressible in terms of classical logic or logic programming [de Bruijn and Heymans 2007; Horrocks and Patel-Schneider 2003; Kifer 2008]. There is a need for combining these formalisms, which is illustrated by the following simple example.

Example 1.1. RDF is the basic data description language of the Semantic Web. Atomic statements (*triples*) in RDF are of the form (subject, predicate, object). Subjects and objects, which may be resources on the Web (denoted by URIs), may be shared between triples, yielding a graph-based data model. As demonstrated by de Bruijn and Heymans [2007], the semantics of RDF [RDF Semantics 2004] can be captured in first-order logic, in terms of a formula where sets of such triples are expressed as conjunctions of facts. For purposes of illustration and compatibility with description logics (DL), we use here a simplified notation for RDF triples where class membership (rdf:type) statements are represented using unary predicates, and all other statements using binary predicates. For a full encoding of RDF in first-order logic, de Bruijn and Heymans [2007] use a ternary predicate *triple* to represent all RDF statements (see also Section 7.2).

Consider a fictitious Web site `gangsterepics.com` that publishes information about gangster movies:

$$
\begin{aligned}
&\exists x. \mathit{title}(\text{TheGodFather}, \text{``The Godfather''}) \land \\
&\quad \mathit{title}(\text{TheGodFather2}, \text{``The Godfather: Part II''}) \land \\
&\quad \mathit{title}(\text{PulpFiction}, \text{``Pulp Fiction''}) \land \mathit{director}(\text{PulpFiction}, \text{quentinTarantino}) \land \\
&\quad \mathit{director}(\text{TheGodFather}, x) \land \mathit{director}(\text{TheGodFather2}, x) \land \\
&\quad \mathit{mentionedAt}(\text{PulpFiction}, \text{gangsterepics.com}) \land \\
&\quad \mathit{mentionedAt}(\text{TheGodFather}, \text{gangsterepics.com}) \land \\
&\quad \mathit{mentionedAt}(\text{TheGodFather2}, \text{gangsterepics.com}).
\end{aligned}
$$

The creator of the page did not know the director of "The Godfather" movies, but wanted to express the fact that both parts had the same director. To this end, he used an existentially quantified variable, called "blank node" in RDF. RDF Schema [2004] and OWL have the possibility to express structural relations between predicates. For instance, an OWL (DL) ontology stating that all the movies mentioned at `gangsterepics.com` are either epics or gangster movies, and that *director* is the inverse of *directorOf* can be expressed in terms of DL axioms

$$\begin{aligned} \exists \mathit{mentionedAt}.\{\text{gangsterepics.com}\} &\sqsubseteq \mathit{Epic} \sqcup \mathit{GangsterMovie}, \\ \mathit{director}^{-} &\equiv \mathit{directorOf}, \end{aligned}$$

which may be viewed as a set of first-order logic sentences

$$\begin{aligned} \forall x.\mathit{mentionedAt}(x, \text{gangsterepics.com}) \supset (\mathit{Epic}(x) \vee \mathit{GangsterMovie}(x)), \\ \forall x, y.\mathit{director}(x, y) \equiv \mathit{directorOf}(y, x). \end{aligned}$$

Apart from classical first-order statements, it may be useful to express nonmonotonic information, e.g., that any gangster movie not mentioned on `gangsterepics.com` is an independent movie. For such nonmonotonic statements, logic programming based rules languages that include negation are better suited. That is, we may use the following rule:

$$\mathit{IndieMovie}(x) \leftarrow \mathit{GangsterMovie}(x), \mathit{not}\ \mathit{mentionedAt}(x, \text{gangsterepics.com}).$$

Consider now the following query, which asks for all directors, using the *directorOf* predicate, written using Datalog notation:

$$\mathit{answer}(x) \leftarrow \mathit{directorOf}(y, x).$$

Answering such a query essentially depends on how to interpret and formally combine data (RDF), ontologies (OWL), and rule bases. Given that each of these parts is expressible as either a classical first-order theory or a logic program, the question is how to combine logic programs and classical first-order theories in a unifying logical framework.

A combination of this kind is not obvious, due to the very different setting of classical logic and logic programming [de Bruijn et al. 2006; Motik et al. 2006], and many proposals for combination have been made (we review several of them in Sections 7 and 9). Like in the previous example, an ontology in the form of a classical theory[1] and a logic program should be viewed as complementary descriptions of the same domain. Therefore, a separation between predicates defined in these two components should not be enforced. Furthermore, it is desirable to neither restrict the interaction between the classical and the rules components nor to impose syntactic or semantic restrictions on the individual components. That is, the classical component may be an arbitrary theory Φ of some first-order language with equality, and the rules component may be an arbitrary non-ground normal or disjunctive logic program P, interpreted using, e.g., the common stable model semantics [Gelfond and Lifschitz 1988; 1991].[2] The goal is a combined theory, $\iota(\Phi, P)$,

---

[1] While ontologies are not always first-order definable, for the purpose of this paper we confine ourselves to such ontologies. We note that the Semantic Web ontology languages are first-order definable [Sattler et al. 2003].

[2] For computational reasons, such restrictions (e.g., DL-safety) may be imposed, cf. Sections 7 and 9.

in a uniform logical formalism. Naturally, this theory should amount to Φ if P is empty, and to P if Φ is empty. Therefore, such a combination must provide *faithful embeddings* $\sigma(\Phi)$ and $\tau(P)$ of Φ and P, respectively, such that $\sigma(\Phi) = \iota(\Phi, \emptyset)$ and $\tau(P) = \iota(\emptyset, P)$. In turn, knowledge combination may be carried out on top of $\sigma(\cdot)$ and $\tau(\cdot)$, where in the simplest case one may choose $\iota(\Phi, P) = \sigma(\Phi) \cup \tau(P)$.

This raises the following questions: (a) which uniform formalism is suitable and (b) which embeddings are suitable and, furthermore, how do potentially suitable embeddings relate to each other and behave under knowledge combination?

Concerning the first question, Motik and Rosati [2007] use a variant of Lifschitz's bimodal nonmonotonic *logic of minimal knowledge and negation-as-failure* (MKNF) [Lifschitz 1991]. While the proposed embeddings of the first-order (FO) theory and the logic program are both faithful in the sense described above, the particular combination proposed by Motik and Rosati is only one among many possible methods and MKNF is only one possible underlying formalism for such combinations (we discuss these issues in more detail in Section 8). Indeed, de Bruijn et al. [2007] use quantified equilibrium logic (QEL) [Pearce and Valverde 2005] as a host formalism. Unlike Motik and Rosati, de Bruijn et al. do not propose a new semantics for combinations, but rather show that QEL can capture the semantics of combinations by Rosati [2006] and can be used, for example, to define notions of equivalence of combinations.

Autoepistemic logic (AEL) [Moore 1985], which extends classical logic with a single nonmonotonic modal belief operator, being essentially the nonmonotonic variant of the modal logic **kd45** [Shvarts 1990; Marek and Truszczyński 1993], is an attractive candidate for serving as a uniform host formalism for combinations. Compared to other well-known nonmonotonic formalisms, like Reiter's default logic [Reiter 1980], FO-AEL offers a uniform language in which (nonmonotonic) rules themselves can be expressed at the object level. This conforms with the idea of treating an ontology and a logic program together as a unified theory. Furthermore, in FO-AEL we can decide, depending on the context, whether (the negation of) a particular atomic formula should be interpreted nonmonotonically simply by including a modal operator. This enables us to use the same predicate in both a monotonic and a nonmonotonic context. This is in contrast to circumscription [McCarthy 1986], in which one has to decide, for the entire theory, which predicates are to be minimized.

Embedding a classical theory in AEL is trivial, and several embeddings of logic programs in AEL have been described [Gelfond and Lifschitz 1988; Marek and Truszczynski 1993; Lifschitz and Schwarz 1993; Chen 1993; Przymusinski 1991a]. However, they all have been developed for the propositional case only, whereas we need to deal with non-ground theories and programs. This requires us to consider *first-order autoepistemic logic* (FO-AEL) [Konolige 1991; Kaminski and Rey 2002; Levesque and Lakemeyer 2000], and non-ground versions of these embeddings. We consider the semantics for FO-AEL as defined by Konolige [1991], because it faithfully extends first-order logic with equality (other variants are discussed in Section 8).

Motivated by these issues, our contribution in this paper is twofold:

(1) We define several embeddings of non-ground logic programs into FO-AEL,

taking into account subtle issues of quantification in FO-AEL. In more detail, we present three embeddings, τ_{HP}, τ_{EB}, and τ_{EH}, for normal logic programs which extend respective embeddings for the propositional case [Gelfond 1987; Gelfond and Lifschitz 1988; Marek and Truszczynski 1993; Chen 1993; Lifschitz 1994], and three embeddings, $\tau^{\vee}_{HP}$, $\tau^{\vee}_{EB}$, and $\tau^{\vee}_{EH}$, for disjunctive logic programs, where $\tau^{\vee}_{HP}$ and $\tau^{\vee}_{EH}$ extend embeddings considered in the ground case [Przymusinski 1991a; Marek and Truszczynski 1993]. We show that all these embeddings are faithful in the sense that the stable models of the logic program P and the sets of objective ground atoms in the stable expansions of the embeddings $\tau_{\chi}(P)$ ($\chi \in \{HP, EB, EH\}$) are in a one-to-one correspondence (Theorem 5.3). However, the embeddings behave differently on formulas beyond ground atoms, in some cases already for simple ground formulas. This, in turn, may impact the behavior of the embeddings when used in combinations of logic programs and classical theories. This raises the question under which conditions the embeddings differ and under which conditions they correspond. Of particular interest for knowledge combination is how these embeddings behave relative to each other in combinations with classical theories.

(2) To answer these questions, we conduct two comparative studies of the behavior of the various embeddings. We consider three classes of programs: ground, safe, and arbitrary logic programs under the stable model semantics.
(a) We first determine correspondences between the stable expansions of different embeddings τ_{χ} beyond ground atomic formulas (Propositions 5.5-5.9), and present inclusion relations between the sets of consequences of the embeddings (Theorems 5.14 and 5.15). These results already allow to draw a few conclusions on the behavior of embeddings in combinations.
(b) We then determine correspondences between stable expansions for combinations of logic programs with classical theories. Here, we take the shape of the logic program, the shape of the classical theory, and the type of formulas of interest for the correspondence into account. To this end, we consider different fragments of classical logic that are important for knowledge representation, including Horn, universal, and generalized Horn theories. The latter are of particular interest for ontologies, since they essentially include RDF Schema [de Bruijn and Heymans 2007], Horn-$\mathcal{SHIQ}$ [Hustadt et al. 2005], and the OWL 2 profiles QL, RL, and EL [OWL 2 Profiles 2009]; furthermore, they essentially include also Tuple Generating Dependencies [Abiteboul et al. 1995], which are a popular class of constraints in databases. Our main result for embeddings in combinations (Theorem 6.2) gives a complete picture of the correspondences, which reveals that they behave differently in general, and shows the restrictions on the program or theory that give rise to correspondence.

The results of these studies not only deepen the understanding of the individual embeddings, but also have practical implications with respect to their use. They tell us in which situations one embedding may be used instead of another.

Noticeably, the embeddings of logic programs we study can be seen as building blocks for actual combinations of a classical theory Φ and a logic program P. The most straightforward combination is $\iota(\Phi, P) = \sigma(\Phi) \cup \tau_{\chi}(P)$, where σ is the identity mapping and τ_{χ} is one of the embeddings we consider. One could also imagine adding axioms to, or changing axioms in Φ; similarly, rules could be changed in, or added to P before translating them (e.g., grounding rules as customary in logic programming). If Φ' and P' are the thus obtained classical theory and logic program, our results are still applicable to the combination $\iota'(\Phi, P) = \Phi' \cup \tau_\chi(P')$. In fact, whenever the combination is of the form $\Phi' \cup \tau_\chi(P')$, regardless of Φ and P, the correspondences and differences between the embeddings we establish hold. Furthermore, the effect of different program rewritings P' in combinations may be assessed.

To illustrate the use of our results, we show applications to the Semantic Web. More specifically, we show that the semantics of existing combinations of ontologies and rules in this context can be captured, and that via our correspondence result properties of the semantics can be derived, as well as their behavior in other (modified) combinations. Finally we show how the embeddings we consider can be used to extend combinations to richer languages, particularly extensions of rule languages with nonmonotonic negation. However, while we focus here on the Semantic Web, applications in other contexts (e.g., data modeling languages like UML plus OCL) might be explored.

The remainder of the paper is structured as follows. We review the definitions of first-order logic and logic programs in Section 2. We proceed to describe first-order autoepistemic logic (FO-AEL) and present a novel characterization of stable expansions for certain kinds of theories in Section 3. The embeddings of normal and disjunctive logic programs and our results about faithfulness of the embeddings are described in Section 4. We investigate the relationships between the embeddings themselves, and under combination with first-order theories, in Sections 5 and 6. We discuss applications to the Semantic Web in Section 7 and further implications in Section 8. We discuss related work in Section 9, and conclude and outline future work in Section 10. Proofs of the results in Sections 5 and 6 can be found in the appendix.

2. PRELIMINARIES

Let us briefly recapitulate some basic elements of first-order logic and logic programs as well as some relevant notation.

2.1 First-Order Logic

We consider first-order logic with equality. A language $\mathcal{L}$ is defined over a signature $\Sigma = (\mathcal{F}, \mathcal{P})$, where $\mathcal{F}$ and $\mathcal{P}$ are countable sets of *function* and *predicate symbols*, respectively. Function symbols with arity 0 are also called *constants*. Furthermore, $\mathcal{V}$ is a countably infinite set of *variables*. Terms and atomic formulas (atoms) are constructed as usual. Ground terms are also called *names*; $\mathcal{N}_\Sigma$ denotes the set of names of a given signature Σ. Complex formulas are constructed as usual using the primitive symbols $\neg$, $\wedge$, $\exists$, '(', and ')'. As usual, $\phi \vee \psi$ is short for $\neg(\neg\phi \wedge \neg\psi)$, $\phi \supset \psi$ is short for $\neg\phi \vee \psi$, and $\forall x.\phi(x)$ is short for $\neg\exists x.\neg\phi(x)$. We sometimes write $t_1 \neq t_2$, where t_1 and t_2 are terms, as an abbreviation for $\neg(t_1 = t_2)$. The universal closure of a formula ϕ is denoted by $(\forall)\ \phi$. $\mathcal{L}_g$ is the restriction of $\mathcal{L}$ to ground formulas and $\mathcal{L}_{ga}$ is the restriction of $\mathcal{L}_g$ to atomic formulas. An *FO theory* $\Phi \subseteq \mathcal{L}$ is a set of closed formulas, i.e., every variable is bound by a quantifier.

An *interpretation* of a language $\mathcal{L}$ is a tuple $w = \langle U, \cdot^I \rangle$, where U is a nonempty set, called the *domain*, and $\cdot^I$ is a mapping which assigns to every n-ary function symbol $f \in \mathcal{F}$ a function $f^I : U^n \to U$ and to every n-ary predicate symbol $p \in \mathcal{P}$ a relation $p^I \subseteq U^n$. A *variable assignment* B for w is a mapping that assigns to every variable $x \in \mathcal{V}$ an element $x^B \in U$. A variable assignment B' is an *x-variant* of B if $y^B = y^{B'}$ for every variable $y \in \mathcal{V}$ such that $y \neq x$. The interpretation of a term t, denoted $t^{w,B}$, is defined as usual; if t is ground, we sometimes write t^w.

We call an individual k *named* if there is some name $t \in \mathcal{N}$ such that $t^w = k$, and *unnamed* otherwise. Interpretations are *named* if all individuals are named. We say that the *unique names assumption* applies to an interpretation if all names are interpreted distinctly, and we say that the *standard names assumption* applies if, in addition, the interpretation is named.[3]

A *name substitution* β is a partial function that assigns variables in $\mathcal{V}$ names from $\mathcal{N}$; we also write $x/\beta(x)$ for $(x, \beta(x))$. As usual, β is *total* if its domain is $\mathcal{V}$. Given a variable assignment B for an interpretation w, we define the set of named variables in B as $V_{\mathcal{N}}^{w,B} = \{x \mid x^B \text{ is named}\}$. A substitution β is *associated with* B if its domain is $V_{\mathcal{N}}^{w,B}$ and $x^B = \beta(x)^w$, for each $x \in V_{\mathcal{N}}^{w,B}$. The *application* of a name substitution β to some term, formula, or theory χ, denoted by $\chi\beta$, is defined as syntactical replacement, as usual. Clearly, if the unique names assumption applies, each variable assignment has a unique associated substitution; if the standard names assumption applies, each associated substitution is total.

Example 2.1. Consider a language $\mathcal{L}$ with constants $\mathcal{F} = \{a, b, c\}$, and an interpretation $w = \langle U, \cdot^I \rangle$ with $U = \{k, l, m\}$ such that $a^w = k$, $b^w = l$, and $c^w = l$, and the variable assignment B: $x^B = k$, $y^B = l$, and $z^B = m$. B has two associated name substitutions, $\beta_1 = \{x/a, y/b\}$ and $\beta_2 = \{x/a, y/c\}$, which are both not total.

2.2 Logic Programs

A *disjunctive logic program* P consists of rules of the form

$$h_1 \mid \ldots \mid h_l \;\leftarrow\; b_1,\; \ldots,\; b_m,\; not\; c_1,\; \ldots,\; not\; c_n \tag{1}$$

where $h_1, \ldots, h_l, b_1, \ldots, b_m, 31, \ldots, c_n$ are equality-free atoms, with $m, n \geq 0$ and $l \geq 1$. $H(r) = \{h_1, \ldots, h_l\}$ is the set of *head atoms* 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 *negated body atoms* of r. If $l = 1$, then r is *normal.* 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 normal (resp., positive, safe), then P is normal (resp., positive, safe).

Each program P has a signature Σ_P, which contains the function and predicate symbols that occur in P. We assume that Σ_P contains some 0-ary function symbol if it has predicate symbols of arity greater than 0. With $\mathcal{L}_P$ we denote the first-order language over Σ_P. As usual, *Herbrand interpretations* M of P are subsets of the set of ground atoms of $\mathcal{L}_P$.

The *grounding* of a logic program P, denoted $gr(P)$, is the union of all possible ground instantiations of P, obtained by replacing each variable in a rule r with a name in $\mathcal{N}_{\Sigma_P}$, for each rule $r \in P$.

---

[3] We note here that the term "standard names assumption" is used with various slightly different meanings in the literature; see Section 8 for further discussion.

Let P be a positive program. A Herbrand interpretation M of P is a *Herbrand model* of P if, for every rule $r \in gr(P)$, $B^+(r) \subseteq M$ implies $H(r) \cap M \neq \emptyset$, and, for every $t \in \mathcal{N}_{\Sigma_P}$, $t = t \in M$. A Herbrand model M is *minimal* iff for every model M' such that $M' \subseteq M$, $M' = M$.

Following Gelfond and Lifschitz [1991], the *reduct* of a logic program P with respect to an interpretation M, denoted P^M, is 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.

Example 2.2. Consider the program

$$P = \{p(a);\ p(b);\ q(x) \mid r(x) \leftarrow p(x), not\ s(x)\}$$

and the interpretation $M_1 = \{p(a), p(b), q(a), r(b)\}$.[4] The reduct

$$P^{M_1} = \{p(a); p(b); q(a) \mid r(a) \leftarrow p(a); q(b) \mid r(b) \leftarrow p(b)\}$$

has M_1 as a minimal model, thus M_1 is a stable model of P. The other stable models of P are $M_2 = \{p(a),\ p(b),\ q(a),\ q(b)\}$, $M_3 = \{p(a),\ p(b),\ q(b),\ r(a)\}$, and $M_4 = \{p(a),\ p(b),\ r(a),\ r(b)\}$.

3. FIRST-ORDER AUTOEPISTEMIC LOGIC

We adopt first-order autoepistemic logic (FO-AEL) under the any- and all-name semantics of Konolige [1991]. These semantics allow quantification over arbitrary domains and generalize classical first-order logic with equality, thereby allowing a trivial embedding of first-order theories (with equality). Other approaches like those by Kaminski and Rey [2002] or Levesque and Lakemeyer [2000] require interpretations to follow the unique or standard names assumptions and therefore do not allow such direct embeddings.

