A Framework for Modular ERDF Ontologies

Anastasia Analyti[1], Grigoris Antoniou[1,2],
Carlos Viegas Damásio[3], and Ioannis Pachoulakis[4]

[1] Institute of Computer Science, FORTH-ICS, Greece
[2] Department of Informatics, University of Huddersfield, UK
[3] CENTRIA, Departamento de Informática da Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[4] Dept. of Applied Informatics & Multimedia, TEI of Crete, Greece
`analyti@ics.forth.gr, G.Antoniou@hud.ac.uk, cd@fct.unl.pt, ip@epp.teicrete.gr`

Abstract. The success of the Semantic Web is impossible without any form of modularity, encapsulation, and access control. In an earlier paper, we extended RDF graphs with weak and strong negation, as well as derivation rules. The ERDF #n-stable model semantics of the extended RDF framework (*ERDF*) is defined, extending RDF(S) semantics. In this paper, we propose a framework for modular ERDF ontologies, called *modular ERDF framework*, which enables collaborative reasoning over a set of ERDF ontologies, while support for hidden knowledge is also provided. In particular, the *modular ERDF stable model semantics* of modular ERDF ontologies is defined, extending the ERDF #n-stable model semantics. Our proposed framework supports local semantics and different points of view, local closed-world and open-world assumptions, and scoped negation-as-failure. Several complexity results are provided.
Keywords: Modular ERDF ontologies, local semantics, local closed-world and open-world assumptions, scoped negation-as-failure.

1 Introduction

Semantic Web is taking shape by providing data, information, and knowledge in the form of Linked Data[5]. Each dataset published can be understood as a module of knowledge, usually expressed by ontologies in the standardized Resource Description Framework (RDF) and RDF Schema vocabulary [49, 42]. These datasets are made publicly accessible and can be queried via SPARQL endpoints. SPARQL 1.0 restricts users to querying a dataset in a single SPARQL endpoint, while the latest SPARQL 1.1 version of the language [41] permits the use of complex federated monotonic and nonmonotonic queries that can combine together the information from several endpoints, as well as the update of information in graph stores. These federated queries are a recent addition to the SPARQL language [8, 62] and allow distributed queries over remote SPARQL endpoints. Graph stores and their datasets are interlinked in the Semantic Web, but are not connected – users are still required to use non-standard and non-declarative mechanisms to combine their datasets with other datasets (e.g. by data replication and procedural code, as argued in [65]).

However, the recent SPARQL 1.1 set of recommendations [41] still does not address important issues:

Modularity: How should knowledge (and datasets) in the Semantic Web be organized to foster data linking?

Visibility: What information should be made publicly available, and what information is for private or registered use?

[5] `http://linkeddata.org`

Access Control: How do graph stores define and provide policies for user's access control?

The problems are challenging and require adequate (formal) languages for knowledge representation and reasoning, with appropriate model-theoretic semantics. In fact, logic programming based declarative languages, and in particular datalog, have been used to address all of the above, most of the times in a compartmented way [56, 45, 17, 18, 14, 15, 1, 51, 20]. In this paper, we propose a framework based on stable model semantics/answer set programming [31] that is capable of addressing the first above two issues, and powerful enough to fully capture the SPARQL semantics.

Regarding modularization of knowledge, the current Semantic Web languages endorsed by W3C, like Rule Interchange Format (RIF) [47], SPARQL, and Web Ontology Language (OWL), have limited modularization mechanisms. We are in a stage where the combination of data in the web is performed by union/merging of all information originating from the specified sources, in an uncontrolled way. OWL 2.0 has just a basic textual import mechanism, while RIF-FLD provides two ways of linking documents (import and module directives) consisting of rules. In fact, the initial semantics of RIF-FLD multi-documents has been shown by ourselves to be incorrect [23] and has been recently updated to fix the problems according to our suggestions [13], resulting from our previous attempts to modularize RDF-based ontologies [2, 3, 22, 4, 23] (preliminary work to the current proposals presented in this paper). SPARQL semantics resorts to the notion of datasets, formed by a default graph (set of triples) and several named graphs identified by an IRI. These datasets are kept in graph stores, which can be updated in the novel SPARQL 1.1 by the usual insert/delete/update operations. Using the RIF-Core entailment regime, SPARQL is able to mixture rules specified in RIF-Core with RDF graphs. Unfortunately, RIF-Core does not include negation formulas and cannot reason with information defined in remote modules.

Regarding visibility, RIF, SPARQL, and OWL do not provide support for information hiding (everything is visible to everyone). Currently, users are forced to (partially) replicate their datasets in order to project their information and in order to hide it, not having the notion of "graph view" to support them in this process [65]. Most of the times, visibility is controlled by implementation dependent mechanisms at the underlying database level of the graph stores.

Other works [52] define access control policies using datalog requiring integrity constraints or strong negation for expressing acceptance and denial of access to data. We will ignore this last issue in the present work, but our approach can be extended easily to handle them since it has the possibility to express authorization and denial of access.