An *FO-AEL language* $\mathcal{L}_{\mathsf{L}}$ is defined relative to a first-order language $\mathcal{L}$ by allowing the unary modal operator L in the construction of formulas—$\mathsf{L}\phi$ is usually read as "ϕ is known" or "ϕ is believed". As usual, closed formulas, i.e., formulas without free variable occurrences, are called *sentences*; formulas of the form $\mathsf{L}\phi$, where ϕ is a formula, are *modal atoms*; and L-free formulas are *objective. Standard autoepistemic logic* is variable-free FO-AEL.

To distinguish between semantic notions defined for the any- resp. all-name semantics, we use the symbols E ("*E*xistence of name") and A ("for *A*ll names") in the respective notations.

An *autoepistemic interpretation* is a pair $\langle w, \Gamma\rangle$, where $w = \langle U, \cdot^I\rangle$ is a first-order interpretation and $\Gamma \subseteq \mathcal{L}_{\mathsf{L}}$ is a set of sentences, called a *belief set.* Satisfaction of a formula $\mathsf{L}\phi$ in an interpretation $\langle w, \Gamma\rangle$ with respect to a variable assignment B under the *any-name semantics*, denoted $(w, B) \models^{\mathrm{E}}_{\Gamma} \mathsf{L}\phi$, is defined as

> $(w, B) \models^{\mathrm{E}}_{\Gamma} \mathsf{L}\phi$ iff, for *some* name substitution β associated with B, $\phi\beta$ is closed and $\phi\beta \in \Gamma$.

Satisfaction of arbitrary formulas is then as follows, where $\phi, \psi \in \mathcal{L}_{\mathsf{L}}$:

---

[4] For brevity, we leave out equality atoms in the example.

—$(w, B) \models^{\mathrm{E}}_{\Gamma} p(t_1, \ldots, t_n)$ iff $(t_1^{w,B}, \ldots, t_n^{w,B}) \in p^I$;
—$(w, B) \models^{\mathrm{E}}_{\Gamma} t_1 = t_2$ iff $t_1^{w,B} = t_2^{w,B}$;
—$(w, B) \models^{\mathrm{E}}_{\Gamma} \neg\phi$ iff $(w, B) \not\models^{\mathrm{E}}_{\Gamma} \phi$;
—$(w, B) \models^{\mathrm{E}}_{\Gamma} \phi \wedge \psi$ iff $(w, B) \models^{\mathrm{E}}_{\Gamma} \phi$ and $(w, B) \models^{\mathrm{E}}_{\Gamma} \psi$;
—$(w, B) \models^{\mathrm{E}}_{\Gamma} \exists x.\phi$ iff for some x-variant B' of B, $(w, B') \models^{\mathrm{E}}_{\Gamma} \phi$.

An interpretation $\langle w, \Gamma\rangle$ is a *model* of ϕ, denoted $w \models^{\mathrm{E}}_{\Gamma} \phi$, if $(w, B) \models^{\mathrm{E}}_{\Gamma} \phi$ for every variable assignment B for w. This extends to sets of formulas in the usual way. A set of formulas $\Phi \subseteq \mathcal{L}_{\mathsf{L}}$ *entails* a formula $\phi \in \mathcal{L}_{\mathsf{L}}$ with respect to a belief set Γ, denoted $\Phi \models^{\mathrm{E}}_{\Gamma} \phi$, if for every interpretation w such that $w \models^{\mathrm{E}}_{\Gamma} \Phi$, $w \models^{\mathrm{E}}_{\Gamma} \phi$.

The notions of satisfaction and entailment under the *all-name semantics*, for which we use the symbol $\models^{\mathrm{A}}_{\Gamma}$, are analogously defined, with the only difference that satisfaction of modal atoms is subject to the following condition:

> $(w, B) \models^{\mathrm{A}}_{\Gamma} \mathsf{L}\phi$ iff, for *all* name substitutions β associated with B, $\phi\beta$ is closed and $\phi\beta \in \Gamma$.

Note that the any- and all-name semantics always coincide for objective formulas and, if the unique (or standard) names assumption applies, also for arbitrary formulas in $\mathcal{L}_{\mathsf{L}}$; this was also observed by Kaminski and Rey [2002]. In such situations, i.e., where both semantics coincide, we sometimes use $\models_{\Gamma}$ rather than $\models^{\mathrm{E}}_{\Gamma}$ or $\models^{\mathrm{A}}_{\Gamma}$. Furthermore, when talking about entailment $\Phi \models_{\Gamma} \phi$ under the standard names assumption, we mean entailment considering only interpretations for which the standard names assumption holds. That is, $\Phi \models_{\Gamma} \phi$ under the standard names assumption if for every interpretation w such that the standard names assumption applies in w and $w \models_{\Gamma} \Phi$, $w \models_{\Gamma} \phi$.

Example 3.1. Consider the formula $\phi = \forall x(p(x) \supset \mathsf{L}p(x))$ and some interpretation $\langle w, \Gamma\rangle$. Then, $w \models^{\mathrm{E}}_{\Gamma} \phi$ iff, for every variable assignment B, $(w, B) \models^{\mathrm{E}}_{\Gamma} p(x) \supset \mathsf{L}p(x)$, which in turn holds iff $(w, B) \not\models^{\mathrm{E}}_{\Gamma} p(x)$ or $(w, B) \models^{\mathrm{E}}_{\Gamma} \mathsf{L}p(x)$. Now, $(w, B) \models^{\mathrm{E}}_{\Gamma} \mathsf{L}p(x)$, with $x^B = k$, iff, for some $t \in \mathcal{N}_{\Sigma}$, $t^w = k$, and $p(t) \in \Gamma$. Thus, ϕ is false (unsatisfied) in any interpretation where p^I contains unnamed individuals. Analogous for the all-name semantics.

The following example illustrates the difference between the any- and all-name semantics.

Example 3.2. Consider a language with constant symbols a, b and unary predicate symbol p, and an interpretation $\langle w, \Gamma\rangle$ with $w = \langle\{k\}, \cdot^I\rangle$ and $\Gamma = \{p(a)\}$. Then, $w \models^{\mathrm{E}}_{\Gamma} \exists x.\mathsf{L}p(x)$, while $w \not\models^{\mathrm{A}}_{\Gamma} \exists x.\mathsf{L}p(x)$, since $b^w = a^w = k$ but $p(b) \notin \Gamma$.

A stable expansion is a set of beliefs of an ideally introspective agent (i.e., an agent with perfect reasoning capabilities and with knowledge about its own beliefs), given some theory $\Phi \subseteq \mathcal{L}_{\mathsf{L}}$. Formally, a belief set $T \subseteq \mathcal{L}_{\mathsf{L}}$ is a *stable*$^{\mathrm{E}}$ *expansion* of a theory $\Phi \subseteq \mathcal{L}_{\mathsf{L}}$ iff $T = \{\phi \mid \Phi \models^{\mathrm{E}}_{T} \phi\}$. Similarly, T is a *stable*$^{\mathrm{A}}$ *expansion* of Φ iff $T = \{\phi \mid \Phi \models^{\mathrm{A}}_{T} \phi\}$.

Recall that $\mathcal{L}_g$ and $\mathcal{L}_{ga}$ denote the restrictions of $\mathcal{L}$ to ground and ground atomic formulas, respectively. Given a set of sentences $\Gamma \subseteq \mathcal{L}_{\mathsf{L}}$, Γ_o, Γ_{og}, and Γ_{oga} denote the restrictions of Γ to objective, objective ground, and objective ground atomic formulas, respectively, i.e., $\Gamma_o = \Gamma \cap \mathcal{L}$, $\Gamma_{og} = \Gamma \cap \mathcal{L}_g$, and $\Gamma_{oga} = \Gamma \cap \mathcal{L}_{ga}$.

Every stable expansion T of Φ is a *stable set* [Stalnaker 1993], which means that it satisfies the following conditions: (a) T is closed under first-order entailment, (b) if $\phi \in T$ then $\mathsf{L}\phi \in T$, and (c) if $\phi \notin T$ then $\neg\mathsf{L}\phi \in T$. Furthermore, if T is consistent, the converse statements of (b) and (c) hold.

Konolige [1991] shows that a stable expansion T of a theory $\Phi \subseteq \mathcal{L}_\mathsf{L}$ is determined by its objective subset T_o, also called the *kernel* of T. He further obtained the following result:

Proposition 3.3 [Konolige 1991]. *Let $\Phi \subseteq \mathcal{L}_\mathsf{L}$ be a theory without nested modal operators, $\Gamma \subseteq \mathcal{L}$ a set of objective formulas, and $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. Then, $\Gamma = \{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma} \phi\}$ iff $\Gamma = T_o$, for some stable$^{\mathrm{X}}$ expansion T of Φ.*

We slightly adapt this result as follows:

Proposition 3.4. *Let $\Phi \subseteq \mathcal{L}_\mathsf{L}$ be a theory with only objective atomic formulas in the scope of occurrences of L, $\Gamma \subseteq \mathcal{L}$ a set of objective formulas, and $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. Then, $\Gamma = \{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma_{oga}} \phi\}$ iff $\Gamma = T_o$, for some stable$^{\mathrm{X}}$ expansion T of Φ.*

Proof. Since modal atoms in Φ contain only objective atomic formulas, we obtain $\Phi \models^{\mathrm{X}}_{\Gamma_o} \phi$ iff $\Phi \models^{\mathrm{X}}_{\Gamma_{oga}} \phi$, because, by the definition of satisfaction of modal formulas, non-ground and non-atomic formulas in Γ_o do not affect satisfaction of formulas in Φ. Thus,$\{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma_{oga}} \phi\} = \{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma_o} \phi\}$ follows.

Since there is no nesting of modal operators in Φ, we combine this result with Proposition 3.3 to obtain $\Gamma_o = \{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma_o} \phi\} = \{\phi \in \mathcal{L} \mid \Phi \models^{\mathrm{X}}_{\Gamma_{oga}} \phi\}$ iff $\Gamma_o = T \cap \mathcal{L}$ is the kernel of a stable$^{\mathrm{X}}$ expansion T of Φ. □

We note here that, unlike in standard autoepistemic logic, in FO-AEL two different stable expansions may have the same objective subsets, both under the any- and all-name semantics. Consider, for example, the theories $\Phi = \{\forall x.p(x)\}$ and $\Phi' = \{\forall x.\mathsf{L}p(x)\}$ and their respective stable expansions T and T'. We have that $T_o = T'_o$ is the closure under first-order entailment of $\{\forall x.p(x)\}$, but we also have that $\forall x.\mathsf{L}p(x) \in T'$ but $\forall x.\mathsf{L}p(x) \notin T$, because $\forall x.\mathsf{L}p(x)$ is not satisfied in any interpretation that has unnamed individuals.

4. EMBEDDING NON-GROUND LOGIC PROGRAMS

We define an embedding τ as a function that takes a logic program P as its argument and returns a set of sentences in the FO-AEL language obtained from Σ_P.

Janhunen [1999] studied translations between nonmonotonic formalisms and formulated a number of desiderata for such translation functions, namely faithfulness, polynomiality, and modularity (FPM). We adapt these notions to our case of embedding logic programs into FO-AEL.

An embedding τ is *faithful* if, for any logic program P, there is a one-to-one correspondence between the stable models of P and the consistent stable expansions of $\tau(P)$ with respect to ground atomic formulas.

An embedding τ is *polynomial* if, for any logic program P, $\tau(P)$ can be computed in time polynomial in the size of P.

An embedding τ is *modular* if, for any two logic programs P_1 and P_2, $\tau(P_1 \cup P_2) = \tau(P_1) \cup \tau(P_2)$. Furthermore, we call τ *signature-modular* if, for any two logic programs P_1 and P_2 with the same signature Σ, $\tau(P_1 \cup P_2) = \tau(P_1) \cup \tau(P_2)$.

Since the unique names assumption does not hold in FO-AEL in general, it is necessary to axiomatize default uniqueness of names (as introduced by Konolige [1991]) to assure faithfulness of several of the embeddings. Given a signature Σ, by UNA_Σ we denote the set of axioms

UNA. $\neg \mathsf{L}(t_1 = t_2) \supset t_1 \neq t_2$, for all distinct $t_1, t_2 \in \mathcal{N}_\Sigma$.

Default uniqueness, in contrast to rigid uniqueness (i.e., UNA axioms of the form $t_1 \neq t_2$), allows first-order theories that are later combined with the embedding to "override" such inequalities, rather than introducing inconsistency. For example, the theory $\Phi = \{\neg \mathsf{L}(a = b) \supset a \neq b\}$ has a single expansion that includes $a \neq b$; the single expansion of $\Phi \cup \{a = b\}$ is consistent and includes $a = b$.

Observe that the UNA axioms depend on the signature. In addition, the union of the UNA axioms of two signatures is not necessarily the same as the set of UNA axioms of the union of these two signatures: given two signatures Σ_1 and Σ_2 such that $\mathcal{F}_1 \neq \mathcal{F}_2$, $\mathrm{UNA}_{\Sigma_1} \cup \mathrm{UNA}_{\Sigma_2} \neq \mathrm{UNA}_{\Sigma_1 \cup \Sigma_2}$, i.e., the UNA axioms corresponding to different signatures cannot be combined in a modular fashion. This means that embeddings that include such UNA signatures are not modular, but may be signature-modular.

We first present the embeddings of normal programs and then proceed with the embeddings of disjunctive programs.

4.1 Embedding Normal Logic Programs

We consider three embeddings of non-ground logic programs into FO-AEL, denoted τ_{HP}, τ_{EB}, and τ_{EH}. "*HP*" stands for "*H*orn for *P*ositive rules" (positive rules are translated to objective Horn clauses); "*EB*" stands for "*E*pistemic rule *B*odies" (the body of a rule can only become true if it is *known* to be true); and "*EH*" stands for "*E*pistemic rule *H*eads" (if the body of a rule is true, the head is *known* to be true).

The *HP* embedding is an extension of the one which originally led Gelfond and Lifschitz to the definition of the stable model semantics [Gelfond 1987; Gelfond and Lifschitz 1988]. The *EB* and *EH* embeddings are extensions of embeddings by Marek and Truszczynski [1993]. The *EH* embedding was independently described by Lifschitz and Schwarz [1993] and by Chen [1993]. The original motivation for the *EB* and *EH* embeddings was the possibility to directly embed programs with strong negation and disjunction. Furthermore, Marek and Truszczynski arrived at their embeddings through embeddings of logic programs in *reflexive autoepistemic logic* [Schwarz 1992], which is equivalent to McDermott's nonmonotonic modal **sw5** [McDermott 1982], and the subsequent embedding of reflexive autoepistemic logic into standard AEL. Lifschitz and Schwarz arrived at the *EH* embedding through an embedding of logic programs in Lifschitz's nonmonotonic logic of *minimal belief and negation-as-failure* (MBNF) [Lifschitz 1994] and the subsequent embedding of MBNF into standard AEL. Finally, Chen also arrived at the *EH* embedding via MBNF, but he subsequently embedded MBNF in Levesque's *logic of only knowing* [Levesque 1990], a subset of which corresponds with standard AEL.

Definition 4.1. Let r be a normal rule of the form (1). Then,

$$\begin{aligned}\tau_{HP}(r) &= (\forall)\ \textstyle\bigwedge_i b_i \wedge \bigwedge_j \neg \mathsf{L} c_j \supset h_1,\\ \tau_{EB}(r) &= (\forall)\ \textstyle\bigwedge_i (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_j \neg \mathsf{L} c_j \supset h_1,\\ \tau_{EH}(r) &= (\forall)\ \textstyle\bigwedge_i (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_j \neg \mathsf{L} c_j \supset h_1 \wedge \mathsf{L} h_1.\end{aligned}$$

Furthermore, given a normal logic program P, we define:

$$\tau_\chi(P) = \{\tau_\chi(r) \mid r \in P\} \cup \mathrm{UNA}_{\Sigma_P}, \quad \chi \in \{HP, EB, EH\}.$$

For all three embeddings, we assume $\Sigma_{\tau_\chi(P)} = \Sigma_P$ (here and henceforth "χ" ranges over HP, EB, and EH). Furthermore, by τ_χ^- we denote the embedding τ_χ *without* the UNA axioms: given a normal logic program P, $\tau_\chi^-(P) = \tau_\chi(P) - \mathrm{UNA}_{\Sigma_P}$. The embeddings τ_χ^- are modular and polynomial. The embeddings τ_χ are signature-modular and polynomial, provided $\mathcal{N}_{\Sigma_P}$ is polynomial in the size of P (e.g., if there are no function symbols with arity greater than 0). In the examples of embeddings in the remainder of the paper we do not write the UNA axioms explicitly.

A notable difference between the embedding τ_{HP}, on the one hand, and the embeddings τ_{EB} and τ_{EH}, on the other, is that, given a logic program P, the stable expansions of $\tau_{HP}(P)$ include the "contrapositives" of the rules in P (viewed classically and where $\neg \mathsf{L} a$ is *not* a), which is not true for $\tau_{EB}(P)$ and $\tau_{EH}(P)$ in general.

Example 4.2. Consider $P = \{p \leftarrow q, not\ r\}$. The stable expansion of $\tau_{HP}(P) = \{q \wedge \neg \mathsf{L} r \supset p\}$ includes $\neg p \supset \neg q \vee \mathsf{L} r$; the expansion of $\tau_{EB}(P) = \{q \wedge \mathsf{L} q \wedge \neg \mathsf{L} r \supset p\}$ includes $\neg p \supset \neg \mathsf{L} q \vee \neg q \vee \mathsf{L} r$, but not $\neg p \supset \neg q \vee \mathsf{L} r$.

For standard AEL and ground logic programs, the following faithfulness result straightforwardly extends results by Gelfond and Lifschitz [1988] and Marek and Truszczynski [1993].

PROPOSITION 4.3. *A Herbrand interpretation M of a ground normal logic program P is a stable model of P iff there exists a consistent stable expansion T of $\tau_\chi^-(P)$ in standard AEL such that $M = T \cap \mathcal{L}_{ga}$.*

Observe from the proposition that we do not require the UNA axioms in the embeddings of ground programs. These axioms are required in the general case when embedding non-ground programs, as illustrated by Example 4.8 below. The following example illustrates the embeddings for the case of non-ground programs.

Example 4.4. Consider $P = \{q(a);\ p(x);\ r(x) \leftarrow not\ s(x),\ p(x)\}$, which has the single stable model $M = \{q(a), p(a), r(a)\}$. Likewise, each of the embeddings $\tau_\chi(P)$ has a single consistent stable expansion T^χ:

$$\begin{aligned}T^{HP} &= \{q(a), p(a), \mathsf{L}p(a), \ldots, \forall x(p(x)), \neg\mathsf{L}\forall x(\mathsf{L}p(x)), \forall x(\neg\mathsf{L}s(x) \supset r(x)), \ldots\},\\ T^{EB} &= \{q(a), p(a), \mathsf{L}p(a), \ldots, \forall x(p(x)), \neg\mathsf{L}\forall x(\mathsf{L}p(x)), \neg\mathsf{L}(\forall x(\neg\mathsf{L}s(x) \supset r(x))), \ldots\},\\ T^{EH} &= \{q(a), p(a), \mathsf{L}p(a), \ldots, \forall x(p(x)), \forall x(\mathsf{L}p(x)), \forall x(\neg\mathsf{L}s(x) \supset r(x)), \ldots\}.\end{aligned}$$

The stable expansions in Example 4.4 agree on objective ground atoms, but not on arbitrary formulas. We now extend Proposition 4.3 to the non-ground case. To this end, we use the following two lemmas.

Lemma 4.5. *Let P be a normal logic program, let* $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$*, let* T *be a stable*$^{\mathrm{X}}$ *expansion of* $\tau_{\mathcal{X}}(P)$*, and let* α *be an objective ground atom. Then,* $\tau_{\mathcal{X}}(P) \models^{\mathrm{X}}_{T_{oga}} \alpha$ *iff* $\tau_{\mathcal{X}}(P) \models_{T_{oga}} \alpha$ *under the standard names assumption. Moreover,* $\tau^{-}_{HP}(P) \models^{\mathrm{A}}_{T_{oga}} \alpha$ *iff* $\tau^{-}_{HP}(P) \models_{T_{oga}} \alpha$ *under the standard names assumption.*

Proof. We start with the first statement.
($\Rightarrow$) This is obvious, as interpretations under the standard names assumption are just special interpretations.

($\Leftarrow$) We start with the case of the any-names assumption. Assume, on the contrary, that $\tau_{\mathcal{X}}(P) \models_{T_{oga}} \alpha$ under the standard names assumption, but $\tau_{\mathcal{X}}(P) \not\models^{\mathrm{E}}_{T_{oga}} \alpha$. This means that there is some interpretation $w = \langle U, \cdot^{I} \rangle$ such that $w \models^{\mathrm{E}}_{T_{oga}} \tau_{\mathcal{X}}(P)$, but $w \not\models^{\mathrm{E}}_{T_{oga}} \alpha$.

By the fact that the only occurrences of the equality symbol in $\tau_{\mathcal{X}}(P)$ are in the UNA axioms, the only atoms in T_{oga} involving equality are of the form $t = t$, for $t \in \mathcal{N}_{\Sigma_P}$. Consider two distinct names $t_1, t_2 \in \mathcal{N}_{\Sigma_P}$ and the UNA axiom $\neg \mathsf{L} t_1 = t_2 \supset t_1 \neq t_2 \in \mathrm{UNA}_{\Sigma_P}$. Since $\langle w, T_{oga} \rangle$ is a model of the axiom and $t_1 = t_2 \notin T_{oga}$, $w \models^{\mathrm{E}}_{T_{oga}} t_1 \neq t_2$. Consequently, it must be the case that $\cdot^{I}$ maps every name to a distinct individual in U.

We assume that the mapping $\cdot^{I}$ extends to ground terms in the natural way, i.e., $f(t_1, \ldots, t_m)^{I} = f^{I}(t_1^{I}, \ldots, t_m^{I})$. We construct the interpretation $w' = \langle U', \cdot^{I'} \rangle$ as follows: $U' = \mathcal{N}$, $t^{I'} = t$, for $t \in \mathcal{N}$, and $\langle t_1, \ldots, t_n \rangle \in p^{I'}$ if $\langle t_1^{I}, \ldots, t_n^{I} \rangle \in p^{I}$ for n-ary predicate symbol p and every $\langle t_1, \ldots, t_n \rangle \in \mathcal{N}^n$. Clearly, the standard names assumption holds for w', and w and w' agree on objective ground atoms, i.e., $w \models \alpha$ iff $w' \models \alpha$ for any $\alpha \in \mathcal{L}_{ga}$. We now show that $w' \models_{T_{oga}} \tau_{\mathcal{X}}(P)$.

Clearly, $\langle w', T_{oga} \rangle$ satisfies the UNA axioms since the standard names assumption holds for w' and since T_{oga} contains only the trivial equalities. We first consider the embedding τ_{EH} and some

$$(\forall) \quad \textstyle\bigwedge_{1 \leq i \leq m} (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_{1 \leq j \leq n} (\neg \mathsf{L} c_j) \supset h_1 \wedge \mathsf{L} h_1 \in \tau_{EH}(P).$$

Since $w \models_{T_{oga}} \tau_{EH}(P)$,

$$(w, B) \models_{T_{oga}} \textstyle\bigwedge_{1 \leq i \leq m} (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_{1 \leq j \leq n} (\neg \mathsf{L} c_j) \supset h_1 \wedge \mathsf{L} h_1$$

for every variable assignment B of w.

Now, consider a variable assignment B' of w' and the corresponding variable assignment B of w, which we define as follows: $x^{B} = k$ iff there is a $t \in \mathcal{N}_{\Sigma_P}$ such that $x^{B'} = t$ and $t^{I} = k$. Observe that B assigns every variable to a named individual. Consider a name substitution β which is associated with B; since all names are interpreted as distinct individuals (by the UNA axioms), (†) β is unique. Moreover, by construction of B, β is also the only substitution associated with B' .

By construction of w', and since β is the unique substitution associated with B (and B'), we have, for every objective atom α such that B is defined for all variables in α, that $(w, B) \models^{\mathrm{E}}_{T_{oga}} \alpha$ iff $(w', B') \models^{\mathrm{E}}_{T_{oga}} \alpha$ and $(w, B) \models^{\mathrm{E}}_{T_{oga}} \mathsf{L}\alpha$ iff $(w', B') \models^{\mathrm{E}}_{T_{oga}} \mathsf{L}\alpha$. Consequently, if $(w, B) \models^{\mathrm{E}}_{T_{oga}} h_1 \wedge \mathsf{L} h_1$, then $(w', B') \models_{T_{oga}} h_1 \wedge \mathsf{L} h_1$, and if $(w, B) \not\models^{\mathrm{E}}_{T_{oga}} \bigwedge b_i \wedge \mathsf{L} b_i$, then $(w', B') \not\models_{T_{oga}} \bigwedge b_i \wedge \mathsf{L} b_i$. Furthermore, $(w, B) \not\models^{\mathrm{E}}_{T_{oga}} \bigwedge \neg \mathsf{L} c_j$ implies $c_i \beta \in T_{oga}$ for some $i \in \{1, \ldots, n\}$. Hence, $(w', B') \not\models_{T_{oga}} \neg \mathsf{L} c_1 \wedge \cdots \wedge \neg \mathsf{L} c_n$. So,

$$(w', B') \models_{T_{oga}} b_1 \wedge \mathsf{L}b_1 \wedge \cdots \wedge b_m \wedge \mathsf{L}b_m \wedge \neg\mathsf{L}c_1 \wedge \cdots \wedge \neg\mathsf{L}c_n \supset h_1 \wedge \mathsf{L}h_1.$$

Thus, we obtain $w' \models_{T_{oga}} \tau_{EH}(P)$. Since w and w' agree on objective ground atoms, $w' \not\models_{T_{oga}} \alpha$, and thus $\tau_{EH}(P) \not\models_{T_{oga}} \alpha$ under the standard names assumption. This contradicts the initial assumption. Therefore, $\tau_{EH}(P) \models^{\mathrm{E}}_{T_{oga}} \alpha$.

The argument for the embeddings τ_{EB} and τ_{HP} is analogous: simply leave out the positive occurrences of modal atoms in the consequents, respectively consequents and antecedents, in the argument above.

Likewise, the argument for the case of the all-name semantics is analogous. Observe that in the argument about variable assignments (†), β is the only name substitution associated with B; hence, the any- and all-name semantics coincide, and the subsequent arguments immediately apply also for the all-name semantics.

For the second statement, consider the above argument without the part about the UNA axioms and the following simple adaptation: if $(w, B) \not\models^{\mathrm{A}}_{T_{oga}} \bigwedge \neg\mathsf{L}c_j$, then for all associated name substitutions β, there is some $c_i\beta \in T_{oga}$, $1 \leq i \leq n$. One of these name substitutions is the one associated with B'; the remainder of the argument remains the same. It follows that $\tau^{-}_{HP}(P) \models^{\mathrm{A}}_{T_{oga}} \alpha$ iff $\tau^{-}_{HP}(P) \models_{T_{oga}} \alpha$. □

The latter fails for the embeddings τ^{-}_{EB} and τ^{-}_{EH}, as there may be several name substitutions associated with the assignment B in the "⇐" direction above, while there is a single substitution associated with B' (see also Example 4.8).

Lemma 4.6. *Let P be a normal logic program and $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. There exists a stable$^{\mathrm{X}}$ expansion T of $\tau_{\chi}(P)$ iff there exists a stable$^{\mathrm{X}}$ expansion T' of $\tau_{\chi}(gr(P))$ such that $T'_{oga} = T_{oga}$. The same result holds for τ^{-}_{HP} and stable$^{\mathrm{A}}$ expansions.*

Proof. We prove the first statement, first for the special case that the standard names assumption applies, and then use Lemma 4.5 to extend it to cases where the standard names assumption does not apply.

Consider a belief set $\Gamma \subseteq \mathcal{L}_{\mathsf{L}}$ and an interpretation w for which the standard names assumption holds. We claim that $(*)$ $w \models^{\mathrm{X}}_{\Gamma} \tau_{\chi}(gr(P))$ iff $w \models^{\mathrm{X}}_{\Gamma} \tau_{\chi}(P)$. By the standard names assumption, we have that $w \models^{\mathrm{X}}_{\Gamma} \tau_{\chi}(P)$ iff for every $\phi \in \tau_{\chi}(P)$, $w \models^{\mathrm{X}}_{\Gamma} \phi$. In turn, this holds iff for every variable assignment B, $(w, B) \models^{\mathrm{X}}_{\Gamma} \phi$, which in turn holds iff for the name substitution β associated with B (which is unique and total, by the standard names assumption), $w \models^{\mathrm{X}}_{\Gamma} \phi\beta$. By definition, $\tau_{\chi}(gr(P))$ contains all (and only) the formulas of the form $\phi\beta$ where $\phi \in \tau_{\chi}(P)$ and β is a name substitution associated with some variable assignment B for w; the claim $(*)$ follows immediately from this.

(⇒) Let T be a stable expansion of $\tau_{\chi}(P)$. By Lemma 4.5 and the above we have:

$$\{\phi \in \mathcal{L}_{ga} \mid \tau_{\chi}(P) \models^{\mathrm{X}}_{T_{oga}} \phi\} = \{\phi \in \mathcal{L}_{ga} \mid \tau_{\chi}(gr(P)) \models^{\mathrm{X}}_{T_{oga}} \phi\}.$$

Hence by Proposition 3.4,

$$T'_o = \{\phi \in \mathcal{L} \mid \tau_{\chi}(gr(P)) \models^{\mathrm{X}}_{T_{oga}} \phi\}$$

is the kernel of a stable expansion T' of $\tau_{\chi}(gr(P))$ and $T' \cap \mathcal{L}_{ga} = T_{ga}$.

The converse is analogous. For the second statement of the lemma, the same proof using Lemma 4.5 works. □

Theorem 4.7. *Let P be a normal logic program and $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. A Herbrand interpretation M is a stable model of P iff there exists a consistent stable$^{\mathrm{X}}$ expansion T of $\tau_{\chi}(P)$ such that $M = T_{oga}$. The same result holds for τ_{HP}^{-} and stable$^{\mathrm{A}}$ expansions.*

Proof. By Lemma 4.6, we can reduce embeddability of non-ground logic programs to embeddability of ground logic programs.

Consider an embedding $\tau_{\chi}(gr(P))$ and a stable expansion T. Clearly, there is no interaction between the UNA axioms and the axioms resulting from rules in P. Therefore, $\tau_{\chi}^{-}(gr(P))$ has a stable expansion T' such that $T'_{oga} = T_{oga}$, and vice versa. The theorem then follows immediately from Proposition 4.3. □

Note that this result does not extend to the embeddings τ_{χ}^{-} under the any-name semantics, nor does it extend to the embeddings τ_{EB}^{-} and τ_{EH}^{-} under the all-name semantics, as illustrated by the following example.

Example 4.8. Consider $P = \{p(n_1); r(n_2); q \leftarrow \mathit{not}\ p(x)\}$ such that Σ_P has only two names, n_1 and n_2. P has one stable model, $M = \{p(n_1), r(n_2), q\}$. $\tau_{HP}^{-}(P) = \{p(n_1); r(n_2); \forall x(\neg \mathsf{L}p(x) \supset q)\}$ has one stable$^{\mathrm{E}}$ expansion, $T = \{p(n_1), r(n_2), \mathsf{L}p(n_1), \mathsf{L}r(n_2), \neg \mathsf{L}p(n_2), \dots\}$. T does *not* include q. To see why this is the case, consider an interpretation w with only one individual k. $\mathsf{L}p(x)$ is trivially true under the any-name semantics, because there is some name for k such that $p(t) \in T$ (viz. $t = n_1$). In the all-name semantics, this situation does not occur, because for $\mathsf{L}p(x)$ to be true, $p(t)$ must be included in T for *every* name ($t = n_1$ and $t = n_2$) for k. One can similarly verify that the result does not apply to the embeddings τ_{EB}^{-} and τ_{EH}^{-} under the all-name semantics, by the positive modal atoms in the antecedents.

4.2 Embedding Disjunctive Logic Programs

The embeddings τ_{HP} and τ_{EB} cannot be straightforwardly extended to the case of disjunctive logic programs, even for the propositional case. Consider the program $P = \{a \mid b \leftarrow\}$, which has two stable models: $M_1 = \{a\}$ and $M_2 = \{b\}$. However, a naive extension of τ_{HP}, $\tau_{HP}(P) = \{a \vee b\}$, has one stable expansion $T = \{a \vee b, \mathsf{L}(a \vee b), \neg \mathsf{L}a, \neg \mathsf{L}b, \dots\}$. In contrast, τ_{EH} can be straightforwardly extended because of the modal atoms in the consequent of the implication: $\tau_{EH}^{\vee}(P) = \{(a \wedge \mathsf{L}a) \vee (b \wedge \mathsf{L}b)\}$ has two stable expansions $T_1 = \{a \vee b, a, \mathsf{L}a, \neg \mathsf{L}b, \dots\}$ and $T_2 = \{a \vee b, b, \mathsf{L}b, \neg \mathsf{L}a, \dots\}$.

The so-called *positive introspection axioms* (PIAs) [Przymusinski 1991a] remedy this situation for defining extensions $\tau_{HP}^{\vee}$ and $\tau_{EB}^{\vee}$ of τ_{HP} and τ_{EB}, respectively. Let PIA_{Σ} be the set of axioms

PIA. $\alpha \supset \mathsf{L}\alpha$, for every objective ground atom α of $\mathcal{L}_{\Sigma}$.

Each PIA ensures that a consistent stable expansion contains either α or $\neg\alpha$.

It would have been possible to define the PIAs in a different way: $(\forall)\ \phi \supset \mathsf{L}\phi$ for any objective atomic formula ϕ. This would, however, effectively close the domain of the predicates in Σ_P (see Example 3.1). We deem this aspect undesirable in combinations with FO theories.

Definition 4.9. Let r be a rule of form (1). Then:

$$\tau^{\vee}_{HP}(r) = (\forall)\ \textstyle\bigwedge_i b_i \wedge \bigwedge_j \neg \mathsf{L} c_j \supset \bigvee_k h_k,$$
$$\tau^{\vee}_{EB}(r) = (\forall)\ \textstyle\bigwedge_i (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_j \neg \mathsf{L} c_j \supset \bigvee_k h_k,$$
$$\tau^{\vee}_{EH}(r) = (\forall)\ \textstyle\bigwedge_i (b_i \wedge \mathsf{L} b_i) \wedge \bigwedge_j \neg \mathsf{L} c_j \supset \bigvee_k (h_k \wedge \mathsf{L} h_k).$$

Given a disjunctive logic program P, we define:

$$\tau^{\vee}_{HP}(P) = \{\tau^{\vee}_{HP}(r) \mid r \in P\} \cup \mathrm{PIA}_{\Sigma_P} \cup \mathrm{UNA}_{\Sigma_P},$$
$$\tau^{\vee}_{EB}(P) = \{\tau^{\vee}_{EB}(r) \mid r \in P\} \cup \mathrm{PIA}_{\Sigma_P} \cup \mathrm{UNA}_{\Sigma_P},$$
$$\tau^{\vee}_{EH}(P) = \{\tau^{\vee}_{EH}(r) \mid r \in P\} \cup \mathrm{UNA}_{\Sigma_P}.$$

As before, by $\tau^{\vee -}_{\chi}$ we denote the embedding $\tau^{\vee}_{\chi}$ *without* the UNA axioms. Note that the observations about modularity of the embeddings τ_{χ} extend to the disjunctive embeddings $\tau^{\vee}_{\chi}$; the PIAs do not compromise modularity. However, polynomiality of embeddings with PIAs is lost if the size of $\mathcal{L}_{ga}$ is not polynomial in the size of P. We do not write the UNA and PIA axioms explicitly in the examples below.

For standard AEL and ground disjunctive logic programs, the correspondence between the stable models of P and the stable expansions $\tau^{\vee}_{HP}(P)$ and $\tau^{\vee}_{EH}(P)$, respectively, is due to Przymusinski [1991a] and Marek and Truszczynski [1993].

PROPOSITION 4.10. *A Herbrand interpretation M of a ground disjunctive logic program P is a stable model of P iff there is a consistent stable expansion T of $\tau^{\vee -}_{HP}(P)$ (resp., $\tau^{\vee -}_{EH}(P)$) in standard AEL such that $M = T_{oga}$.*

We generalize this result to the case of FO-AEL and non-ground programs, and additionally for $\tau^{\vee}_{EB}$, similar to the case of normal programs.

LEMMA 4.11. *Let P be a logic program, let $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$, let T be a stable$^{\mathrm{X}}$ expansion of $\tau^{\vee}_{\chi}(P)$, and let α be an objective ground atom. Then, $\tau^{\vee}_{\chi}(P) \models^{\mathrm{X}}_{T_{oga}} \alpha$ iff $\tau^{\vee}_{\chi}(P) \models_{T_{oga}} \alpha$ under the standard names assumption. Moreover, $\tau^{\vee -}_{EH}(P) \models^{\mathrm{A}}_{T_{oga}} \alpha$ iff $\tau^{\vee -}_{EH}(P) \models_{T_{oga}} \alpha$.*

PROOF. ($\Rightarrow$) Trivial (cf. the "$\Rightarrow$" direction in Lemma 4.5).
($\Leftarrow$) The argument is a straightforward adaptation of the argument in the "$\Leftarrow$" direction in the proof of Lemma 4.5: simply replace the consequent $h_1 \wedge \mathsf{L} h_1$ with the disjunction $(h_1 \wedge \mathsf{L} h_1) \vee \cdots \vee (h_l \wedge \mathsf{L} h_l)$. Furthermore, it is also easy to see that, as w and w' agree on ground atomic formulas, if the PIA axioms are satisfied in $\langle w, T_{oga} \rangle$, then they are satisfied in $\langle w', T_{oga} \rangle$. □

LEMMA 4.12. *Let P be a logic program and let $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. There exists a stable$^{\mathrm{X}}$ expansion T of $\tau^{\vee}_{\chi}(P)$ iff there exists a stable$^{\mathrm{X}}$ expansion T' of $\tau^{\vee}_{\chi}(gr(P))$ with $T'_{oga} = T_{oga}$. The same result holds for $\tau^{\vee -}_{EH}$ and stable$^{\mathrm{A}}$ expansions.*

PROOF. The proof is obtained from the proof of Lemma 4.6 by replacing occurrences of τ_{χ} with $\tau^{\vee}_{\chi}$ and using Lemma 4.11 in place of Lemma 4.5. □

THEOREM 4.13. *Let P be a logic program and let $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. A Herbrand interpretation M is a stable model of P iff there exists a consistent stable$^{\mathrm{X}}$ expansion T of $\tau^{\vee}_{\chi}(P)$ such that $M = T_{oga}$. The same result holds for $\tau^{\vee -}_{EH}$ and stable$^{\mathrm{A}}$ expansions.*

Proof. The reduction of embeddability of non-ground programs in FO-AEL to ground logic programs in standard AEL follows from Lemma 4.12.

Embeddability of $gr(P)$ using $\tau^{\vee}_{HP}$ and $\tau^{\vee}_{EH}$ follows from Proposition 4.10. Embeddability of $gr(P)$ using $\tau^{\vee}_{EB}$ then follows from the embeddability of $\tau^{\vee}_{HP}$, combined with the PIA axioms ($\alpha \supset \mathsf{L}\alpha$): consider a formula $\bigwedge(b_i \wedge \mathsf{L}b_i) \wedge \bigwedge(\neg\mathsf{L}c_j) \supset \bigvee h_k$ in $\tau^{\vee}_{EB}(gr(P))$ and some b_i. If some model $\langle w, \Gamma\rangle$ of $\tau^{\vee}_{EB}(gr(P))$ satisfies b_i, then $\mathsf{L}b_i$ must also be satisfied in $\langle w, \Gamma\rangle$ (by the PIA axioms). Therefore, the stable expansions of $\tau^{\vee}_{HP}(P)$ and $\tau^{\vee}_{EB}(P)$ must be the same. □

A notable difference between the embeddings $\tau^{\vee}_{HP}$ and $\tau^{\vee}_{EB}$, on the one hand, and $\tau^{\vee}_{EH}$, on the other, is the presence, respectively absence, of the PIA axioms, as illustrated in the following example.