Additionally, there are known expressivity limitations of RDF and SPARQL that can be solved using answer-set based semantics. The SPARQL language provides complex query mechanisms over RDF graph stores, but does not define the notion of graph view. The advantages and expressiveness gains of graph views have been defended in [65], and allow the definition of recursive graph views. These recursive graph views (mutual dependencies among graph views in graph stores) will naturally occur in the Semantic Web. For instance, they will allow that a graph view can be constructed from a query to a remote view in a graph store which depends on the former graph view, in order to cooperate in the definition of some shared knowledge. SPARQL does not specify an appropriate semantics for the case of recursive queries. In fact, the only recursive mechanisms the language offers are property paths for performing (reflexive or not) transitive closures of properties. In our view, this limitation comes from the fact that SPARQL has monotonic and nonmonotonic constructs[6], and aggregates, which can be combined together. It is well-known that datalog programs under stable model semantics is more expressive than SPARQL, and embeddings have been reported in the literature (see [59, 61]).The semantics for the general recursive queries case is possible, by the mapping into stable models semantics [30] for which even exist extensions capable of dealing with recursive aggregates [27]. Recursive views over SPARQL have been reported in the literature [65], resulting in Networked Graphs an extension of RDF graphs, including distributed evaluation with a semantics based on Well-founded Semantics, which is less expressive than the Stable Model/Answer Set Semantics adopted in the current work.

Ontologies and automated reasoning are the building blocks of the Semantic Web initiative. Derivation rules can be included in an ontology to define derived concepts based on base concepts. For example, rules allow to define the extension of a class or property based on a complex relation between the extensions of the same or other classes and properties (e.g. $\texttt{ex:choose\_trav\_package}(\texttt{ex:Ann}, ?c) \leftarrow \texttt{ex:likes\_package}(\texttt{ex:Peter}, ?c)$, $\texttt{ex:packageOf}(?c, \texttt{ex:Pyramis})$). The current SPARQL 1.1 semantics defines an entailment regime that permits limited forms of combination of rules with ontologies, as discussed before. On the other hand, the inclusion of negative information both in the form of negation-as-failure (e.g. for capturing the OPTIONAL construct of the SPARQL query language) and explicit negative information is also needed to enable various forms of reasoning (e.g. express access denials, that woman are not man, likes and dislikes in social web, polarity of sentiment analysis, etc.). The importance of representing negative knowledge has been recognized in the December 2012 OWL 2 recommendation [71, 36], where syntactic mechanisms have been introduced to express negative object and data property assertions (negative facts). In the OWL 2 Primer [44] is stated that "Negative property assertions provide a unique opportunity to make statements, where we know something that is not true. This kind of information is particularly important in OWL, where the default stance is that anything is possible until you say otherwise." As we have been arguing for several years [6], such a mechanism is even more important for the RDF and RDF schema, since there is no way in the language to represent negative information, e.g. that Bill does not a wife named Mary, or that a patient does not live in the Île-de-France Region – example and use case of the OWL 2 language [44, 36]). In [6], the Semantic Web language RDFS [49, 42] is extended to accommodate the two negations of Partial Logic [43], namely *weak negation* $\sim$ (expressing negation-as-failure or non-truth) and *strong negation* $\neg$ (expressing explicit negative information or falsity), as well as derivation rules. The new language is called *Extended RDF* (*ERDF*). Specifically, in [6], the *stable model semantics* of ERDF ontologies is developed, based on Partial Logic, extending the model-theoretic semantics of RDFS [42]. By adopting stable model semantics we are able to define a language capable of addressing in an unified way the two major issues identified at the beginning of this section.

Intuitively, an ERDF ontology is the combination of (i) an ERDF graph G containing (implicitly existentially quantified) positive and negative information, and (ii) an ERDF

---

[6] SPARQL Query Language is designed to deal with incomplete data, and provides several constructs to do that, e.g. OPTIONAL, MINUS, and NOT EXISTS. The most relevant in practice is OPTIONAL, used to complement solutions with information that for some cases may be absent. For instance, the query `SELECT * WHERE { { ?ctr a geo:Country } OPTIONAL { ?ctr geo:capital ?cap } }` will return all the countries in the dataset, as well as the corresponding capitals, whenever the latter are known (for those countries without a capital in the dataset, the variable `?cap` will be unbound). The previous query with an additional condition FILTER (!BOUND(?cap)) becomes nonmonotonic, since it returns all the countries without a capital. It is well-known that the OPTIONAL construct increases expressivity of the language from NP-complete to PSPACE-complete [58]. For more details see [41].

program P containing derivation rules, with possibly all connectives $\sim$, $\neg$, $\supset$, $\wedge$, $\vee$, $\forall$, $\exists$ in the body of a rule, and strong negation $\neg$ in the head of a rule.