Example 4.14. Consider $P = \{p \,|\, q \leftarrow\ \}$. Then, $\tau^{\vee}_{HP}(P) = \{p \vee q\} \cup \mathrm{PIA}_{\Sigma_P}$ has the stable expansions $T^{HP}_1 = \{p, \neg q, \mathsf{L}p, \neg\mathsf{L}q, \dots\}$ and $T^{HP}_2 = \{q, \neg p, \mathsf{L}q, \neg\mathsf{L}p, \dots\}$, while $\tau^{\vee}_{EH}(P) = \{\ (p \wedge \mathsf{L}p) \vee (q \wedge \mathsf{L}q)\ \}$ has the stable expansions $T^{EH}_1 = \{p, \mathsf{L}p, \neg\mathsf{L}q, \dots\}$ and $T^{EH}_2 = \{q, \mathsf{L}p, \neg\mathsf{L}p, \dots\}$; the latter include neither $\neg q$ nor $\neg p$.

Note that the embedding τ_{HP} cannot be naively extended to logic programs with strong ("classical") negation $\sim$ [Gelfond and Lifschitz 1991], even for the propositional case. Take, for example, the logic program $P = \{p \leftarrow \sim p\}$; it has one stable model, namely $M = \emptyset$. The naive extension of τ_{HP} treats strong negation as negation in classical logic and the embedding of P yields $\{\neg p \supset p\}$, which has one stable expansion, which includes p. It was shown by Marek and Truszczynski [1993] that, for the propositional case, the embeddings τ_{EB} and τ_{EH} *can* be naively extended to the case of logic programs with strong negation: consider a rule of the form (1) such that h_i, b_j, c_k are either atoms or strongly negated atoms, and an extension of the embeddings τ_{EB}, τ_{EH} such that $\sim$ is translated to classical negation $\neg$; then, Proposition 4.3 straightforwardly extends to these extended versions of τ_{EB} and τ_{EH} [Marek and Truszczynski 1993]. These results can be straightforwardly extended to the non-ground case. Embedding of logic programs with strong negation using τ_{HP} can be done by rewriting P to a logic program P' without strong negation and subsequently embedding P'; see [Gelfond and Lifschitz 1991] for such a rewriting.

5. RELATIONSHIPS BETWEEN THE EMBEDDINGS

In this section, we explore correspondences between the embeddings presented in the previous section. We compare the stable expansions of the individual embeddings and, at the level of inference, we compare the sets of autoepistemic consequences. To this end we introduce the following notation:

Definition 5.1. Let $\Phi_1, \Phi_2 \subseteq \mathcal{L}_{\mathsf{L}}$ be FO-AEL theories and $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$. We write $\Phi_1 \equiv^{\mathrm{X}} \Phi_2$ if Φ_1 and Φ_2 have the same stable$^{\mathrm{X}}$ expansions. For $\gamma \in \{o, og, oga\}$ we write $\Phi_1 \equiv^{\mathrm{X}}_{\gamma} \Phi_2$ if, for each stable$^{\mathrm{X}}$ expansion T of Φ_1, there exists some stable$^{\mathrm{X}}$ expansion T' of Φ_2 such that $T_\gamma = T'_\gamma$, and vice versa.

Note the implication chain $\Phi_1 \equiv^{\mathrm{X}} \Phi_2 \Rightarrow \Phi_1 \equiv^{\mathrm{X}}_{o} \Phi_2 \Rightarrow \Phi_1 \equiv^{\mathrm{X}}_{og} \Phi_2 \Rightarrow \Phi_1 \equiv^{\mathrm{X}}_{oga} \Phi_2$.

Definition 5.2. A formula ϕ is an *autoepistemic*$^{\mathrm{X}}$ *consequence* of a theory $\Phi \subseteq \mathcal{L}_{\mathsf{L}}$, $\mathrm{X} \in \{\mathrm{E}, \mathrm{A}\}$, if ϕ belongs to every stable$^{\mathrm{X}}$ expansion of Φ. $Cn^{\mathrm{X}}(\Phi)$ denotes the set of all autoepistemic$^{\mathrm{X}}$ consequences of Φ.

The properties stated in this section holds regardless of whether $\mathrm{X} = \mathrm{E}$ or $\mathrm{X} = \mathrm{A}$ is considered. Therefore, we omit the superscript X from $\models^{\mathrm{X}}$, $\equiv^{\mathrm{X}}_{\gamma}$, Cn^{X}, stable$^{\mathrm{X}}$, etc. Furthermore, we write $Cn_{\gamma}(\Phi)$ for $Cn(\Phi)_{\gamma}$ $(= Cn(\Phi) \cap \mathcal{L}_{\mathsf{L}\gamma})$.

In our analysis, we consider different classes of logic programs. With the symbols $\mathcal{LP}$, $s\mathcal{LP}$, and $g\mathcal{LP}$ we denote the classes of arbitrary, safe, and ground disjunctive logic programs, respectively. Observe the following inclusions between the classes:

$$g\mathcal{LP} \subseteq s\mathcal{LP} \subseteq \mathcal{LP}.$$

We use the letter n to denote the restriction of the respective classes to the case of normal programs: $n\mathcal{LP}$, $sn\mathcal{LP}$, and $gn\mathcal{LP}$.

We start in Section 5.1 with an investigation of the correspondences between stable expansions and subsequently consider in Section 5.2 correspondences between sets of consequences. Note that while $\Phi_1 \equiv_{\gamma} \Phi_2$ implies $Cn_{\gamma}(\Phi_1) = Cn_{\gamma}(\Phi_2)$, the converse is not true in general. Thus, for applications based on consequence rather than stable expansions, more flexibility between the choice of equivalent embeddings can be expected as one-to-one correspondence between stable expansions is not required. In order not to interrupt the flow of reading, the proofs of most of the results in this section can be found in the appendix.

5.1 Relationships between Stable Expansions of Embeddings

From Theorems 4.7 and 4.13 we immediately obtain the following result concerning correspondence of stable expansions, which is our main result in this regard.

Theorem 5.3. *For every $P \in \mathcal{LP}$, $\tau(P) \equiv_{oga} \tau'(P)$ for all $\tau, \tau' \in \{\tau^{\vee}_{HP}, \tau^{\vee}_{EB}, \tau^{\vee}_{EH}\}$, and if $P \in n\mathcal{LP}$, then $\tau(P) \equiv_{oga} \tau'(P)$ for all $\tau, \tau' \in \{\tau_{HP}, \tau_{EB}, \tau_{EH}, \tau^{\vee}_{HP}, \tau^{\vee}_{EB}, \tau^{\vee}_{EH}\}$.*

Thus, all embeddings may be used interchangeably when concerned with ground atoms. This does not hold for the case of arbitrary objective ground formulas.

Example 5.4. Consider the logic program $P = \{a \leftarrow b\}$. Then $\tau_{HP}(P) = \{b \supset a\}$ has a single stable expansion, which contains $b \supset a$; also $\tau_{EB}(P) = \{b \wedge \mathsf{L}b \supset a\}$ has a single stable expansion, but it does not contain $b \supset a$. Note that while the latter contains $\mathsf{L}b \supset b$, it does not contain $b \supset \mathsf{L}b$ (which would enable obtaining $b \supset a$).

The situation changes for the embeddings $\tau^{\vee}_{HP}$ and $\tau^{\vee}_{EB}$ due to the PIA axioms.

Proposition 5.5. *For every $P \in n\mathcal{LP}$, $\tau_{EB}(P) \equiv_{og} \tau_{EH}(P)$, and for every $P \in \mathcal{LP}$, $\tau^{\vee}_{HP}(P) \equiv_{og} \tau^{\vee}_{EB}(P)$.*

For non-ground formulas we obtain the following result.

Proposition 5.6. *For every $P \in sn\mathcal{LP}$, $\tau_{EB}(P) \equiv \tau_{EH}(P)$.*

For arbitrary normal programs, the embeddings τ_{EB} and τ_{EH} differ.

Example 5.7. Consider $P = \{p(a); p(x); q(x) \leftarrow p(x)\}$. Then, the embedding $\tau_{EH}(P) = \{p(a) \wedge \mathsf{L}p(a), \forall x.p(x) \wedge \mathsf{L}p(x), \forall x.p(x) \wedge \mathsf{L}p(x) \supset q(x) \wedge \mathsf{L}q(x)\}$ has one stable expansion, which contains $\forall x.q(x)$, while $\tau_{EB}(P) = \{p(a), \forall x.p(x), \forall x.p(x) \wedge \mathsf{L}p(x) \supset q(x)\}$ has one stable expansion, which does not contain $\forall x.q(x)$, because

$\forall x.\mathsf{L}p(x)$ is not necessarily true when $\forall x.p(x)$ is true; in other words, the converse Barcan formula ($\mathsf{L}\forall x.\phi(x) \supset \forall x.\mathsf{L}\phi(x)$) is not universally valid, which is a property of FO-AEL under both the any- and all-name semantics [Konolige 1991].

Note that the result also does not extend to the embeddings τ_{HP} and $\tau_{HP}^{\vee}$.

Example 5.8. Consider $P = \{q(x) \leftarrow p(x)\}$. Then, $\tau_{HP}(P) = \{\forall x.p(x) \supset q(x)\}$ has one stable expansion, which contains $\forall x.p(x) \supset q(x)$, while $\tau_{EB}(P) = \{\forall x.p(x) \wedge \mathsf{L}p(x) \supset q(x)\}$ has one stable expansion which does not contain $\forall x.p(x) \supset q(x)$. This difference is caused by the fact that $\mathsf{L}p(x)$ will be false in case an unnamed individual is assigned to x. Similar observations hold for $\tau_{HP}^{\vee}$; the PIA axioms do not help, since they are only concerned with ground atoms and thus do not apply to unnamed individuals.

PROPOSITION 5.9. *If* $P \in g\mathcal{LP}$, *then* $\tau_{HP}^{\vee}(P) \equiv \tau_{EB}^{\vee}(P)$.

Note that this result does not extend to the embedding $\tau_{EH}^{\vee}$; it does not include the PIA axioms, and thus the argument used in the proof of Proposition 5.9 does not apply.

5.2 Relationships between Consequences of Embeddings

In order to investigate the relationships between the embeddings with respect to autoepistemic consequences, we first compare the embeddings with respect to their autoepistemic models. Recall that an autoepistemic interpretation $\langle w, T\rangle$ consists of a first-order interpretation w and a belief set $T \subseteq \mathcal{L}_{\mathsf{L}}$.

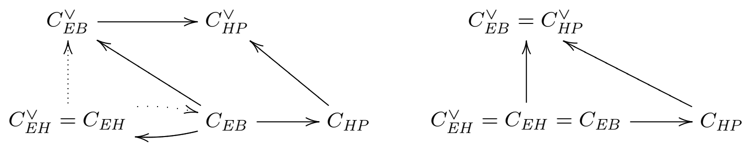
PROPOSITION 5.10. *For every* $P \in n\mathcal{LP}$ *and every interpretation* $\langle w, T\rangle$, $w \models_T \tau_{EH}(P)$ *implies* $w \models_T \tau_{EB}(P)$ *and* $w \models_T \tau_{HP}(P)$ *implies* $w \models_T \tau_{EB}(P)$.

PROPOSITION 5.11. *For every* $P \in \mathcal{LP}$ *and every interpretation* $\langle w, T\rangle$, $w \models_T \tau_{HP}^{\vee}(P)$ *implies* $w \models_T \tau_{EB}^{\vee}(P)$. *Furthermore, if* P *is safe, then* $w \models_T \tau_{EB}^{\vee}(P)$ *implies* $w \models_T \tau_{EH}^{\vee}(P)$.

We now consider the relative behavior of the embeddings with respect to autoepistemic consequences. In order to present our results in a compact and accessible way, we show a small (yet sufficient) number of relationships between the sets of consequences in a graph (Figure 1). Every particular relationship between embeddings can be easily derived from paths in this graph.

Specifically, in Figure 1(a), $C_{\chi}^{(\vee)}$ is short for $Cn_o(\tau_{\chi}^{(\vee)}(P))$, the straight arrow $\longrightarrow$ represents set inclusion ($\subseteq$), and the dotted arrow $\dashrightarrow$ represents set inclusion in case P is safe. Since $\longrightarrow$ implies $\dashrightarrow$, dotted arrows are only shown if straight arrows are absent. Similarly, in Figure 1(b), $C_{\chi}^{(\vee)}$ is short for $Cn_{og}(\tau_{\chi}^{(\vee)}(P))$, and $\longrightarrow$ represents set inclusion.

The main results visible from Figure 1 are that with respect to all objective consequences, τ_{EB} is the weakest embedding (yielding a smallest set of conclusions) while $\tau_{HP}^{\vee}$ and $\tau_{EH}^{\vee}$ are strongest; if the embedded program is safe, then $\tau_{EH}^{\vee}$ is the strongest embedding and τ_{EB} the weakest, collapsing with τ_{EH} and $\tau_{EH}^{\vee}$. With respect to ground objective consequences, $\tau_{HP}^{\vee}$ collapses with $\tau_{EB}^{\vee}$ and is the strongest embedding, while τ_{EB} is the weakest and again collapses with τ_{EH} and $\tau_{EH}^{\vee}$; safety of the program does not change the picture. Note that with respect to objective ground atomic consequences, all embeddings collapse (cf. Theorem 5.3).



(a) (b)

Fig. 1. Inclusion between sets of (a) all objective and (b) all objective ground consequences.

The following lemma states the correctness of Figure 1(a).

Lemma 5.12. *If* $C_{\chi}^{(\vee)} \longrightarrow C_{\gamma}^{(\vee)}$ *(resp.,* $C_{\chi}^{(\vee)} \dashrightarrow C_{\gamma}^{(\vee)}$*) in the graph of Figure 1(a), then* $Cn_o(\tau_{\chi}^{(\vee)}(P)) \subseteq Cn_o(\tau_{\gamma}^{(\vee)}(P))$ *for every* $P \in n\mathcal{LP}$ *(resp., for every* $P \in sn\mathcal{LP}$*). Furthermore, if* $C_{\chi}^{\vee} \longrightarrow C_{\gamma}^{\vee}$ *(resp.,* $C_{\chi}^{\vee} \dashrightarrow C_{\gamma}^{\vee}$*), then* $Cn_o(\tau_{\chi}^{\vee}(P)) \subseteq Cn_o(\tau_{\gamma}^{\vee}(P))$ *for every* $P \in \mathcal{LP}$ *(resp., for every* $P \in s\mathcal{LP}$*).*

Note that by transitivity of $\subseteq$, paths in the graph yield further relations; e.g., $Cn_o(\tau_{EB}(P)) \subseteq Cn_o(\tau_{HP}^{\vee}(P))$ since C_{EB} reaches $C_{HP}^{\vee}$ via a path with straight edges.

We now show that the graph exactly characterizes the containment relationships via paths. To this end, we first note some negative relationships between embeddings.

Lemma 5.13. *The following inclusion relations do not hold:* $Cn_o(\tau_{EH}^{\vee}(P)) \subseteq Cn_o(\tau_{HP}^{\vee}(P))$*, for every* $P \in \mathcal{LP}$*;* $Cn_o(\tau_{EB}^{\vee}(P)) \subseteq Cn_o(\tau_{HP}(P))$*, for every* $P \in sn\mathcal{LP}$*; and* $Cn_o(\tau_{HP}(P)) \subseteq Cn_o(\tau_{EB}^{\vee}(P))$*, for every* $P \in sn\mathcal{LP}$*.*

From these negative relationships, combined with the positive ones above, we can infer further negative relationships. For example, from $C_{EH}^{\vee} \not\subseteq C_{EB}^{\vee}$ and $C_{EH}^{\vee} \subseteq C_{EH}$, we infer $C_{EH} \not\subseteq C_{EB}^{\vee}$. Exploiting this, we show the following result.

Theorem 5.14. *For* $P \in n\mathcal{LP}$ *(resp.,* $P \in sn\mathcal{LP}$*),* $Cn_o(\tau_{\chi}^{(\vee)}(P)) \subseteq Cn_o(\tau_{\gamma}^{(\vee)}(P))$ *iff* $C_{\gamma}^{(\vee)}$ *is reachable from* $C_{\chi}^{(\vee)}$ *in the graph in Figure 1(a) on a path with* $\longrightarrow$ *arcs (resp., with arbitrary arcs).*

Likewise, for $P \in \mathcal{LP}$ *(resp.,* $P \in s\mathcal{LP}$*), we have that* $Cn_o(\tau_{\chi}^{\vee}(P)) \subseteq Cn_o(\tau_{\gamma}^{\vee}(P))$ *iff* $C_{\gamma}^{\vee}$ *is reachable from* $C_{\chi}^{\vee}$ *on a path with* $\longrightarrow$ *(resp., with arbitrary arcs).*

Proof. By Lemmas 5.12 and 5.13, the respective containment relationships are correct. Clearly, by reflexivity and transitivity of set inclusion, paths in the graph of Figure 1(a) are sound with respect to positive containments. Their completeness, for both arbitrary P and safe P, is established using the following basic properties of non-inclusion: (i) $A \not\subseteq B$ and $C \subseteq B$ implies $A \not\subseteq C$ and (ii) $A \not\subseteq B$ and $A \subseteq C$ implies $C \not\subseteq B$.[5] Exhaustive application to the (non-)containments in Lemmas 5.12 and 5.13 (e.g., using a simple logic program) yields one of $C_{\chi}^{(\vee)} \subseteq C_{\gamma}^{(\vee)}$ and $C_{\chi}^{(\vee)} \not\subseteq C_{\gamma}^{(\vee)}$ for each pair $C_{\chi}^{(\vee)}$, $C_{\gamma}^{(\vee)}$. □

---

[5] Note that non-inclusion for normal programs implies non-inclusion for disjunctive programs, since every normal program is *a fortiori* a disjunctive program.

Accordingly, $C_{EB} \subseteq C^{\vee}_{EH}$ and $C_{EB} \subseteq C^{\vee}_{HP}$ are the only nontrivial inclusions for arbitrary programs besides those in Figure 1(a); for safe programs, there are more. We note that the figure is minimal, in the sense that if any of the arcs is removed (or turned from solid into dashed), the theorem no longer holds.

The containment relationships in Figure 1(b) are easily obtained from the results already established.

Theorem 5.15. *For every* $P \in n\mathcal{LP}$, $Cn_{og}(\tau^{(\vee)}_{\chi}(P)) \subseteq Cn_{og}(\tau^{(\vee)}_{\gamma}(P))$ *holds iff* $C^{(\vee)}_{\gamma}$ *is reachable from* $C^{(\vee)}_{\chi}$ *in the graph in Figure 1(b). Furthermore, for every* $P \in \mathcal{LP}$, $Cn_{og}(\tau^{\vee}_{\chi}(P)) \subseteq Cn_{og}(\tau^{\vee}_{\gamma}(P))$ *holds iff* $C^{\vee}_{\gamma}$ *is reachable from* $C^{\vee}_{\chi}$.

Proof. By Theorem 5.14 and Proposition 5.5, $C^{\vee}_{EH}$, C_{EH}, and C_{EB} collapse, and by Proposition 5.5, $C^{\vee}_{EB}$ and $C^{\vee}_{HP}$ collapse. That the other relationships in Figure 1(a) remain unchanged follows from Example 5.4, the proof of Lemma 5.13, and reasoning about subsets as in the proof of Theorem 5.14. □

6. COMBINATIONS WITH FIRST-ORDER THEORIES

In this section, we explore correspondences between the logic program embeddings from Section 4 in combinations with FO theories. To this end, we consider a basic combination of logic programs P and FO theories Φ defined as

$$\iota^{(\vee)}_{\chi}(\Phi, P) = \Phi \ \cup \ \tau^{(\vee)}_{\chi}(P) \subseteq \mathcal{L}_{\mathsf{L}}$$

where $\Sigma_{\mathcal{L}_{\mathsf{L}}}$ is the union of the signatures Σ_{Φ} and Σ_P. More involved combinations (e.g., which augment P and Φ with further rules and axioms, respectively) might be recast to such basic combinations.