In [6], it is shown that stable model entailment conservatively extends RDFS entailment from RDF graphs to ERDF ontologies. Unfortunately, satisfiability and entailment under the ERDF stable model semantics are in general undecidable. This is due to the fact that the RDF vocabulary is infinite. Therefore, to achieve decidability of reasoning in the general case, in [7, 5], we propose a modified semantics, called *ERDF #n-stable model semantics* (for $n \in I\!N$), in which from the RDF vocabulary, we remove the infinite set of terms $\{rdf{:}\_i \mid i > n\}$. The new semantics also extends RDFS entailment from RDF graphs to ERDF ontologies. Additionally, in [7, 5], we provide an equivalence statement between ERDF stable model entailment and ERDF $\#n$-stable model entailment on an ERDF ontology O, in the case that the bodies of the rules in O contain only the connectives $\neg$ and $\wedge$.

As we have argued, the success of the Semantic Web is impossible without any form of modularity, encapsulation, and access control. In this paper, we propose a framework for modular ERDF ontologies, called *modular ERDF framework*, in which a modular ERDF ontology $\mathcal{R}$ is a set of `r`-ERDF ontologies. Intuitively, an `r`-ERDF ontology[7] $O \in \mathcal{R}$ is an ERDF ontology that can import or just reference knowledge about a property or class x from other `r`-ERDF ontologies in $\mathcal{R}$ that define x and are willing to export this knowledge to O. Thus, our modular ERDF framework enables collaborative reasoning over a set of `r`-ERDF ontologies, while support for hidden knowledge is also provided. Additionally, it supports local semantics and different points of view, local closed-world and open-world assumptions, and scoped negation-as-failure. All the referred mechanisms are necessary to address the initial identified issues. In particular, visibility issues are important because an ontology may not want to export all properties and classes to other ontologies in $\mathcal{R}$, for example for safety. Additionally, an ontology may not want to import all properties and classes from other ontologies in $\mathcal{R}$, for example for reasons of trust. Specifically, in this paper:

- We define the *modular* (*ERDF*) *stable models* of an `r`-ERDF ontology w.r.t. a modular ERDF ontology. Several properties of the modular stable model semantics are provided, including that modular stable model theory faithfully extends $\#n$-stable model entailment on ERDF ontologies, and thus also, faithfully extends RDFS entailment on RDF graphs. This property is very important because we show how our modular ERDF model theory compares with the $\#n$-stable model theory on ERDF ontologies and to the RDFS model theory on RDFS graphs.[8]
- We show that if $\mathcal{R}$ is a simple modular ERDF ontology (i.e., the bodies of the rules of the `r`-ERDF ontologies in $\mathcal{R}$ contain only the connectives $\sim$, $\neg$, $\wedge$) then query answering under the modular ERDF stable model semantics reduces to query answering under the answer set semantics [31].
- We provide algorithms that compute the modular ERDF stable models of simple and general modular ERDF ontologies, even when cyclic monotonic and non-monotonic definitions exist among the several modules.
- We provide complexity results for the modular ERDF stable model semantics on (i) simple modular ERDF ontologies, (ii) modular ERDF ontologies without quantifiers, and (ii) general modular ERDF ontologies. In particular, the complexity of query answering under the modular ERDF stable model semantics (i) on simple modular ERDF ontologies is Π_2^P = co-NP$^{\mathrm{NP}}$-complete, (ii) on modular ERDF ontologies

[7] Prefix `r` in `r`-ERDF stands for *remote*.

[8] Note that providing a model theory for a framework is important for understanding the semantics of the framework.

without quantifiers is co-NP$^{\text{NP}}$-complete, and (iii) on general modular ERDF ontologies is PSPACE-complete. So, our features come at no extra cost when compared to standard SPARQL reasoning, which has PSPACE-complete complexity [58].

We would like to mention that the goal of our modular ERDF framework is on interconnecting independently developed `r`-ERDF ontologies over the web and *not* on querying a large ontology by decomposing it into smaller sub-ontologies. The latter problem has been considered for answer set semantics in [56], but [56] prohibits the existence of positive recursion among modules, a serious limitation for the Semantic Web setting. In contrast, in our framework, considered `r`-ERDF ontologies may be interconnected via cyclic references. For example, an `r`-ERDF ontology O may be created any time after the independent creation of the `r`-ERDF ontologies on which it depends (which later may be updated, possibly referring to O).

This work extends our conference paper [3] by (i) showing that if $\mathcal{R}$ is a simple modular ERDF ontology then query answering reduces to query answering under the answer set semantics, (ii) providing algorithms that compute the modular ERDF stable models of simple and general modular ERDF ontologies, (iii) providing additional properties and examples, (iv) by extending related work, and (v) by providing proofs of all propositions.