In the preceding sections we have considered both the any- and all-name semantics, both in the definition of the embeddings and in our analysis of the differences between the embeddings of logic programs. It turned out that the embeddings are faithful for both semantics (cf. Theorems 4.7 and 4.13), implying correspondence with respect to objective ground atoms between the two semantics for all embeddings $\tau^{(\vee)}_{\chi}$, and the relationships between the embeddings stated in the previous section hold for both semantics. However, in combinations with FO theories, the two semantics diverge since names from the first-order part may not be provably identical to or different from other names. The following example illustrates differences between the semantics in the face of positive and negative occurrences of the modal operator.

Example 6.1. Consider the logic program P:

$$\begin{aligned} q(a)&, \\ r \leftarrow p(x)&, not\ q(x), \\ s(x) \leftarrow p(x)&, \end{aligned}$$

and the FO theory $\Phi = \{p(b)\}$. We note here that the signature of P contains only one function symbol, the constant constant a. Consequently, $\mathrm{UNA}_{\Sigma_P} = \emptyset$.

$\iota_{EB}(\Phi, P)$ has one stable$^{\mathrm{E}}$ expansion T^{E} and one stable$^{\mathrm{A}}$ expansion T^{A}. T^{E} contains $q(a)$, but not $q(b)$; both contain $p(b)$, but not $p(a)$. Consider an interpretation $w = \langle U, \cdot^I \rangle$ such that $a^w = b^w = k$, $k \in p^I, k \in q^I, k \in s^I$, and $w \not\models r$, and a variable assignment B such that $x^B = k$. Then, $\beta = \{x/a\}$ is an associated name substitution and $q(x)\beta \in T^{\mathrm{E}}$, and so $(w, B) \models^{\mathrm{E}}_{T^{\mathrm{E}}} \mathsf{L}q(x)$. Another associated name substitution is $\beta = \{x/b\}$, and so $(w, B) \models^{\mathrm{E}}_{T^{\mathrm{E}}} \mathsf{L}p(x)$. So, $w \models^{\mathrm{E}}_{T^{\mathrm{E}}} \iota_{EB}(\Phi, P)$ and thus $r \notin T^{\mathrm{E}}$. One can straightforwardly argue that $\iota_{EB}(\Phi, P) \models^{\mathrm{E}}_{T} s(b)$, and therefore $s(b) \in T^{\mathrm{E}}$.

Consider an interpretation w' that is like w, except that $k \notin s^I$ and $w' \models r$. Since T^{A} does not contain $p(a)$, $(w', B) \not\models^{\mathrm{A}}_{T^{\mathrm{A}}} \mathsf{L}p(x)$. T^{A} does not contain $q(b)$, but $(w, B) \models^{\mathrm{A}}_{T^{\mathrm{A}}} \mathsf{L}q(x)$ holds only if $q(x)\beta \in T^{\mathrm{A}}$ for *every* β associated with B, including $\beta = \{x/b\}$, and so $(w', B) \models^{\mathrm{A}}_{T^{\mathrm{A}}} \neg\mathsf{L}q(x)$ and $w' \models^{\mathrm{A}}_{T^{\mathrm{A}}} \iota_{EB}(\Phi, P)$. Consequently, $s(b) \notin T^{\mathrm{A}}$. It is straightforward to verify that $r \in T^{\mathrm{A}}$.

In order to avoid a proliferation of results, following Konolige [1991], we concentrate in this section on the any-name semantics. In the following section we discuss our results in the light of the standard names assumption, for which the any- and all-name semantics coincide.

In our analysis we consider the same syntactic classes of programs as in the previous section and we consider the following classes of objective theories:

> arbitrary ($\mathcal{F}o\mathcal{L} = 2^{\mathcal{L}}$), universal ($\mathcal{U}ni$), Horn ($\mathcal{H}orn$), generalized Horn ($g\mathcal{H}orn$), propositional ($\mathcal{P}rop$), and empty ($\{\emptyset\}$, i.e., in semantic terms, tautological) FO theories.

Generalized Horn formulas have the form $(\forall) b_1 \wedge \cdots \wedge b_n \supset \exists\vec{y}.h$ where all b_i and h are atomic and variables $\vec{y}$ occur only in h, which may be absent. Note that similar formulas $(\forall) b_1 \wedge \cdots \wedge b_n \supset \exists\vec{y}.(h_1 \wedge \cdots \wedge h_m)$ where all b_i and h_j are atomic, can be easily cast to $g\mathcal{H}orn$ formulas by replacing the conjunction $h_1 \wedge \cdots \wedge h_m$ with $p(\vec{x})$, where $\vec{x}$ are its free variables and p is a new predicate symbol, and including formulas $\forall\vec{x}.p(\vec{x}) \supset h_j$ for each h_j; the resulting theory is equivalent with respect to the original signature. The class $g\mathcal{H}orn$ captures RDF Schema [de Bruijn and Heymans 2007], DLs such as Horn-$\mathcal{SHIQ}$ [Hustadt et al. 2005] and the OWL 2 profiles EL, QL, and RL, as well as Tuple Generating Dependencies (TGDs), which play an important role in relational databases; Calì et al. [2009] survey decidable fragments of TGDs that amount to decidable fragments of $g\mathcal{H}orn$ (see also Sections 7 and 9.1).

The order diagram is as follows (arrows stand for set inclusion):

$$\begin{tikzcd}
 & \mathcal{P}rop \arrow[r] & \mathcal{U}ni \arrow[dr] & \\
\{\emptyset\} \arrow[ur] \arrow[dr] & & & \mathcal{F}o\mathcal{L} \\
 & \mathcal{H}orn \arrow[r] \arrow[uur] & g\mathcal{H}orn \arrow[ur] &
\end{tikzcd}$$

For all pairs of classes of logic programs and FO theories, we determine the relationships between stable expansions of different combinations $\iota^{(\vee)}_{\chi}(\Phi, P)$ and $\iota^{(\vee)}_{\gamma}(\Phi, P)$ at different levels of granularity. As in Section 5.1, we concentrate here on correspondences of stable expansions $\equiv^{\mathrm{E}}_{\mathrm{x}}$; they imply that relative to the class x of formulas, the embeddings $\tau^{(\vee)}_{\chi}(P)$ and $\tau^{(\vee)}_{\gamma}(P)$ are interchangeable in combinations.

In Section 6.1, we state our main result on the relationships between stable expansions of combinations and make several observations. In Section 6.2, we

Table I. Correspondences between stable expansions of combinations (on programs in the joint definition range); $\iota_x^{(\vee)}$ is short for $\iota_x^{(\vee)}(\Phi, P)$.

$\Phi \setminus P$	$\mathcal{LP}$	$s\mathcal{LP}$	$g\mathcal{LP}$
$\mathcal{FoL}$	$\iota_{EH} \equiv^{\mathrm{E}} \iota^{\vee}_{EH}$		$\iota_{EB} \equiv^{\mathrm{E}} \iota_{EH}$ $\iota^{\vee}_{HP} \equiv^{\mathrm{E}} \iota^{\vee}_{EB}$
$\mathcal{U}ni$			
$g\mathcal{H}orn$			$\iota_{HP} \equiv^{\mathrm{E}}_{oga} \iota_{EH}$ $\iota^{\vee}_{HP} \equiv^{\mathrm{E}}_{oga} \iota^{\vee}_{EH}$
$\mathcal{H}orn$			
$\mathcal{P}rop$	$\iota^{\vee}_{HP} \equiv^{\mathrm{E}}_{og} \iota^{\vee}_{EB}$ $\iota_{EB} \equiv^{\mathrm{E}}_{og} \iota_{EH}$	$\iota_{EB} \equiv^{\mathrm{E}} \iota_{EH}$	
$\{\emptyset\}$	$\iota_{HP} \equiv^{\mathrm{E}}_{oga} \iota_{EB}$ $\iota_{HP} \equiv^{\mathrm{E}}_{oga} \iota_{EH}$ $\iota^{\vee}_{HP} \equiv^{\mathrm{E}}_{oga} \iota^{\vee}_{EH}$		

establish the partial results necessary for deriving our main result. The proofs of the partial results can be found in the appendix.

6.1 Relationships between Stable Expansions of Combinations

Our results are summarized in Table I, which gives a complete picture of the correspondences, where each entry represents a *most general correspondence*, i.e., neither the correspondence $\equiv^{\mathrm{E}}_{\mathrm{x}}$ nor the logic program or FO theory class may be relaxed. This is formally stated in the main theorem of this section (Theorem 6.2). In brief, our central results are that several of the embeddings become interchangeable when considering positive normal programs combined with $g\mathcal{H}orn$ or $\mathcal{H}orn$ theories (cf. Proposition 6.3) as well as the correspondences for combinations with ground logic programs, even allowed to contain negation (cf. the rightmost column of Table I).

We call $\Phi_1 \equiv^{\mathrm{E}}_{\mathrm{x}} \Phi_2$ a *trivial inference* from a set Q of equivalences if it is derivable from Q by the fact that $\Phi_1 \equiv^{\mathrm{E}} \Phi_2$ implies $\Phi_1 \equiv^{\mathrm{E}}_{og} \Phi_2$ and $\Phi_1 \equiv^{\mathrm{E}}_{og} \Phi_2$ implies $\Phi_1 \equiv^{\mathrm{E}}_{oga} \Phi_2$, as well as by reflexivity, transitivity, and symmetry of $\equiv^{\mathrm{E}}_{\mathrm{y}}$, $\mathrm{y} \in \{\epsilon, og, oga\}$.

Theorem 6.2. *Let $\mathcal{X}$ be a class of FO theories, let $\mathcal{Y}$ be a class of programs, and let* $\mathrm{x} \in \{\epsilon, og, oga\}$*. Then $\iota_\chi(\Phi, P) \equiv^{\mathrm{E}}_{\mathrm{x}} \iota_\gamma(\Phi, P)$ holds for all $\Phi \in \mathcal{X}$ and all $P \in \mathcal{Y}$ iff $\iota_\chi(\Phi, P) \equiv^{\mathrm{E}}_{\mathrm{x}} \iota_\gamma(\Phi, P)$ follows for cell $(\mathcal{X}, \mathcal{Y})$ in Table I by trivial inferences, where $\chi, \gamma \in \{{}^{\vee}_{HP}, {}^{\vee}_{EB}, {}^{\vee}_{EH}\}$ if $P \in \mathcal{LP}$ and $\chi, \gamma \in \{{}_{HP}, {}_{EB}, {}_{EH}, {}^{\vee}_{HP}, {}^{\vee}_{EB}, {}^{\vee}_{EH}\}$ if $P \in n\mathcal{LP}$.*

We will establish the results of Table I and provide some intuitive explanations about partial results in the next subsection.

Note that removing any statement from Table I or modifying any correspondence type invalidates the theorem. We do not explicitly consider correspondence of stable expansions with respect to objective formulas, i.e., $\equiv^{\mathrm{E}}_{o}$. Clearly, $\Phi_1 \equiv^{\mathrm{E}} \Phi_2$ implies $\Phi_1 \equiv^{\mathrm{E}}_{o} \Phi_2$; in addition, all the counterexamples to $\Phi_1 \equiv^{\mathrm{E}} \Phi_2$ presented in the following subsection also apply to $\Phi_1 \equiv^{\mathrm{E}}_{o} \Phi_2$. Hence, $\equiv^{\mathrm{E}}_{o}$ coincides with $\equiv^{\mathrm{E}}$.

The use of negation is essential for establishing non-correspondence in some cases, as we have the following result for positive programs and Horn theories.

Proposition 6.3. *For every $(\Phi, P) \in g\mathcal{H}orn \times sn\mathcal{LP}$ such that P is positive, $\iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$, and for every $(\Phi, P) \in \mathcal{H}orn \times n\mathcal{LP}$ such that P is positive, $\iota_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.*

This result does not extend to the disjunctive embeddings, because there are no PIA axioms for the atoms involving names not in P, and it does not extend to more general formulas (cf., e.g., Example 4.2).

We make the following further observations.

—The various combinations behave differently in the general case. Only two of them, ι_{EH} and $\iota^{\vee}_{EH}$, are always equivalent (they coincide on normal programs).

—For combinations with arbitrary FO theories, further correspondences are only present for ground logic programs in some cases. Narrowing to any of the classes that allow predicates of arity > 0 ($\mathcal{H}orn$, $g\mathcal{H}orn$, $\mathcal{U}ni$) does not change the picture.

—For arbitrary logic programs, only in case of propositional theories do some combinations behave equivalently. Requiring safety leads only for propositional theories and in one case (ι_{EB} and ι_{EH}) to a stronger correspondence.

—$\mathcal{U}ni$ and $\mathcal{H}orn$ show no most general correspondences, which means that with respect to more general or more restrictive classes, their change in syntax does not affect equivalence.

—In contrast, the important class $g\mathcal{H}orn$ has maximal correspondences for ground programs. Thus, for combinations with FO theories from the classes $\mathcal{H}orn$, $g\mathcal{H}orn$, and $\mathcal{U}ni$, we have equivalent behavior only for ground programs in some cases (apart from ι_{EH} and $\iota^{\vee}_{EH}$).

—The effect of program grounding (which is a customary technique applied by nonmonotonic rule engines) in combinations is thus beneficial for the embeddings τ_{EB} and τ_{EH}, making them interchangeable; this is similar for $\tau^{\vee}_{HP}$ and $\tau^{\vee}_{EB}$ and for τ_{HP}, τ_{EB}, τ_{EH}, resp. $\tau^{\vee}_{HP}$, $\tau^{\vee}_{EB}$, $\tau^{\vee}_{EH}$, in combinations with a $g\mathcal{H}orn$ first-order part with respect to ground atomic formulas. In fact, the interchangeability of τ_{EB} and τ_{EH} lifts by Proposition 6.3 to the non-ground level for $g\mathcal{H}orn$ FO parts and safe positive programs (which are at the core of rule bases in practice), with respect to ground atomic formulas. Importantly, the proof of Proposition 6.3 shows that the combination is invariant under program grounding, so this technique can be readily applied. We get a similar picture for τ_{HP}, τ_{EB}, τ_{EH} in combinations of $\mathcal{H}orn$ FO parts with positive programs. (For applications, see Section 7.)

We illustrate the use of the result in Theorem 6.2 with an example. Note that if the stable expansions of two embeddings or combinations correspond with respect to a certain class formulas, then the embeddings, resp. combinations, also agree on autoepistemic consequences for these classes.

Example 6.4. Consider $P = \{q(a);\ p(x);\ r(x) \leftarrow \mathit{not}\ s(x),\ p(x)\}$ from Example 4.4, which is neither safe nor ground. Hence, to determine correspondence between embeddings, we use the first column of Table I. As P is normal, all equations in the column are applicable. We have that, e.g., $\tau_{EB}(P) \equiv^{\mathrm{E}}_{og} \tau_{EH}(P)$, $\tau^{\vee}_{HP}(P) \equiv^{\mathrm{E}}_{og} \tau^{\vee}_{EB}(P)$, and $\tau_{HP}(P) \equiv^{\mathrm{E}}_{oga} \tau_{EB}(P)$. Let Φ be a propositional theory; then we also have $\iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{og} \iota_{EH}(\Phi, P)$ and $\iota^{\vee}_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{og} \iota^{\vee}_{EB}(\Phi, P)$, but not $\iota_{HP}(P) \equiv^{\mathrm{E}}_{oga} \iota_{EB}(P)$. Furthermore, we can conclude that $\iota_{EB}(P)$ and $\iota_{EH}(P)$, and also $\iota^{\vee}_{HP}(P)$ and $\iota^{\vee}_{EB}(P)$, agree on objective ground autoepistemic consequences.

6.2 Derivation of the Results

We start with the positive results. Trivially, ι_{EH} and $\iota_{EH}^{\vee}$ coincide for arbitrary FO theories, and the the equivalence results for empty Φ in Table I carry over from the respective results on embeddings in Section 5.

We show that for ground programs, the τ_{EB} and τ_{EH} embeddings are interchangeable in any combination with an FO theory.

PROPOSITION 6.5. *For every* $(\Phi, P) \in \mathcal{FoL} \times gn\mathcal{LP}$, $\iota_{EB}(\Phi, P) \equiv^{\mathrm{E}} \iota_{EH}(\Phi, P)$.

Intuitively, this holds because only named individuals matter in rules, and hence the modal atoms $\mathsf{L}h$ in embedded rule heads do not matter. However, this does not generalize from ground to safe programs, as the evaluation of literals $\neg\mathsf{L}p(x)$ in the rule bodies does not amount to grounding (see Proposition 6.10(4)).

Also the $\tau_{HP}^{\vee}$ and $\tau_{EH}^{\vee}$ embeddings are interchangeable in combinations with arbitrary FO theories if the logic program is ground.

PROPOSITION 6.6. *For every* $(\Phi, P) \in \mathcal{FoL} \times g\mathcal{LP}$, $\iota_{HP}^{\vee}(\Phi, P) \equiv^{\mathrm{E}} \iota_{EB}^{\vee}(\Phi, P)$.

The reason is that we can eliminate all modal atoms $\mathsf{L}b$ from rule bodies with the PIA axioms in $\iota_{EB}^{\vee}(\Phi, P)$ and obtain $\iota_{HP}^{\vee}(\Phi, P)$. Such elimination is not possible in the non-ground case, since the PIAs only apply to atoms from Σ_P.

Moving now to fragments of $\mathcal{FoL}$, i.e., down the rows in Table I, we first have:

PROPOSITION 6.7. *For* $(\Phi, P) \in g\mathcal{H}orn \times g\mathcal{LP}$, $\iota_{HP}^{\vee}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}^{\vee}(\Phi, P)$, *and if* $P \in gn\mathcal{LP}$, *then* $\iota_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.

Intuitively, in the first case, we can add modal atoms $\mathsf{L}b$ in the bodies and $\mathsf{L}h$ in the heads of $\tau_{HP}(P)$, by the PIA axioms, thereby obtaining $\iota_{EH}^{\vee}(\Phi, P)$. To go from τ_{EH} to τ_{HP} is possible if Φ is not disjunctive with respect to atoms h. This is the case for a Horn Φ, and similarly for a generalized Horn Φ, as we can apply skolemization. In the second case, there are no PIA axioms, but we can similarly apply skolemization and obtain a disjunction-free theory that is Horn modulo modal atoms. Skolemization does not work for non-ground programs in this case, as previously unnamed individuals are named by Skolem terms.

We note that, combined with previous results, we can infer from Proposition 6.7 that Theorem 5.3 generalizes from embeddings to combinations with generalized Horn theories for the case of ground logic programs.

For propositional theories, we obtain a result symmetric to Proposition 6.6 for arbitrary logic programs.

PROPOSITION 6.8. *For every* $(\Phi, P) \in \mathcal{P}rop \times \mathcal{LP}$, $\iota_{HP}^{\vee}(\Phi, P) \equiv^{\mathrm{E}}_{og} \iota_{EB}^{\vee}(\Phi, P)$.

Intuitively, this holds because a propositional Φ cannot interfere with names of individuals—it has no names. Therefore, as in the case of the embeddings $\tau_{HP}^{\vee}$ and $\tau_{EB}^{\vee}$, we can eliminate all modal atoms $\mathsf{L}b$ from rule bodies in $\iota_{EB}^{\vee}(\Phi, P)$ to obtain $\iota_{HP}^{\vee}(\Phi, P)$. For similar reasons, also correspondence results for the embeddings τ_{EB} and τ_{EH} extend to combinations with propositional theories.

PROPOSITION 6.9. *For every* $(\Phi, P) \in \mathcal{P}rop \times n\mathcal{LP}$, $\iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{og} \iota_{EH}(\Phi, P)$, *and if* $P \in sn\mathcal{LP}$, *then* $\iota_{EB}(\Phi, P) \equiv^{\mathrm{E}} \iota_{EH}(\Phi, P)$.

It turns out that the results in the preceding propositions cannot be extended to more general classes of programs or theories, or larger subsets of stable expansions.