The rest of the paper is organized as follows: In Section 2, we review open and closed-world assumptions in ERDF. In Section 3, we review ERDF graphs, which we extend to `r`-ERDF formulas. Then, we define `r`-ERDF ontologies and valid modular ERDF ontologies. Section 4 defines the modular ERDF interpretations of an `r`-ERDF ontology w.r.t. a modular ERDF ontology. Then, it defines satisfiability of an `r`-ERDF formula by such a modular ERDF interpretation and an `r`-ERDF ontology. In Section 5, we define the modular stable model semantics of an `r`-ERDF ontology w.r.t. a modular ERDF ontology, and provide its properties. Section 6 shows that if $\mathcal{R}$ is a simple modular ERDF ontology then query answering under the modular ERDF stable model semantics reduces to query answering under the answer set semantics. Additionally, it provides the complexity of query answering under the modular ERDF stable model semantics on simple modular ERDF ontologies. Section 7 provides an algorithm that computes the modular ERDF stable models of general modular ERDF ontologies. Additionally, it provides the complexity of query answering under the modular ERDF stable model semantics on modular ERDF ontologies without quantifiers and general modular ERDF ontologies. Section 8 reviews related work and Section 9 concludes the paper. Appendix A reviews for self-containment the ERDF $\#n$-stable model semantics, provided in [7, 5]. Appendix B provides the proofs of all Propositions. Finally, Appendix C provides a table of symbols.

2 Open and Closed-World Assumptions in ERDF

ERDF [6] enables the combination of closed-world (non-monotonic) and open-world (monotonic) reasoning, in the same framework, through the presence of weak negation (in the body of the program rules) and the new metaclasses *erdf*:*TotalProperty* and *erdf*:*TotalClass*, respectively. Therefore, we provide the essential monotonic and non-monotonic forms of reasoning necessary to encode SPARQL. In particular, relating strong and weak negation at the interpretation level, ERDF distinguishes two categories of properties and classes [6]. *Partial properties* are properties p that may have truth-value gaps, that is $p(x, y)$ is possibly neither true nor false (e.g. a user may not like or dislike a particular web page). *Total properties* are properties p that satisfy *totalness*, that is $p(x, y)$ is either true or false. Partial and total classes c are defined similarly, by replacing $p(x, y)$ by *rdf*:*type*(x, c). In [6], it is shown that on total properties and total classes, the *Open-World Assumption (OWA)* applies.

ERDF also distinguishes properties and classes that are completely represented in a knowledge base with respect to an (optional) ERDF formula F, corresponding to the *context* where the completion takes place. Such a completeness assumption for *closing* a partial property p by default may be expressed in ERDF by means of the rule $\neg p(?x, ?y) \leftarrow F \wedge \sim p(?x, ?y)$ and for a partial class c, by means of the rule $\neg rdf{:}type(?x, c) \leftarrow F \wedge \sim rdf{:}type(?x, c)$, where F is an ERDF formula.

In particular if a class or property is declared as total, this means that information in the ERDF ontology is not complete about this class or property. Additionally, if information about a class or property in the ERDF ontology is complete then one of the above completeness rules applies.

3 Modular ERDF Ontologies

In this Section, we define r-ERDF formulas, valid r-ERDF ontologies, and valid modular ERDF ontologies. Additionally, we provide a comprehensive example of a modular ERDF ontology.

A (Web) *vocabulary* V is a set of URI references and/or literals (plain or typed) [42]. We denote the set of all URI references by *URI*, the set of all plain literals by $\mathcal{PL}$, and the set of all typed literals by $\mathcal{TL}$.[9] We consider a set *Var* of variable symbols, such that the sets *Var*, *URI*, $\mathcal{PL}$, and $\mathcal{TL}$ are pairwise disjoint. In our examples, variable symbols are prefixed by "?".

Below, we review the definition of an ERDF triple from [6]. Let V be a vocabulary. A (*normal*) *ERDF triple* over V is an expression of the form $p(s, o)$ or $\neg p(s, o)$, where $s, o \in V \cup Var$ are called *subject* and *object*, respectively, and $p \in V \cap URI$ is called *property*.

Below we extend the definition of an ERDF formula, provided in [6], to r-ERDF formulas. We consider the connectives $\{\sim, \neg, \wedge, \vee, \supset, \exists, \forall\}$, where $\neg$, $\sim$, and $\supset$ are called *strong negation*, *weak negation*, and *material implication* respectively. Let V be a vocabulary and let $\mathcal{O}_{\mathtt{nam}} \subseteq URI$ be a set of r-ERDF ontology names. We define $L(V)$ to be the smallest set that contains the ERDF triples over V and is closed with respect to the following conditions: if $F, G \in L(V)$ then $\{(\sim F), (\neg F), (F \wedge G), (F \vee G), (F \supset G), (\exists x F), (\forall x F)\} \subseteq L(V)$, where $x \in Var$. A (*normal*) *ERDF formula* over V is an element of $L(V)$. A *qualified ERDF formula* over V and $\mathcal{O}_{\mathtt{nam}}$ has the form $F@oname$, where $F \in L(V)$ and $oname \in \mathcal{O}_{\mathtt{nam}}$ (i.e., F will be evaluated at the r-ERDF ontology identified by $oname$).

Definition 1 (r-ERDF formula). Let V be a vocabulary and let $\mathcal{O}_{\mathtt{nam}} \subseteq URI$. We define $L(V, \mathcal{O}_{\mathtt{nam}})$ to be the smallest set that (i) contains the ERDF formulas over V and the qualified ERDF formulas over V and $\mathcal{O}_{\mathtt{nam}}$, and (ii) is closed with respect to the following conditions: if $F, G \in L(V, \mathcal{O}_{\mathtt{nam}})$ then $\{(\sim F), (\neg F), (F \wedge G), (F \vee G), (F \supset G), (\exists x F), (\forall x F)\} \subseteq L(V, \mathcal{O}_{\mathtt{nam}})$, where $x \in Var$. An r-*ERDF formula* F over V and $\mathcal{O}_{\mathtt{nam}}$ is an element of $L(V, \mathcal{O}_{\mathtt{nam}})$. We denote the set of variables appearing in F by $Var(F)$, and the set of free variables[10] appearing in F by $FVar(F)$. □

Next, we review the definition of an ERDF graph G and the skolemization of G from [6]. An *ERDF graph* G over a vocabulary V is a set of ERDF triples over V. We denote the variables appearing in G by $Var(G)$, and the set of URI references and

[9] The symbols $\mathcal{PL}$ and $\mathcal{TL}$ appear in the definition of a partial interpretation (Definition 6), extending the definition of a simple interpretation of RDFS semantics.

[10] Without loss of generality, we assume that a variable cannot have both free and bound occurrences in F, and more than one bound occurrence.

literals appearing in G by V_G. Intuitively, an ERDF graph G represents an existentially quantified conjunction of ERDF triples. Specifically, let $G = \{t_1, ..., t_m\}$ be an ERDF graph, and let $Var(G) = \{x_1, ..., x_k\}$. Then, G represents the ERDF formula $formula(G) = \exists x_1, ..., \exists x_k\ t_1 \wedge ... \wedge t_m$. Existentially quantified variables in ERDF graphs are handled by *skolemization*. Let G be an ERDF graph. The *skolemization function* of G is an 1:1 mapping $sk_G : Var(G) \rightarrow URI$, where for each $x \in Var(G)$, $sk_G(x)$ is an artificial URI, denoted by $G{:}x$. The *skolemization* of G, denoted by $sk(G)$, is the ground ERDF graph derived from G after replacing each $x \in Var(G)$ by $sk_G(x)$.

Below, we extend the definitions of ERDF rule and ERDF program, provided in [6], to `r`-ERDF rule and `r`-ERDF program, respectively.

Definition 2 (`r`-ERDF rule, `r`-ERDF program). An `r`-*ERDF rule* r over a vocabulary V and $\mathcal{O}_{\texttt{nam}} \subseteq URI$ is an expression of the form: $G \leftarrow F$, where $F \in L(V, \mathcal{O}_{\texttt{nam}}) \cup \{\texttt{true}\}$ (called *condition*) and G (called *conclusion*) is either an ERDF triple over V or `false`. We assume that no bound variable in F appears free in G. We denote the set of variables and the set of free variables of r by $Var(r)$ and $FVar(r)$[11], respectively. Additionally, we write $Cond(r) = F$ and $Concl(r) = G$.
An `r`-*ERDF program* P over a vocabulary V and $\mathcal{O}_{\texttt{nam}} \subseteq URI$ is a finite set of `r`-ERDF rules over V and $\mathcal{O}_{\texttt{nam}}$. We denote the set of URI references and literals appearing in P by V_P. $\square$

Below, we extend the definition of an ERDF ontology, provided in [6], to an `r`-ERDF ontology.

Definition 3 (`r`-ERDF ontology). An `r`-*ERDF ontology* O over a vocabulary V and $\mathcal{O}_{\texttt{nam}} \subseteq URI$ is a triple $O = \langle Nam_O, L_O, Int_O \rangle$, where: (i) $Nam_O \in \mathcal{O}_{\texttt{nam}}$ is the *name* of O, (ii) $L_O = \langle G_O, P_O, \rangle$, is the *logic* of O, where G_O is an ERDF graph over V and P_O is an `r`-ERDF program over V and $\mathcal{O}_{\texttt{nam}}$, and (iii) $Int_O = \langle Export_O^{\texttt{pr}}, Export_O^{\texttt{cl}}, Import_O^{\texttt{pr}}, Import_O^{\texttt{cl}} \rangle$ is the *interface* of O, where: For $\texttt{t} \in \{\texttt{pr}, \texttt{cl}\}$, it holds that:

- $Export_O^{\texttt{t}}$ is a (partial) function from $V \cap URI$ to $\mathcal{P}(\mathcal{O}_{\texttt{nam}} - \{Nam_O\}) \cup \{*\}$.[12] We define: $Exported_O^{\texttt{t}} = \{x \mid Export_O^{\texttt{t}}(x) \text{ is defined }\}$.
- $Import_O^{\texttt{t}}$ is a (partial) function from $V \cap URI$ to $\mathcal{P}(\mathcal{O}_{\texttt{nam}} - \{Nam_O\}) \cup \{*\}$. We define: $Imported_O^{\texttt{t}} = \{x \mid Import_O^{\texttt{t}}(x) \text{ is defined }\}$.