PROPOSITION 6.10. *There are pairs* (Φ, P) *in*

(1) $\mathcal{P}rop{\times}gn\mathcal{LP}$ *such that* $\iota_\chi(\Phi, P) \not\equiv^{\mathrm{E}}_{oga} \iota_\gamma(\Phi, P)$, *for* $(\chi, \gamma) \in \{(_{HP},_{EB}), (_{HP},\overset{\vee}{_{HP}}), (\overset{\vee}{_{HP}},\overset{\vee}{_{EH}})\}$;

(2) $\{\emptyset\}{\times}gn\mathcal{LP}$ *such that* $\iota_\chi(\Phi, P) \not\equiv^{\mathrm{E}}_{og} \iota_\gamma(\Phi, P)$, *for* $(\chi, \gamma) \in \{(_{HP},_{EB}), (_{HP},\overset{\vee}{_{HP}}), (\overset{\vee}{_{HP}},\overset{\vee}{_{EH}})\}$;

(3) $\{\emptyset\}{\times}sn\mathcal{LP}$ *such that* $\iota^{\vee}_{HP}(\Phi, P) \not\equiv^{\mathrm{E}} \iota^{\vee}_{EB}(\Phi, P)$;

(4) $\mathcal{H}orn{\times}sn\mathcal{LP}$ *such that* $\iota_\chi(\Phi, P) \not\equiv^{\mathrm{E}}_{oga} \iota_\gamma(\Phi, P)$, *for* $(\chi, \gamma) \in \{(_{HP}, _{EB}), (_{HP}, _{EH}), (_{EB}, _{EH}), (_{HP}, \overset{\vee}{_{HP}}), (_{HP}, \overset{\vee}{_{EB}}), (_{EB}, \overset{\vee}{_{EB}}), (_{EB}, \overset{\vee}{_{HP}}), (\overset{\vee}{_{HP}}, \overset{\vee}{_{EB}}), (_{EH}, \overset{\vee}{_{EB}}), (_{EH}, \overset{\vee}{_{HP}})\}$; *and*

(5) $\{\emptyset\}{\times}n\mathcal{LP}$ *such that* $\iota_{EB}(\Phi, P) \not\equiv^{\mathrm{E}} \iota_{EH}(\Phi, P)$.

PROOF OF THEOREM 6.2. Correctness of the table (i.e., the '$\Leftarrow$' direction of the theorem) follows from the fact that $\tau_{EH}(P)$ and $\tau^{\vee}_{EH}(P)$ are identical for normal programs P (thus $\iota_{EH}(\Phi, P) = \iota^{\vee}_{EH}(\Phi, P)$), Theorem 5.3, Propositions 6.5–6.9, the inclusion relations between the classes of programs and FO theories, and the properties of the $\equiv^{\mathrm{E}}_{\mathrm{x}}$ relation. Completeness (i.e., the '$\Rightarrow$' direction) is shown analogously, by exploiting the counterexamples in Proposition 6.10. One can verify with little yet tedious effort that the table is complete and that no entries can be relaxed (e.g., using a simple logic program). □

7. APPLICATION TO THE SEMANTIC WEB

The original motivation for our work was the interest of combinations of rules and ontologies in the Semantic Web. Below we illustrate how our results may be applied in this context. Briefly, some uses are:

—to capture the semantics of proposed combinations in a uniform language;
—to derive properties of such and other combinations;
—to design semantics for combinations, such that the ontology and rule parts are faithfully captured, and controlling the effect of aspects like grounding and working with open vs. closed domains.

We note that, apart from an obvious relationship to the open vs. closed domain issue, grounding rules is important from a practical perspective, since many rule engines in use today employ grounding. In fact, powerful rule engines like smodels[6], DLV[7], and clasp[8], along with many others that offer stable and/or well-founded semantics are essentially based on evaluation of ground programs. Thus, aspects such as the invariance of a combination with respect to grounding the rules prior to evaluation (formally captured in Definition 7.1 below) are also important from a practical perspective.

We concentrate on two prominent Semantic Web languages, namely (i) RDF and its extension RDF Schema (RDFS) [RDF Concepts 2004], and (ii) OWL DL

---

[6] `http://www.tcs.hut.fi/Software/smodels/`.
[7] `http://www.dbai.tuwien.ac.at/proj/dlv/`.
[8] `http://www.cs.uni-potsdam.de/clasp/`.

(Version 2) [OWL 2 2009]. For reasons of clarity we restrict ourselves to normal programs.

In the remainder of this section, we consider the combinations ι_χ, which are defined as $\iota_\chi(\Phi, P) = \Phi \cup \tau_\chi(P)$, with $\chi \in \{HP, EB, EH\}$, where Φ is an FO theory (the ontology) and P is a normal logic program.

7.1 Grounding Invariance and Closed Domains

In order to state our results for RDF and OWL concerning grounding and open vs. closed domains, we first formally define grounding invariance and closed domain semantics. In the following, $\mathcal{X}$ is a class of FO theories and $\mathcal{Y}$ is a class of normal logic programs.

Definition 7.1. A combination ι_χ is *invariant under grounding* (or *fulfills grounding invariance*) for $\mathcal{X}, \mathcal{Y}$, if $\iota_\chi(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_\chi(\Phi, gr(P))$, for every $(\Phi, P) \in \mathcal{X} \times \mathcal{Y}$.

When speaking about open and closed domain semantics in the context of combinations of rules and ontologies, we are interested in the effective domain of quantification of the variables in the rules. In the open domain semantics, variables quantify over arbitrary domains, while in the closed domain semantics variables quantify over a fixed domain, e.g., the set of ground terms obtained from the constants and function symbols appearing in the rules or ontologies.

Recall that, given a normal program P and a rule $r \in P$, the embedding $\tau_\chi(r)$ is a formula of the form $(\forall)\ b_r \supset h_r$.

Definition 7.2. A combination ι_χ is *closed-domain* for $\mathcal{X}, \mathcal{Y}$ if, for every $(\Phi, P) \in \mathcal{X} \times \mathcal{Y}$ and every stable expansion T of $\iota_\chi(\Phi, P)$, the following property holds: For every interpretation w such that $w \models^{\mathrm{E}}_T \iota_\chi(\Phi, P)$ and variable assignment B, whenever $w, B \models^{\mathrm{E}}_T b_r$ for some rule $r \in P$, then B assigns every variable x in r to a named individual, i.e., $x^B = t^I$, for some name t. Otherwise, ι_χ is *open-domain* for $\mathcal{X}, \mathcal{Y}$.

Essentially, a combination is closed domain if rules can only be applied (i.e., the body is satisfied in a model and variable assignment) if all variables are assigned to named individuals.

We have that combinations involving only ground logic programs are trivially closed-domain.

Proposition 7.3. *Combinations defined as $\iota_\chi(\Phi, gr(P))$ are closed-domain for every $\mathcal{X}, \mathcal{Y}$.*

Since combinations that are invariant under grounding are equivalent (with respect to ground atoms) to the combination obtained by grounding the program, grounding invariance essentially implies closed-domain. The following observations follow straightforwardly from the definitions of the respective combinations and the properties of autoepistemic logic.

Proposition 7.4. *The combinations ι_{EB} and ι_{EH} are closed-domain for FO theories and safe logic programs. The combination ι_{HP} is open-domain already for empty theories and positive safe normal logic programs.*

Of particular interest to combinations of rules and ontologies on the Semantic Web are *DL-safe* programs [Motik et al. 2005], which yield grounding invariance (for positive programs), thereby effectively imposing a closed-domain semantics. An atom $p(\vec{t})$ is a *rule atom* if p appears only in P. We call the program P *DL-safe*, if P is safe and every variable in every rule r of P appears in a rule atom in $B^+(r)$.

The next propositions follow straightforwardly from the proof of Proposition 6.3.

PROPOSITION 7.5. *Let Φ be a $g\mathcal{H}orn$ theory and P a DL-safe positive normal program. Then, $\iota_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.*

PROPOSITION 7.6. *The following combinations fulfill grounding invariance:*

(*1*) *ι_{EB} and ι_{EH} for $g\mathcal{H}orn$ theories and safe positive normal programs;*
(*2*) *ι_{HP}, ι_{EB}, and ι_{EH} for $\mathcal{H}orn$ theories and safe positive normal programs; and*
(*3*) *ι_{HP}, ι_{EB}, and ι_{EH} for $g\mathcal{H}orn$ theories and DL-safe positive normal programs.*

Even under DL-safety, a generalization of this result from positive to normal logic programs fails, and the combinations ι_{HP}, ι_{EB}, and ι_{EH} behave differently—this can be shown by replacing in the proof of Proposition 6.10(4) the theory Φ with $\{p(b),\, p(a) \supset q\}$; the program P is then DL-safe. From Theorem 6.2, we can then conclude the following:

PROPOSITION 7.7. *The combinations ι_{HP}, ι_{EB}, and ι_{EH} are not invariant under grounding for $\mathcal{H}orn$ theories and DL-safe normal logic programs.*

Observe that this result, combined with Proposition 7.4, shows that closed-domain does not imply grounding invariance. In contrast, Proposition 7.3 shows that the converse effectively holds, as long as one is interesting only in ground atoms.

A weaker notion of safety, namely *weak DL-safety* [Rosati 2006] (see also Section 9.2.2) also plays an important role in the Semantic Web context, because of the possibility to write conjunctive queries over DL ontologies. A program P is *weakly DL-safe*, if it is safe and for every rule r and every variable x in r, x either appears only in non-rule atoms in $B^+(r)$ or x appears in a rule atom in $B^+(r)$. As weak DL-safety is stronger than ordinary safety, clearly Proposition 7.6(1) and Proposition 7.6(2) extend to weakly DL-safe programs. However, Proposition 7.6(3) does not, and the same happens also to Proposition 7.5.

PROPOSITION 7.8. *There is a pair $(\Phi, P) \in g\mathcal{H}orn \times sn\mathcal{LP}$ such that P is weakly DL-safe and positive, $\iota_{HP}(\Phi, P) \not\equiv^{\mathrm{E}}_{oga} \iota_{EB}(\Phi, P)$, and $\iota_{HP}(\Phi, P) \not\equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.*

PROOF. Consider $\Phi = \{\exists x.p(x)\}$ and $P = \{q \leftarrow p(x)\}$: ι_{HP} (Φ, P) allows to conclude q, whereas ι_{EB} (Φ, P) and ι_{EH} (Φ, P) do not. □

7.2 RDF, RDF Schema, and Rules

Recall that RDF is the basic data description language of the Semantic Web, in which atomic statements have the form *triple*(*subject*, *predicate*, *object*). RDFS has further axioms about the meaning of certain triples; for example, that the facts *triple*(a, rdfs:subClassOf, b) and *triple*(b, rdfs:subClassOf, c) imply *triple*(a, rdfs:subClassOf, c).

As shown by de Bruijn and Heymans [2007], (finite) RDF graphs S are essentially $g\mathcal{H}orn$ theories of the form $\Phi = \{\exists \vec{x}. \bigwedge S\} \cup \Psi$, where the free variables in S are among $\vec{x}$ and Ψ is a set of function-free Horn logic formulas, which capture the RDFS semantics [RDF Semantics 2004].

Combinations of RDF graphs with rules—e.g., the RIF RDF and OWL compatibility recommendation [RIF RDF-OWL 2009] and Jena[9]—are common, because of the flexibility to manipulate data that rules offer. Note that in this context it is not possible to make a strict separation between ontology and rule predicates, as the *triple* predicate is "defined" by both the ontology and the rules.

Current combinations of RDFS with rules are typically limited to positive Horn rules—a notable exception being the work by Analyti et al. [2008]. For example, the RIF-RDFS [RIF RDF-OWL 2009] semantics essentially defines the combination of an RDF graph Φ and a set of positive normal rules P as the first-order logic theory $\iota_{HP}(\Phi, P) = \Phi \cup \tau_{HP}(P)$.[10]

This semantics can be straightforwardly extended to normal rules P by interpreting the FO-AEL theory $\Phi \cup \tau_{HP}(P)$ using the any- or all-name semantics. Such an extension keeps the spirit of the RIF-RDFS semantics by having an *open domain*, i.e., not only the constants, but also the existentially quantified variables in the RDF graphs matter (see also Proposition 7.4).

However, rules typically have a closed-domain semantics. One may thus argue that combinations should respect this semantics and enforce a closed domain in the interaction between the RDF statements and the rules; examples of such combinations are $\Phi \cup \tau_\chi(gr(P))$ and $\Phi \cup \tau_{EH}(P)$: the former enforces closing of the domain through grounding, while the latter forces closing through the use of the modal operator L in the rules (cf. Proposition 7.3 and Proposition 7.4). Note that, by Theorem 6.2, the embeddings τ_{HP}, τ_{EB}, and τ_{EH} may be used interchangeably in the combination $\Phi \cup \tau_\chi(gr(P))$ (as long as we are interested only in ground atomic consequences), as Φ is in $g\mathcal{H}orn$ and $gr(P)$ is in $gn\mathcal{LP}$. The following example illustrates the difference between combinations with open and with closed domain semantics, respectively.

Example 7.9. Consider the RDF graph

$$\Phi = \{\exists x.triple(x, \text{director}, \text{TheGodfather})\} \cup \Psi$$

encoding the fact that there is a director of the film "The Godfather". Consider also the program $P = \{hasDirector(x) \leftarrow triple(y, \text{director}, x)\}$ encoding that whenever someone directs a film, then this film has a director. We have

$$\begin{aligned}\tau_{HP}(P) =& \{\forall x, y.triple(y, \text{director}, x) \supset hasDirector(x)\} \text{ and}\\ \tau_{EH}(P) =& \{\forall x, y.triple(y, \text{director}, x) \wedge \mathsf{L}triple(y, \text{director}, x) \supset\\ & \quad hasDirector(x) \wedge \mathsf{L}hasDirector(x)\}.\end{aligned}$$

Clearly, $hasDirector(\text{TheGodFather})$ is a consequence of $\Phi \cup \tau_{HP}(P)$, but not of $\Phi \cup \tau_{EH}(P)$, as there is no constant c such that $triple(\text{c, director, TheGodFather})$ is included in the single stable expansion of $\Phi \cup \tau_{EH}(P)$.

Similarly, $hasDirector(\text{TheGodFather})$ is not a consequence of $\Phi \cup \tau_\chi(gr(P))$, since there is no constant representing the director.

---

[9] `http://jena.sourceforge.net/`.

[10] We avoid here to go into unnecessary and tedious detail concerning the RIF-RDFS semantics specification, which does not give further insight.

Proposition 7.6 shows that grounding P or not, prior to combination with Φ, does not matter if Φ is an RDF graph without blank nodes (as then Φ is in $\mathcal{H}orn$) and P is positive—in particular, $\Phi \cup \tau_\chi(gr(P)) \equiv^{\mathrm{E}}_{oga} \Phi \cup \tau_\chi(P)$. Similarly for τ_{EB} and τ_{EH}, if Φ is an arbitrary RDF graph, and P is safe and positive. If P is moreover DL-safe (see Section 7.1), this invariance under grounding for arbitrary RDF graphs and safe positive programs also extends to the τ_{HP} embedding, and thus τ_{EB}, τ_{EH}, and τ_{HP} are all interchangeable. We furthermore have that the open and closed domain semantics coincide in the cases mentioned in this paragraph.

If S is an RDF graph, we define $\iota_\chi(S, P) = \Phi \cup \tau_\chi(P)$, where $\Phi = \{\exists\vec{x}. \bigwedge S\} \cup \Psi$, as before. An RDF graph S is *ground* if it does not contain free variables. Such a graph is equivalent to the Horn theory $\Phi = \{\bigwedge S\} \cup \Psi$. From Propositions 7.5 and 7.6 we then obtain:

COROLLARY 7.10. *The combinations ι_{EB} and ι_{EH} fulfill grounding invariance for RDF graphs and safe positive normal programs, and ι_{HP} fulfills it for ground RDF graphs and safe positive normal programs, as well as for RDF graphs and DL-safe positive normal programs. Moreover, for RDF graphs Φ and DL-safe positive normal programs P, it holds that $\iota_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.*

A notable further consequence of Proposition 7.6 is the following observation concerning the standard RIF-RDFS semantics [RIF RDF-OWL 2009].

COROLLARY 7.11. *The RIF-RDFS combination semantics fulfills grounding invariance for DL-safe positive normal programs.*

For a possible use case scenario, suppose the ontology Φ is a ground RDF graph. Now, suppose the user wants to add a set of DL-safe positive normal rules and follow the standard RIF-RDFS combination semantics [RIF RDF-OWL 2009]. If the user is interested only in ground atomic consequences, Corollary 7.11 tells us that this semantics is invariant under grounding and thus essentially closed-domain, by Proposition 7.3. Even when extending the graph with variables, the combination remains invariant under grounding and thus closed-domain. However, grounding invariance may be lost when extending the program with negation, by Proposition 7.7.

7.3 OWL DL and Rules

The Web Ontology Language OWL DL is based on Description Logics (DLs); Version 1 [OWL Semantics 2004] is based on the DL $\mathcal{SHOIN}$ and Version 2 [OWL 2 2009] on the DL $\mathcal{SROIQ}$. Both DLs can be viewed as subsets of first-order logic [Sattler et al. 2003]. An influential proposal for combining OWL DL ontologies with positive normal rules is the Semantic Web Rules Language (SWRL) [Horrocks et al. 2004], which gives a standard first-order semantics to their union.

A SWRL theory Φ consists of a set of DL axioms and a set of Horn-like formulas. We obtain the following correspondence with ι_{HP} combinations.

PROPOSITION 7.12. *Let Φ be a SWRL theory. Then, there is an FO theory Φ' and a safe positive normal logic program P such that $\Phi \models \alpha$ iff α is a consequence of $\iota_{HP}(\Phi', P)$, for every objective ground atom α.*

An approach similar to SWRL was adopted by the RIF working group for positive normal RIF rules [RIF RDF-OWL 2009]. If Φ is the FOL-equivalent of an OWL

DL ontology and P is a set of positive normal rules, the semantics of RIF-OWL DL combinations is given by the FO theory $\iota_{HP}(\Phi, P) = \Phi \cup \tau_{HP}(P)$. Proposition 7.12 implies that this is equivalent to SWRL.

Regarding open vs. closed domains, similar considerations as in Section 7.2 apply to combinations of OWL DL with rules: $\Phi \cup \tau_{HP}(P)$ yields an open domain, while $\Phi \cup \tau_{\chi}(gr(P))$ and $\Phi \cup \tau_{EH}(P)$ yield a closed domain on the rule side (see Example 7.9). However, interchangeability and invariance of the embeddings τ_{HP}, τ_{EB}, and τ_{EH} under grounding may not be guaranteed, as Φ need not be in $g\mathcal{H}orn$.

Example 7.13. Consider $\Phi = \{A(a), \forall x.A(x) \supset B(x) \vee C(x)\}$, which captures a simple OWL DL ontology, and $P = \{q \leftarrow B(x); q \leftarrow C(x)\}$. Now,

$$\tau_{HP}(gr(P)) = \{B(a) \supset q, C(a) \supset q\} \text{ and}$$

$$\tau_{EH}(gr(P)) = \{B(a) \wedge \mathsf{L}B(a) \supset q \wedge \mathsf{L}q, C(a) \wedge \mathsf{L}C(a) \supset q \wedge \mathsf{L}q\}.$$

We have that q is a consequence of $\Phi \cup \tau_{HP}(gr(P))$, but not of $\Phi \cup \tau_{EH}(gr(P))$, since neither $B(a)$ nor $C(a)$ is included in the single stable expansion of $\Phi \cup \tau_{EH}(gr(P))$.

There are important fragments of OWL DL that are essentially included in $g\mathcal{H}orn$, such as the OWL 2 profiles EL, QL, and RL [OWL 2 Profiles 2009], and the fragment corresponding to Horn-$\mathcal{SHIQ}$ [Hustadt et al. 2005]. As was the case with RDF, when considering the combination $\Phi \cup \tau_{\chi}(gr(P))$ (see Propositions 7.5 and 7.6), the embeddings τ_{HP}, τ_{EB}, and τ_{EH} may be used interchangeably, and, for safe positive programs, τ_{EB} and τ_{EH} are invariant under grounding. Furthermore, OWL 2 RL is essentially in $\mathcal{H}orn$. Therefore, when considering combinations of OWL 2 RL ontologies with positive normal programs, ι_{HP} (Φ, P), ι_{EB} (Φ, P), and ι_{EH} (Φ, P) may be used interchangeably, by Proposition 6.3, and the combinations are invariant under grounding of the rules, by Proposition 7.6. Moreover, they are closed-domain.