Let O be an `r`-ERDF ontology. Intuitively, each statement $Export^{\texttt{pr}}(x) = Exp$ (resp. $Export^{\texttt{cl}}(x) = Exp$) corresponds to an `export` declaration of O, where x is a property (resp. class) exported by O and Exp is the list of `r`-ERDF ontologies to which O is willing to export x. If O is willing to export x to any requesting `r`-ERDF ontology then $Exp = *$.

Similarly, $Import^{\texttt{pr}}(x) = Imp$ (resp. $Import^{\texttt{cl}}(x) = Imp$) corresponds to an `import` declaration of O, where x is a property (resp. class) requested by O, and Imp is the list of `r`-ERDF ontologies from which x is requested. If O requests x from any providing `r`-ERDF ontology then $Imp = *$.

Definition 4 (Modular ERDF ontology). A *modular ERDF ontology* $\mathcal{R}$ is a set of `r`-ERDF ontologies. $\square$

Example 1. Consider the modular ERDF ontology $\mathcal{R} = \{O_1, O_2, O_3, O_4, O_5\}$, shown in Figure 1[13]. Ontology O_1, with $Nam_{O_1} =$`<http://geography.int>`, provides geographical information, stating that the list of European countries is positively closed (w.r.t.

[11] $FVar(r) = FVar(F) \cup FVar(G)$.
[12] Let S be a set. By $\mathcal{P}(S)$, we denote the powerset of S.
[13] Following usual convention, we have replaced $\wedge$ by "," in the program rules.

Ontology O_1

```
⟨http://geography.int⟩
exports class geo:Country to ∗.
exports class geo:Europ_Country to ∗.
exports property geo:capital to ∗.
G_O1 =
rdfs:subclass(geo:Europ_Country,geo:Country).
rdf:type(geo:Egypt,geo:Country).
rdf:type(geo:Italy,geo:Europ_Country).
rdf:type(geo:Croatia,geo:Europ_Country).
geo:capital(geo:Cairo,geo:Egypt).
geo:capital(geo:Zagreb,geo:Croatia).   ···
P_O1 =
¬ rdf:type(?x,geo:Europ_Country) ←
        rdf:type(?x,geo:Country),
      ∼ rdf:type(?x,geo:Europ_Country).
```

Ontology O_2

```
⟨http://europa.eu⟩
imports class geo:Country from
          ⟨http://geography.int⟩.
exports class eu:CountryEU to ∗.
G_O2 =
rdf:type(eu:CountryEU,erdf:TotalClass).
rdf:type(geo:Italy,eu:CountryEU).
rdf:type(geo:Greece,eu:CountryEU). ···
```

Ontology O_3

```
⟨http://www.pyramis.gr⟩
exports property vac:travel to ∗.
exports property vac:visit to ∗.
G_O3 =
vac:travel(pyr:package1,geo:Egypt).
vac:visit(pyr:package1,geo:Cairo).
vac:travel(pyr:package2,geo:Egypt).
vac:visit(pyr:package2,geo:Cairo).
vac:visit(pyr:package2,geo:Luxor).
```

Ontology O_4

```
⟨http://www.travel_plan.gr⟩
exports property vac:travel to ∗.
exports property vac:visit to ∗.
G_O4 =
vac:travel(trav:package1,geo:Italy).
vac:visit(trav:package1,geo:Rome).
vac:travel(trav:package2,geo:Croatia).
vac:visit(trav:package2,geo:Zagreb).
vac:visit(trav:package2,geo:Trogir).
```

Ontology O_5

```
⟨http://www.anne_travel_pref.gr⟩
imports class geo:Europ_Country from ⟨http://geography.int⟩.
imports property geo:capital from ⟨http://geography.int⟩.
imports class eu:CountryEU from ⟨http://europa.eu⟩.
imports property vac:travel from ∗.
imports property vac:visit from ∗.
exports property ann:choose_trav_package to ⟨http://www.peter_travel_pref.gr⟩.
P_O5 =
eq:id(?x,?x) ← true.

ann:choose_trav_package(?package,?country) ← ¬ rdf:type(?country,geo:Europ_Country),
  (vac:travel(?package,?country), vac:visit(?package,?city))@⟨http://www.pyramis.gr⟩,
  ∀ ?city′ vac:visit(?package,?city′)@⟨http://www.pyramis.gr⟩ ⊃ eq:id(?city,?city′).

ann:choose_trav_package(?package,?country) ← rdf:type(?country,geo:CountryEU),
  (vac:travel(?package,?country), vac:visit(?package,?city),
   vac:visit(?package,?city′))@⟨http://www.travel_plan.gr⟩, ∼ eq:id(?city,?city′).

ann:choose_trav_package(?package,?country) ← rdf:type(?country,geo:Europ_Country),
 ¬ rdf:type(?country,geo:CountryEU),
   (vac:travel(?package,?country), vac:visit(?package,?city))@⟨http://www.travel_plan.gr⟩,
   geo:capital(?city,?country).
```

Fig. 1. A modular ERDF ontology

the list of countries). This local CWA is expressed by the single rule in P_{O_1}. Ontology O_2, with Nam_{O_2} =`<http://europa.eu>`, defines the list of European Union countries (which does not include `Croatia`) and states that this list is open (w.r.t. the resources of O_2)[14] by declaring the class `eu:CountryEU` as total. This local OWA is expressed by the first ERDF triple in G_{O_2}. Ontology O_3, with Nam_{O_3} =`<http://www.pyramis.gr>`, provides information regarding the package tours of the greek travel agency *Pyramis*. Similarly, ontology O_4, with Nam_{O_4} =`<http://www.travel_plan.gr>`, provides information regarding the package tours of the greek travel agency *Travel Plan*.

Finally, ontology O_5, with Nam_{O_5} =`<http://www.anne_travel_pref.gr>`, presents the travel preferences of Anne. Specifically, Anne prefers either (i) a trip to a non-European country by *Pyramis* that visits only one city, or (ii) a trip to an EU country by *Travel Plan* that visits at least one city, or (iii) a trip to a European but not EU country by *Travel Plan* that visits the capital of the country. Note that O_5 imports the properties `vac:travel` and `vac:visit` from any providing `r`-ERDF ontology in $\mathcal{R}$ (that is, O_3 and O_4). Additionally, note that `r`-ERDF ontology O_5 exports property `ann:choose_trav_package` to an `r`-ERDF ontology, named `<http://www.peter_travel_pref.gr>`, not in $\mathcal{R}$. □

Let $\mathcal{R}$ be a modular ERDF ontology, let $O \in \mathcal{R}$, and let $x \in \mathit{Exported}^{\mathtt{t}}_{O}$, for $\mathtt{t} \in \{\mathtt{pr}, \mathtt{cl}\}$. We define:

$$\mathit{Export}^{\mathtt{t}}_{O,\mathcal{R}}(x) = \begin{cases} \{\mathit{Nam}_{O'} \mid O' \in \mathcal{R} - \{O\}\} & \text{if } \mathit{Export}^{\mathtt{t}}_{O}(x) = \{*\} \\ \mathit{Export}^{\mathtt{t}}_{O}(x) \cap \{\mathit{Nam}_{O'} \mid O' \in \mathcal{R}\} & \text{otherwise} \end{cases}$$

Intuitively, $\mathit{Export}^{\mathtt{pr}}_{O,\mathcal{R}}(x)$ (resp. $\mathit{Export}^{\mathtt{cl}}_{O,\mathcal{R}}(x)$) denotes the `r`-ERDF ontologies in $\mathcal{R}$ to which O is willing to export property (resp. class) x.

Example 2. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. Then, it holds $\mathit{Export}^{\mathtt{cl}}_{O_2,\mathcal{R}}(\mathtt{eu{:}CountryEU}) = \{O_1, O_3, O_4, O_5\}$, because $\mathit{Export}^{\mathtt{cl}}_{O_2}(\mathtt{eu{:}CountryEU}) = \{*\}$. Additionally, it holds $\mathit{Export}^{\mathtt{pr}}_{O_5,\mathcal{R}}(\mathtt{ann{:}choose\_trav\_package})=\{\}$. □

Let $\mathcal{R}$ be a modular ERDF ontology, let $O \in \mathcal{R}$, and let $x \in \mathit{Imported}^{\mathtt{t}}_{O}$, for $\mathtt{t} \in \{\mathtt{pr}, \mathtt{cl}\}$. We define:

$$\mathit{Import}^{\mathtt{t}}_{O,\mathcal{R}}(x) = \begin{cases} \mathit{ExportingTo}^{\mathtt{t}}_{\mathcal{R}}(x, O) & \text{if } \mathit{Import}^{\mathtt{t}}_{O}(x) = \{*\} \\ \mathit{Import}^{\mathtt{t}}_{O}(x) \cap \mathit{ExportingTo}^{\mathtt{t}}_{\mathcal{R}}(x, O) & \text{otherwise,} \end{cases}$$

where $\mathit{ExportingTo}^{\mathtt{t}}_{\mathcal{R}}(x, O) = \{\mathit{Nam}_{O'} \mid O' \in \mathcal{R},\ x \in \mathit{Exported}^{\mathtt{t}}_{O'}, \text{ and } \mathit{Nam}_O \in \mathit{Export}^{\mathtt{t}}_{O',\mathcal{R}}(x)\}$.

Intuitively, $\mathit{ExportingTo}^{\mathtt{pr}}_{\mathcal{R}}(x, O)$ (resp. $\mathit{ExportingTo}^{\mathtt{cl}}_{\mathcal{R}}(x, O)$) denotes the `r`-ERDF ontologies in $\mathcal{R}$ that are willing to export property (resp. class) x to O. Additionally, $\mathit{Import}^{\mathtt{pr}}_{O,\mathcal{R}}(x)$ (resp. $\mathit{Import}^{\mathtt{cl}}_{O,\mathcal{R}}(x)$) denotes the `r`-ERDF ontologies in $\mathcal{R}$ from which O imports property (resp. class) x.