As shown by Motik et al. [2005], reasoning with OWL DL plus DL-safe rules (i.e., SWRL having DL-safe rules) is decidable. From Propositions 7.5 and 7.6, we obtain the following corollary. Here, OWL $g\mathcal{H}orn$ theories are theories of OWL 2 EL, OWL 2 QL, OWL 2 RL, or Horn-$\mathcal{SHIQ}$.

COROLLARY 7.14. *The combinations ι_{EB} and ι_{EH} fulfill grounding invariance for OWL $g\mathcal{H}orn$ theories and safe positive normal programs, and ι_{HP} fulfills it for OWL 2 RL theories and safe positive normal programs, as well as for OWL $g\mathcal{H}orn$ theories and DL-safe positive normal programs. Moreover, if Φ is an OWL $g\mathcal{H}orn$ theory and P a DL-safe positive normal program, then $\iota_{HP}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EB}(\Phi, P) \equiv^{\mathrm{E}}_{oga} \iota_{EH}(\Phi, P)$.*

Consider a scenario in which the ontology Φ is in both OWL 2 RL and OWL 2 EL and one wants to add positive rules that are safe (but not DL-safe), using the standard RIF-OWL combination semantics [RIF RDF-OWL 2009]. Corollary 7.14 tells us that one may employ any of the considered combinations ι_{χ} and may ground the rules, as long as one is only interested in atomic formulas. However, if we were to extend the ontology Φ towards full OWL 2 EL by introducing existentially quantified variables (also called *someValuesFrom* restrictions in OWL) and we want to stay faithful to the RIF-OWL combination semantics, we may no longer use the ι_{EB} or ι_{EH} combinations, as illustrated by Example 7.9. In addition, we may not ground the rules prior to reasoning. Therefore, if such a future extension towards OWL 2 EL is likely, one should choose the ι_{HP} embedding rather than ι_{EB} or ι_{EH}, and should not rely on grounding for reasoning.

8. DISCUSSION

In this section, we discuss implications of our results. We first discuss consequences on the relationships between the embeddings, and make a number of observations about those relationships. We then discuss how the results in this paper can be used in the context of combining classical theories (ontologies) with logic programs (rules)—specifically, how the embeddings studied in this paper can be used as building blocks for such combinations. Finally, we discuss our choice of FO-AEL as the underlying formalism, and compare the semantics for quantification (quantifying-in) with other approaches to quantifying-in in autoepistemic logic [Levesque 1990; Levesque and Lakemeyer 2000; Kaminski and Rey 2002].

8.1 Relationships between the Embeddings

Using the results obtained in Sections 5 and 6, we can make a number of observations about the embeddings:

(1) The differences between the embeddings by themselves do not depend on the use of negation in the program. Generally speaking, the differences originate from the positive use of the modal operator in the antecedent and the consequent, and the use of the PIA axioms. However, in combinations with FO theories, the interaction between names in the theories for which there are no UNA axioms and negation in the rules gives rise to different behavior of the embeddings (see Proposition 6.10(4)).

(2) The stable expansions of embeddings with and without the PIAs generally tend to differ. However, we can note that the former are generally stronger in terms of autoepistemic consequences (cf. Figure 1 and Example 4.14).

(3) The embeddings τ_{HP} and $\tau_{HP}^{\vee}$ are generally the strongest in terms of consequences (see Figure 1), when comparing to other embeddings without and with PIAs, respectively. They allow to derive the contrapositive of rules (cf. Example 4.2) and the bodies of rules are applicable to unnamed individuals, whereas the antecedents of the axioms in the other embeddings are only applicable to named individuals, because of the positive modal atoms in the bodies.

(4) For unsafe programs, the embeddings τ_{EH} and $\tau_{EH}^{\vee}$ are generally not comparable with the others; embeddings of unsafe rules may result in axioms of the form $\forall x.\mathsf{L}p(x)$ (cf. Example 4.4), which result in all individuals being named.

(5) If names in a theory Φ that lack UNA axioms and rules in a program P do not interact (e.g., Φ is propositional or P is ground), then τ_{EB} and τ_{EH} are in most cases interchangeable.

Special care needs to be taken if one selects an embedding that includes the PIA axioms (i.e., $\tau_{HP}^{\vee}$ and $\tau_{EB}^{\vee}$). These axioms of the form $\alpha \supset \mathsf{L}\alpha$ ensure that α or $\neg\alpha$ is included in every stable expansion, for every ground atom of Σ_P. Note that the PIA axioms have no effect when considering individuals that are not named by ground terms in Σ_P.

The UNA axioms in embeddings, which serve to make individuals different by default, may interact with the FO theory in a combination. For example, consider $P = \{p(a);\ p(b)\}$ and $\Phi = \{a \neq b \supset r,\ a \neq c \supset s\}$. Then, every stable expansion of $\iota_\chi(\Phi, P)$, for any embedding τ_χ we considered, contains r as $a \neq b$ is concluded by default, but not s (as c is unknown in P). To shortcut such (possibly undesired) inequality transfers from P to Φ, the unique names or even the standard names assumption may be adopted *a priori*. Recall that the results on the embeddings in Section 4 were obtained by stepping through the standard names assumption, and thus they also hold under the unique names or standard names assumption, as shown by de Bruijn et al. [2008]. On the one hand, this should extend to the positive results about correspondences in Sections 5 and 6, whose proofs rely on named interpretations and no equalities between individuals are enforced. On the other hand, some counterexamples for correspondences fail, including those for the first item in Lemma 5.13 and Proposition 6.10(4), and thus further correspondences may hold. An in-depth study of the effect of unique names and standard names assumptions on the correspondences and differences between the embeddings is an interesting subject for further work.

8.2 Different Embeddings and Combinations

Recall the general setting for combining a first-order theory Φ and a logic program P in a unifying formalism (FO-AEL) that we sketched in the introduction. The combination operator ι takes as arguments the theory Φ and the program P, and returns an FO-AEL theory $\iota(\Phi, P)$. The operator provides two embedding functions: σ and τ map first-order theories, respectively logic programs, to FO-AEL theories. We also mentioned that in the simplest case the combination is the union of the two individual embeddings: $\iota(\Phi, P) = \sigma(\Phi) \cup \tau(P)$.

In Section 4 we investigated several candidates for the embedding function for logic programs, τ. All these embedding functions are faithful, in the sense that the stable models of the program P correspond to the sets of objective ground atomic formulas in the stable expansions of the embedding $\tau(P)$. In Section 5 we investigated the relationships between the stable expansions of these embeddings when considering more general formulas. It turned out that there are already significant differences between the expansions when considering non-ground or non-atomic formulas.

Now, in Section 6, we investigated the relationships between the expansions when considering combinations of the embeddings with first-order theories. We have found that, under certain circumstances—namely, when the first-order theory and program are of particular shapes and we are interested in a particular kind of formulas (e.g., ground formulas)—certain embeddings can be used interchangeably (cf. Table I). For example, if the program is normal and ground ($P \in gn\mathcal{LP}$), the theory is generalized Horn ($\Phi \in g\mathcal{H}orn$), and we are interested in objective formulas, we can use the embeddings τ_{EB} and τ_{EH} interchangeably: $\iota_{EB} \equiv \iota_{EH}$ for $P \in gn\mathcal{LP}$ and $\Phi \in \mathcal{F}o\mathcal{L}$, according to Table I, as $\mathcal{H}orn \subseteq \mathcal{F}o\mathcal{L}$ and the set of objective formulas is a subset of the set of formulas.

Our results are not limited to combinations of the form $\iota(\Phi, P) = \Phi \cup \tau_\chi^{(\vee)}(P)$, where $\tau_\chi^{(\vee)}$ is one of the embeddings investigated in this paper. One could imagine adding axioms to Φ or rules to P to achieve the desired interoperation between the two components, or even changing the axioms or rules (e.g., by grounding), obtaining a first-order theory Φ' and program P'. In this more general setting, the combination is defined as

$$\iota(\Phi, P) = \Phi' \cup \tau_{\chi}^{(\vee)}(P'),$$

where Φ' and P' are obtained from Φ and P by adding and/or replacing axioms and rules. The results of Section 6 can be applied, provided that Φ' and P' are in the respective classes of theories and programs, independent of the shapes of Φ and P (see also Section 7).

As discussed in Section 4, embeddings that include the UNA axioms are not modular in general, but only signature-modular. This can be remedied by instead using the single axiom

$$(\forall)\ \mathsf{L}x = x \wedge \mathsf{L}y = y \wedge \neg \mathsf{L}x = y \supset x \neq y$$

which has the same effect for embeddings. However, using this axiom would entail default uniqueness on all names in a combination, not only those from the signature of the program (if desired, such default uniqueness can be easily accomplished by just mentioning respective terms in the logic program). As a consequence, also the combinations behave differently.

8.3 Quantifying-in in First-Order Autoepistemic Logic

We consider here FO-AEL, with the semantics for quantifying-in as defined by Konolige [1991], as an underlying formalism for combinations of first-order theories and logic programs. However, further semantics for quantifying-in have been proposed in the literature.

Levesque [1990] defined the logic of only knowing (see also the subsequent work by Levesque and Lakemeyer [2000]), which is essentially a superset of FO-AEL. Levesque's semantics for quantifying-in is slightly different from the one of Konolige [1991] that we used in this paper. He adopted a standard names assumption that amounts to a special case of the notion in Section 2.1; there is a countably infinite number of constant symbols in the language, but there are no (other) function symbols. Likewise, the variant of FO-AEL by Kaminski and Rey [2002] also employs a standard names assumption, although under a somewhat different guise: the domain of every interpretation is an extended Herbrand interpretation, i.e., it is a superset of the set of constant symbols in the theory; function symbols are not considered. The semantics of Konolige does not impose such restrictions, e.g., the domain may be infinite, while the number of constants is finite, and function symbols are allowed.

It is well known that reasoning in standard first-order logic can be reduced to reasoning in first-order logic with the standard names assumption, as long as there are sufficiently many constant symbols available [Fitting 1996].

Different from Levesque [1990], Kaminski and Rey [2002] did not consider equality in the language. However, equality in first-order logic with standard names behaves quite differently from equality in standard first-order logic. In the latter case, two constant symbols may be interpreted as the same element in the domain, whereas in the former case, all constant symbols are interpreted distinctly, e.g., $a = b$ cannot be satisfied if a and b are distinct constant symbols.[11] It is, however, possible to reduce reasoning in standard first-order logic with equality to reasoning in first-order logic with standard names using a special congruence predicate [Fitting 1996, Theorem 9.3.9]. Motik and Rosati [2007] use such a predicate in their variant of the logic MKNF [Lifschitz 1991; 1994], as do de Bruijn et al. [2008] in a variant of FO-AEL with standard names; see Section 9.2 for further discussion about this work.

9. RELATED WORK

We review here two areas of related work: extensions of logic programming and description logic semantics with open domains and nonmonotonicity, respectively, and approaches to combining rules and ontologies.

9.1 Extensions of LP and DL Semantics

We have studied the combination of logic programs and ontologies using embeddings in a unifying formalism (FO-AEL). One could imagine, in contrast, extensions of the semantics of logic programs or ontologies to incorporate (parts of) the other formalism. One such extension of logic programming semantics is that of open domains [Gelfond and Przymusinska 1993]. Such extended semantics can be used to accommodate incomplete knowledge, an important aspect of ontology languages.

Van Belleghem et al. [1997] define *open logic programs*, which are combinations of sets of rules and first-order logic formulas; the set of predicate symbols is partitioned into a set of *open* and a set of *closed* predicates. The semantics of the program is the first-order theory consisting of Clark's completion of the closed predicates and the first-order formulas in the open program. They then discuss how description logics can be embedded in such open logic programs and they discuss the correspondence between abduction in open programs and reasoning in description logics.

Heymans et al. [2006] describe an extension of the stable model semantics with open domains, called *open answer set programming* (*OASP*). They show how the expressive DL $\mathcal{SHIQ}$ can be embedded in this language and Heymans et al. [2008] show how OASP can be used for combinations of rules and ontologies, following the $\mathcal{DL}+log$ semantics [Rosati 2006] (see Section 9.2).

Recently, Calì et al. [2009] presented Datalog$^{\pm}$ as a language that, similarly as OASP, can be used to enhance ontologies with rules. In essence, Datalog$^{\pm}$ amounts to a skolemized form of $g\mathcal{H}orn$ in a relational setting, where for decidability rules must satisfy a guardedness condition. As reported by Calì et al., various DLs can be encoded into Datalog$^{\pm}$, and thus, like in OASP, combination of rules and ontologies can be achieved by adding rules to this encoding. Furthermore, Calì et al. present a semantics for Datalog$^{\pm}$ programs with stratified negation that generalizes the usual notion of stratified programs, which thus enables combinations with nonmonotonic rules. An embedding of (stratified) Datalog$^{\pm}$ into FO-AEL via the embedding τ_{HP} seems easily possible, such that its (operational) semantics can be reconstructed in logical terms, as well of the combination with the DLs described. Moreover,

---

[11] Levesque and Lakemeyer [2000] extend the logic of only knowing by allowing the use of constants and function symbols different from standard names; several ground terms may be associated with one *standard name*, and for any constant symbols a and b with this property, $a = b$ is satisfied.

the embedding can be used to give semantics to unstratified Datalog$^{\pm}$ programs via FO-AEL, and the results of this paper can be exploited to derive properties. Investigating this in detail remains for future work.

Several nonmonotonic extensions of description logics have been defined in the literature [Baader and Hollunder 1995; Donini et al. 1998; Donini et al. 2002; Bonatti et al. 2006]. These might be further extended to accommodate logic programs by well-known correspondences of the latter to nonmonotonic formalisms. In more detail, extensions of DL semantics with defaults and circumscription have been described by Baader and Hollunder [1995] and Bonatti et al. [2006], respectively. Extensions with nonmonotonic modal operators, inspired by the logic MKNF [Lifschitz 1991], have been described by Donini et al. [1998; 2002]. Both works mention a notion of procedural or default rules, which are rules involving description logic concepts. Donini et al. [1998] allow rules of the form $C \Rightarrow D$, where C and D are DL concepts (i.e., unary predicates); such rules are intuitively read "if an individual is proved to be an instance of C, then derive that it is also an instance of D". The default rules considered by Donini et al. [2002] are a generalization; they are of the form $C_0, \mathit{not}\ C_1, \ldots, \mathit{not}\ C_n \Rightarrow D$, $n \geq 0$, where all C_i and D are DL concepts. Intuitively, "if an individual is proved to be an instance of C_0 and is not proved to be an instance of C_1, …, or C_n, then derive that it is also an instance of D". The work of Donini et al. inspired some more advanced formalisms for combining rules and ontologies, which we consider next.

9.2 Combinations of Rules and Ontologies

Roughly speaking, we can distinguish between three kinds of combinations of rules and ontologies: (1) *uniform combinations* (e.g., CARIN [Levy and Rousset 1998] and SWRL [Horrocks et al. 2005]), (2) *hybrid combinations* (e.g., dl-programs [Eiter et al. 2008] and $\mathcal{DL}$+*log* [Rosati 2006]), and (3) *embedding combinations* (e.g., the MKNF combination by Motik and Rosati [2007] and a combination based on quantified equilibrium logic [de Bruijn et al. 2007]); for more discussion, see, e.g., the works of Eiter et al. [2008] and de Bruijn et al. [2006]. We also note the recent approach by de Bruijn et al. [2008] for embeddings of dl-programs, $\mathcal{DL}$+*log*, and MKNF into FO-AEL.

9.2.1 *Uniform Combinations.* With uniform combinations we mean combinations of ontologies that are essentially classical first-order theories and of Horn logic formulas that are essentially positive rules. The combined theory, which is the set-theoretic union of the formulas in the ontology and the Horn formulas, is interpreted under the standard first-order logic semantics.

In the CARIN approach [Levy and Rousset 1998], the ontologies are theories of the description logic $\mathcal{ALCNR}$ and the rules are Datalog rules, i.e., safe positive normal rules as defined in Section 2.2, with the further restriction that predicates which occur in the ontology may not be used in rule heads. Levy and Rousset show that reasoning with these combinations is undecidable in general, but becomes decidable when suitably restricting either the ontology or the rules. As discussed in Section 7, Motik et al. [2005] demonstrated decidability of SWRL—the combination of OWL DL with normal positive rules—restricted to DL-safe rules.

9.2.2 *Hybrid Combinations.* Hybrid approaches combine logic programs with nonmonotonic negation (usually, under the stable model semantics or the well-founded semantics) with a description logic knowledge base or, in more abstract terms, theories in first-order logic. The two most prominent such approaches are dl-programs [Eiter et al. 2008] and $\mathcal{DL}$+*log* [Rosati 2006]. The main difference between them is the way in which the interaction between the individual components (the logic program and the ontology) is managed. For both, we assume that the ontology component is a DL theory and the logic program is function-free and safe.

In dl-programs, the interoperation between the program and the ontology is achieved by *DL queries*, which are queries to the DL ontology, in the bodies of the rules; prior to evaluation, information from the program may be temporarily added to the ontology for a query. Eiter et al. [2008] show that query answering in dl-programs is decidable as long as reasoning in the individual components (ontology and logic program) is decidable. HEX-programs [Eiter et al. 2005] generalize dl-programs to more general external evaluations that are not limited to queries on DL ontologies.

$\mathcal{DL}$+*log* makes a distinction between ontology and rules predicates; rules predicates may not occur in the ontology, but the ontology predicates may occur in the rules. The combination is interpreted by a single first-order interpretation, but the part of the interpretation concerned with the rules predicates is subject to stability conditions corresponding to the usual definition of stable models. Thus, the interoperation is based on single models, resulting in a broad interface between the program and the ontology. Rosati [2005] shows that if the rules are DL-safe and satisfiability checking in the ontology component is decidable, then reasoning with the combination is decidable. Rosati [2006] shows that reasoning is decidable if the problem of containment of conjunctive queries in unions of conjunctive queries is decidable for the underlying DL, provided that the rules are *weakly DL-safe*; this notion dispenses DL-safety for variables that occur only in ontology predicates in rule bodies, which makes it possible to access unnamed individuals in rules. $\mathcal{AL}$-*log* [Donini et al. 1998] can be seen as a precursor of $\mathcal{DL}$+*log* that considers only positive programs and that allows (unary) ontology predicates only in rule bodies and effectively requires DL-safety. The differences between the underlying principles of dl-programs and $\mathcal{DL}$+*log* are discussed in more detail by de Bruijn et al. [2006].

Since we did not distinguish between rule and ontology predicates in our embeddings—indeed, in the introduction we claimed this is undesirable—there is no straightforward correspondence between any of the embeddings we considered and the mentioned hybrid approaches. The embeddings we considered in this paper can be used to construct combinations that have a tight integration between the components and that do not have a separation between ontology and rules predicates. In fact, the $\mathcal{DL}$+*log* approach can be reconstructed by an extension of simple combinations $\iota_\chi\ (\Phi, P) = \Phi \cup P$ with *classical interpretation axioms*, which, loosely speaking, fix the value of classical predicates for stable expansions; we refer to de Bruijn et al. [2008] for details.

9.2.3 *Embedding Combinations.* Motik and Rosati [2007] propose a combination of DL ontologies and nonmonotonic logic programs through an embedding into the bimodal nonmonotonic logic MKNF [Lifschitz 1991], which uses the modal operators K, which stands for “knowledge”, and not, which stands for “negation as failure”. The variant of MKNF used by Motik and Rosati employs a standard names assumption similar to the approach of Levesque [1990]: there is a one-to-one correspondence between the countably many constant symbols in the language and elements in the domains of interpretations (functions symbols are not considered). The equality symbol of first-order logic ($=$) is embedded using a special binary predicate symbol $\approx$ and the usual congruence axioms [Fitting 1996, Chapter 9] are added. Logic programs are embedded into MKNF using the transformation described by Lifschitz [1994]: a rule r of form (1) is embedded as the formula

$$\tau_{MKNF}(r) = \bigwedge_i \mathsf{K}b_i \wedge \bigwedge_j \mathsf{not}\, c_j \supset \bigvee_k \mathsf{K}h_k.$$

A classical theory Φ is embedded as a conjunction comprising all the formulas in the theory, preceded by the modal operator K: $\sigma_{MKNF}(\Phi) = \mathsf{K}(\bigwedge \Phi)$. Finally, the combination of the logic program P and the first-order theory Φ is simply $\iota_{MKNF} = \tau_{MKNF}(P) \cup \{\sigma_{MKNF}(\Phi)\}$.

Comparing τ_{MKNF} to the embeddings in Section 4, we can see that it is close in spirit to the embedding $\tau_{EH}^{\vee}$; both embeddings feature modal belief operators in front of positive atoms in both the body and the head of the rule. In fact, it turns out that, when using a variant of FO-AEL with standard names, there is a one-to-one correspondence between the stable expansions of $\tau_{EH}^{\vee-}(P)$ and the MKNF models of $\tau_{MKNF}(P)$ (recall that $\tau_{EH}^{\vee-}(P)$ is $\tau_{EH}^{\vee}(P)$ without the UNA axioms); however, this correspondence does not extend to combinations with FO theories, as shown by de Bruijn et al. [2008].

Besides the obvious differences between MKNF and autoepistemic logic—illustrated by the differences between the τ_{MKNF} and $\tau_{EH}^{\vee-}$ embedding functions—there is a difference in the semantics for quantifying-in between the variant of MKNF used by Motik and Rosati [2007] and Konolige’s any- and all-name semantics that we used in this paper. Since FO-AEL permits arbitrary interpretations, we needed to utilize UNA axioms. Motik and Rosati employ the standard names assumption and thus do not need such axioms.

As already pointed out, de Bruijn et al. [2007] used another nonmonotonic logic for combining ontologies and logic programs, namely quantified equilibrium logic (QEL) [Pearce and Valverde 2005]. While FO-AEL and MKNF are nonmonotonic modal logics, QEL is based on the nonclassical *logic of here-and-there*, which is an intermediate logic between classical and intuitionistic logic. Negation in QEL is nonmonotonic; however, by axiomatizing the law of the excluded middle (LEM) through $\forall \vec{x}(p(\vec{x}) \vee \neg p(\vec{x}))$, one can enforce that a predicate p is interpreted classically, and negation of this predicate becomes classical. Actually, de Bruijn et al. [2007] used a slightly generalized version of QEL that does not assume uniqueness of names and includes equality to show that the QEL theory obtained by adding such LEM axioms to the combination $\iota(\Phi, P) = \Phi \cup P$ of a FO theory Φ and a logic program P yields the $\mathcal{DL}$+*log* semantics.

10. CONCLUSION

We have defined various embeddings of non-ground programs into first-order autoepistemic logic (FO-AEL) that generalize respective embeddings of propositional logic programs into standard AEL, and we have investigated their semantic properties. We have shown that these embeddings are faithful, in the sense that the stable models (or answer sets) of a given non-ground logic program P are in one-to-one correspondence to the stable expansions of the embeddings $\tau_{\chi}^{(\vee)}(P)$ with respect to objective ground atomic formulas. Furthermore, we have analyzed the correspondences between the embeddings at more fine-grained levels, revealing their commonalities and differences.

Our results provide a basis and a stepping stone for the more complex endeavor to combine classical knowledge bases and non-ground logic programs in a uniform logical formalism (which is one of the targets of the Semantic Web architecture), namely the well-known and amply studied formalism of autoepistemic logic. Indeed, since the combination of positive RIF rules with RDF and OWL DL [RIF RDF-OWL 2009] corresponds to one of the combinations we studied, our results are directly applicable to such combinations.

In this direction, we have investigated correspondences between simple combinations of embeddings of logic programs with FO theories for various classes of logic programs and FO theories. The results of our investigation provide useful insights into the behavior of different embeddings for logic programs with respect to a context, given by a first-order theory, and allows some conclusions about the replaceability of one embedding by another without altering the behavior of the combination. Based on the results in the present paper, more elaborated combinations of logic programs with FO theories are investigated by de Bruijn et al. [2008], who show how well-known approaches to combining rules and ontologies in the Semantic Web context can be embedded into FO-AEL, like those of Eiter et al. [2008], Rosati [2006], and Motik and Rosati [2007]. Notably, the $\mathcal{DL}$+*log* approach can be embedded into FO-AEL by adding further axioms to the simple combination that we have considered here.

Several issues remain for future work. In the present paper, we focused on semantic aspects of embeddings of logic programs, but we did not address computational issues. Since the embeddings are easily computed, they may be exploited to establish decidable fragments of combinations of rules and ontologies, and to craft sound (but possibly incomplete) algorithms for specific reasoning tasks for such combinations. There are several promising starting points for devising algorithms for computing stable expansions and/or autoepistemic consequences in FO-AEL. Niemelä [1992] presents a general procedure for computing stable expansions in FO-AEL without quantifying-in. Levesque and Lakemeyer [2000] present a sound, but incomplete proof theory for the logic of only knowing, which extends FO-AEL with standard names. Finally, Rosati [1999] presents techniques for reasoning with first-order MKNF (with standard names) with a limited form of quantifying-in; the not operator in MKNF is equivalent to $\neg\mathsf{L}$ in autoepistemic logic [Rosati 1997].

Other issues are extensions of the language used for logic programs. Adding classical negation to the τ_{EB} and τ_{EH} is routine, and has been done by de Bruijn et al. [2008] for FO-AEL with standard names. Other interesting extensions include nesting [Lifschitz et al. 1999], where the closeness between nesting in logic programs and the logic MKNF suggests that an embedding is straightforward, and aggregates [Faber et al. 2004; Ferraris 2005; Pelov et al. 2007; Son and Pontelli 2007].

Furthermore, in the present work, we considered embeddings of logic programs interpreted under the stable model semantics, which adopts a two-valued semantics. It would be interesting to consider also embedding of logic programs under many-valued semantics, most importantly under the well-founded semantics [Gelder et al. 1991], which is a three-valued semantics for logic programs with negation that has also been considered for combination of rules and ontologies [Knorr et al. 2008; Drabent et al. 2007]. Three-valued extensions of autoepistemic logic [Denecker et al. 2003; Bonatti 1995; Przymusinski 1991b] may be used as a starting point.

Lastly, the initial motivation for our work has been the application to Semantic Web languages. Combinations of positive RIF rules with RDF and OWL DL ontologies, as we have discussed in the present paper, are just a first step. Nonmonotonic extensions of RIF [Kifer 2008], and also the RDF Query Language SPARQL [2008], are instances of the combination problems we have sketched in the present paper. The semantics of both nonmonotonic RIF and SPARQL can be expressed in terms of nonmonotonic logic programs [Kifer 2008; Angles and Gutierrez 2008; Polleres 2007], but their combination with OWL ontologies is still an open issue on W3C's agenda in completing the Semantic Web architecture [Bratt 2007]. We expect that our results can be used to provide valuable insights towards the definition of a *unifying logic* encompassing the Semantic Web Ontology (OWL, RDFS), Rules (RIF) and Query languages (SPARQL).

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: `http://www.acm.org/pubs/citations/journals/tocl/20YY-V-N/p1-debruijn`.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their valuable comments which helped to improve the paper.

REFERENCES

ABITEBOUL, S., HULL, R., AND VIANU, V. 1995. *Foundations of Databases.* Addison-Wesley.

ANALYTI, A., ANTONIOU, G., DAMÁSIO, C. V., AND WAGNER, G. 2008. Extended RDF as a semantic foundation of rule markup languages. *Journal of Artificial Intelligence Research 32*, 37–94.

ANGLES, R. AND GUTIERREZ, C. 2008. The expressive power of SPARQL. In *Proc. 7th International Semantic Web Conference* (*ISWC'08*). LNCS, vol. 5318. Springer, Karlsruhe, Germany, 114–129.

BAADER, F. AND HOLLUNDER, B. 1995. Embedding defaults into terminological knowledge representation formalisms. *Journal of Automated Reasoning 14*, 149–180.

BONATTI, P., LUTZ, C., AND WOLTER, F. 2006. Expressive non-monotonic description logics based on circumscription. In *Proc. 10th International Conference on Principles of Knowledge Representation and Reasoning* (*KR'06*). AAAI Press, 400–410.

BONATTI, P. A. 1995. Autoepistemic logics as a unifying framework for the semantics of logic programs. *Journal of Logic Programming 22,* 2, 91–149.

BRATT, S. 2007. Semantic web, and other technologies to watch. Available at `http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/`.

CALÌ, A., GOTTLOB, G., AND LUKASIEWICZ, T. 2009. Datalog$^{\pm}$: A unified approach to ontologies and integrity constraints. In *Proc. 12th International Conference on Database Theory* (*ICDT'09*). ACM Press, 14–30.

CHEN, J. 1993. Minimal knowledge + negation as failure = only knowing (sometimes). In *Proc. 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning* (*LPNMR'93*). MIT Press, 132–150.

DE BRUIJN, J., EITER, T., POLLERES, A., AND TOMPITS, H. 2006. On representational issues about combinations of classical theories with nonmonotonic rules. In *Proc. 1st International Conference on Knowledge Science, Engineering and Management* (*KSEM'06*). Springer, 1–22.

DE BRUIJN, J., EITER, T., AND TOMPITS, H. 2008. Embedding approaches to combining rules and ontologies into autoepistemic logic. In *Proc. 11th International Conference on Principles of Knowledge Representation and Reasoning* (*KR'08*). AAAI Press, 485–495.

DE BRUIJN, J. AND HEYMANS, S. 2007. Logical foundations of (e)RDF(S): Complexity and reasoning. In *Proc. 6th International Semantic Web Conference* (*ISWC'07*). Springer, 86–99.

DE BRUIJN, J., PEARCE, D., POLLERES, A., AND VALVERDE, A. 2007. Quantified equilibrium logic and hybrid rules. In *Proc. 1st International Conference on Web Reasoning and Rule Systems* (*RR'07*). Springer, 58–72.

DENECKER, M., MAREK, V. W., AND TRUSZCZYNSKI, M. 2003. Uniform semantic treatment of default and autoepistemic logics. *Artificial Intelligence 143,* 1, 79–122.

DONINI, F. M., LENZERINI, M., NARDI, D., NUTT, W., AND SCHAERF, A. 1998. An epistemic operator for description logics. *Artificial Intelligence 100,* 1–2, 225–274.

DONINI, F. M., LENZERINI, M., NARDI, D., AND SCHAERF, A. 1998. $\mathcal{AL}$-log: integrating datalog and description logics. *Journal of Intelligent Information Systems 10*, 227–252.

DONINI, F. M., NARDI, D., AND ROSATI, R. 2002. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic 3,* 2, 177–225.

DRABENT, W., HENRIKSSON, J., AND MALUSZYNSKI, J. 2007. Hybrid reasoning with rules and constraints under well-founded semantics. In *Proc. 1st International Conference on Web Reasoning and Rule Systems* (*RR'07*). Springer, 348–357.

EITER, T., IANNI, G., LUKASIEWICZ, T., SCHINDLAUER, R., AND TOMPITS, H. 2008. Combining answer set programming with description logics for the semantic web. *Artificial Intelligence 172,* 12-13, 1495–1539.

EITER, T., IANNI, G., SCHINDLAUER, R., AND TOMPITS, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *Proc. 19th International Joint Conference on Artificial Intelligence* (*IJCAI'05*). Morgan Kaufmann, 90–96.

FABER, W., LEONE, N., AND PFEIFER, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Proc. 9th European Conference on Logics in Artificial Intelligence* (*JELIA'04*). Springer, 200–212. Full paper to appear in *Artificial Intelligence.*

FERRARIS, P. 2005. Answer sets for propositional theories. In *Proc. 8th International Conference on Logic Programming and Nonmonotonic Reasoning* (*LPNMR'05*). Springer, 119–131.

FITTING, M. 1996. *First Order Logic and Automated Theorem Proving* (*2nd ed.*). Springer, New York.

GELDER, A. V., ROSS, K., AND SCHLIPF, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 620–650.

GELFOND, M. 1987. On stratified autoepistemic theories. In *Proc. 6th National Conference on Artificial Intelligence* (*AAAI'87*). Morgan Kaufmann, 207–211.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proc. 5th International Conference on Logic Programming* (*ICLP'88*). MIT Press, 1070–1080.

GELFOND, M. AND LIFSCHITZ, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9,* 3–4, 365–386.

GELFOND, M. AND PRZYMUSINSKA, H. 1993. Reasoning on open domains. In *Proc. 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning* (*LPNMR'93*). MIT Press, 397–413.

HEYMANS, S., DE BRUIJN, J., PREDOIU, L., FEIER, C., AND NIEUWENBORGH, D. V. 2008. Guarded hybrid knowledge bases. *Theory and Practice of Logic Programming 8,* 3, 411–429.

Heymans, S., van Nieuwenborgh, D., and Vermeir, D. 2006. Conceptual logic programs. *Annals of Mathematics and Artificial Intelligence 47,* 1–2, 103–137.

Horrocks, I. and Patel-Schneider, P. F. 2003. Reducing OWL entailment to description logic satisfiability. In *Proc. 2nd International Semantic Web Conference* (*ISWC'03*). Springer, 17–29.

Horrocks, I., Patel-Schneider, P. F., Bechhofer, S., and Tsarkov, D. 2005. OWL rules: A proposal and prototype implementation. *Journal of Web Semantics 3,* 1, 23–40.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. 2004. SWRL: A semantic web rule language combining OWL and RuleML. Member submission 21 may 2004, W3C.

Hustadt, U., Motik, B., and Sattler, U. 2005. Data complexity of reasoning in very expressive description logics. In *Proc. 19th International Joint Conference on Artificial Intelligence* (*IJCAI'05*). AAAI Press, 466–471.

Janhunen, T. 1999. On the intertranslatability of non-monotonic logics. *Annals of Mathematics and Artificial Intelligence 27,* 1-4, 79–128.

Kaminski, M. and Rey, G. 2002. Revisiting quantification in autoepistemic logic. *ACM Transactions on Computational Logic 3,* 4, 542–561.

Kifer, M. 2008. Rule interchange format: The framework. In *Proc. 2nd International Conference on Web Reasoning and Rule Systems* (*RR'08*). LNCS, vol. 5341. Springer, 1–11.

Knorr, M., Alferes, J. J., and Hitzler, P. 2008. A coherent well-founded model for hybrid MKNF knowledge bases. In *Proc. 18th European Conference on Artificial Intelligence* (*ECAI'08*). IOS Press.

Konolige, K. 1991. Quantification in autoepistemic logic. *Fundamenta Informaticae 15,* 3–4, 275–300.

Levesque, H. J. 1990. All I know: A study in autoepistemic logic. *Artificial Intelligence 42,* 2–3, 263–309.

Levesque, H. J. and Lakemeyer, G. 2000. *The Logic of Knowledge Bases.* MIT Press, Cambridge, MA, USA.

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

Lifschitz, V. 1991. Nonmonotonic databases and epistemic queries. In *Proc. 12th International Joint Conference on Artificial Intelligence* (*IJCAI'91*). Morgan Kaufmann, 381–386.

Lifschitz, V. 1994. Minimal belief and negation as failure. *Artificial Intelligence 70,* 1–2, 53–72.

Lifschitz, V. and Schwarz, G. 1993. Extended logic programs as autoepistemic theories. In *Proc. 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning* (*LP-NMR'93*). MIT Press, 101–114.

Lifschitz, V., Tang, L. R., and Turner, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence 25,* 3–4, 369–389.

Marek, V. W. and Truszczyński, M. 1993. *Nonmonotonic Logic: Context-dependent Reasoning.* Artificial Intelligence. Springer-Verlag, Berlin Heidelberg.

Marek, V. W. and Truszczynski, M. 1993. Reflexive autoepistemic logic and logic programming. In *Proc. 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning* (*LPNMR'93*). MIT Press, 115–131.

McCarthy, J. 1986. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence 28*, 89–116.

McDermott, D. 1982. Nonmonotonic logic II: Nonmonotonic modal theories. *Journal of the ACM 29,* 1, 33–57.

Moore, R. C. 1985. Semantical considerations on nonmonotonic logic. *Artificial Intelligence 25,* 1, 75–94.

Motik, B., Horrocks, I., Rosati, R., and Sattler, U. 2006. Can OWL and logic programming live together happily ever after? In *Proc. 5th International Semantic Web Conference* (*ISWC'06*), I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, Eds. Lecture Notes in Computer Science, vol. 4273. Springer, 501–514.

Motik, B. and Rosati, R. 2007. A faithful integration of description logics with logic programming. In *Proc. 20th International Joint Conference on Artificial Intelligence* (*IJCAI'07*). AAAI Press, 477–482.

Motik, B., Sattler, U., and Studer, R. 2005. Query answering for OWL-DL with rules. *Journal of Web Semantics 3,* 1, 41–60.

Niemelä, I. 1992. On the decidability and complexity of autoepistemic reasoning. *Fundamenta Informaticae 17,* 1–2, 117–155.

OWL 2 2009. OWL 2 web ontology language structural specification and functional-style syntax. Recommendation 27 October 2009, W3C.

OWL 2 Profiles 2009. OWL 2 web ontology language profiles. Recommendation 27 October 2009, W3C.

OWL Semantics 2004. OWL web ontology language semantics and abstract syntax. Recommendation 10 February 2004, W3C.

Pearce, D. and Valverde, A. 2005. A first-order nonmonotonic extension of constructive logic. *Studia Logica 80*, 321–246.

Pelov, N., Denecker, M., and Bruynooghe, M. 2007. Well-founded and stable semantics of logic programs with aggregates. *Theory and Practice of Logic Programming 7,* 3, 301–353.

Polleres, A. 2007. From SPARQL to rules (and back). In *Proc. 16th International World Wide Web Conference* (*WWW'07*). ACM Press, Banff, Canada, 787–796.

Przymusinski, T. C. 1991a. Stable semantics for disjunctive programs. *New Generation Computing 9,* 3–4, 401–424.

Przymusinski, T. C. 1991b. Three-valued nonmonotonic formalisms and semantics of logic programs. *Artificial Intelligence 49,* 1–3, 309–343.

RDF Concepts 2004. Resource description framework (RDF): Concepts and abstract syntax. Recommendation 10 February 2004, W3C.

RDF Schema 2004. RDF vocabulary description language 1.0: RDF Schema. Recommendation 10 February 2004, W3C.

RDF Semantics 2004. RDF semantics. Recommendation 10 February 2004, W3C.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence 13*, 81–132.

RIF BLD 2009. RIF basic logic dialect. Recommendation 22 June 2010, W3C.

RIF RDF-OWL 2009. RIF RDF and OWL compatibility. Recommendation 22 June 2010, W3C.

Rosati, R. 1997. Embedding minimal knowledge into autoepistemic logic. In *Proc. 6th Conference of the Italian Association for AI* (*AI*IA-97*). Springer, 231–241.

Rosati, R. 1999. Towards first-order nonmonotonic reasoning. In *Proc. 5th International Conference on Logic Programming and Nonmonotonic Reasoning* (*LPNMR'99*). Springer, 332–346.

Rosati, R. 2005. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics 3,* 1, 61–73.

Rosati, R. 2006. *$\mathcal{DL}$+log*: Tight integration of description logics and disjunctive datalog. In *Proc. 10th International Conference on Principles of Knowledge Representation and Reasoning* (*KR'06*). AAAI Press, 68–78.

Sattler, U., Calvanese, D., and Molitor, R. 2003. Relationships with other formalisms. In *The Description Logic Handbook*, F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds. Cambridge University Press, Cambridge, UK, Chapter 4, 142–183.

Schwarz, G. 1992. Reflexive autoepistemic logic. *Fundamenta Informaticae 17,* 1–2, 157–173.

Shvarts, G. F. 1990. Autoepistemic modal logics. In *Proc. 3rd Conference on Theoretical Aspects of Reasoning about Knowledge* (*TARK'90*). Morgan Kaufmann, 97–109.

Son, T. C. and Pontelli, E. 2007. A constructive semantic characterization of aggregates in answer set programming. *Theory and Practice of Logic Programming 7,* 3, 355–375.

SPARQL 2008. SPARQL Query Language for RDF. Recommendation 15 January 2008, W3C. Jan.

Stalnaker, R. 1993. A note on non-monotonic modal logic. *Artificial Intelligence 64,* 2, 183–196.

Van Belleghem, K., Denecker, M., and De Schreye, D. 1997. A strong correspondence between description logics and open logic programming. In *Proc. 14th International Conference on Logic Programming* (*ICLP97*). MIT Press, 346–360.

Received November 2008; revised June 2010; accepted June 2010