Example 3. For the modular ERDF ontology $\mathcal{R}$ of Example 1, $\mathit{ExportingTo}^{\mathtt{pr}}_{\mathcal{R}}(\mathtt{vac{:}travel}, O_5) = \{O_3, O_4\}$. Additionally, $\mathit{Import}^{\mathtt{pr}}_{O_5,\mathcal{R}}(\mathtt{vac{:}travel}) = \{O_3, O_4\}$. □

The multiple definition of *import* and *export* functions is due to the facts: (i) a *g*-RDF ontology may implicitly export a property or class to any other *g*-RDF ontology, (ii) a *g*-RDF ontology may explicitly export a property or class to *g*-RDF ontologies not in $\mathcal{R}$, (iii) a *g*-RDF ontology may implicitly import a property or class from any providing *g*-RDF ontology, and (iv) a *g*-RDF ontology O may explicitly import a property or class from *g*-RDF ontologies not in $\mathcal{R}$ or not willing to export the desired information to O.

In order for a modular rule base to be *valid*, it has to satisfy a number of validity constraints.

[14] Note that ontology O_2 imports class `geo:Country` from ontology O_1.

Definition 5 (Valid modular ERDF ontology). A modular ERDF ontology $\mathcal{R}$ is *valid* iff:

1. If $O, O' \in \mathcal{R}$ and $O \neq O'$ then $Nam_O \neq Nam_{O'}$.
2. If $O \in \mathcal{R}$ and $x \in Imported^{\mathtt{t}}_O$, for $\mathtt{t} \in \{\mathtt{pr}, \mathtt{cl}\}$, then $Import^{\mathtt{t}}_O(x) = \{*\}$ or $Import^{\mathtt{t}}_O(x) \subseteq ExportingTo^{\mathtt{t}}_{\mathcal{R}}(x, O)$.
3. If $O \in \mathcal{R}$ and $r \in P_O$ such that a qualified ERDF formula $F@Nam_{O'}$ appears in $Cond(r)$ then: (i) $O' \in \mathcal{R}$, (ii) for each $p(s, o)$, where $p \neq \mathtt{rdf:type}$, appearing in F, it holds that $O \in Export^{\mathtt{pr}}_{O',\mathcal{R}}(p)$, and (iii) for each $\mathtt{rdf:type}(x, c)$, appearing in F, it holds that (a) $O \in Export^{\mathtt{pr}}_{O',\mathcal{R}}(\mathtt{rdf:type})$ or (b) $c \in URI$ and $O \in Export^{\mathtt{cl}}_{O',\mathcal{R}}(c)$. □

Let $\mathcal{R}$ be a valid modular ERDF ontology. Constraint (1) of Definition 5 expresses that different r-ERDF ontologies in $\mathcal{R}$ should have different names in order to be uniquely identified. Let $O \in \mathcal{R}$. Constraint (2) expresses that if O requests a property or class x *explicitly* from an r-ERDF ontology O' then it should hold that $O' \in \mathcal{R}$ and O' is willing to export x to O. Assume now that it exists $r \in P_O$ s.t. $Cond(r)$ refers to an ERDF formula F of an r-ERDF ontology O'. Constraint (3.i) expresses that it should hold that $O' \in \mathcal{R}$. Constraint (3.ii) expresses that if r refers to $p(s, o)$ of O', where $p \neq \mathtt{rdf:type}$, then O' should be willing to export property p to O. Additionally, constraint (3.iii) expresses that if O refers to $\mathtt{rdf:type}(x, c)$ of O' then O' should be willing to either (a) export to O the property rdf:type, expressing that all classes of O' are exported to O, or (b) export to O just the class c (if $c \in URI$).

Example 4. Modular rule base $\mathcal{R}$ of Example 1 is valid. □

Convention: In this work, we consider valid modular ERDF ontologies, only. Additionally, for simplification reasons, by the unqualified term "ontology", we refer to an "r-ERDF ontology".

4 Modular ERDF and Herbrand Interpretations

In this section, we define the modular ERDF interpretations of an ontology w.r.t. a modular ERDF ontology. Additionally, we define satisfaction of an r-ERDF formula by such a modular ERDF interpretation and an ontology. Further, we define the modular Herbrand interpretations of an ontology w.r.t. a modular ERDF ontology.

Below we review the definition of a partial interpretation of a vocabulary V [6], which is an extension of the definition of a simple interpretation of V [42], such that each property is associated not only with a truth extension but also with a falsity extension, allowing for partial properties.

Definition 6 (Partial interpretation of a vocabulary). A *partial interpretation* I of a vocabulary V_I consists of:

- A non-empty set of resources Res_I, called the *domain* or *universe* of I.
- A set of properties $Prop_I$.
- A vocabulary interpretation mapping $I_{\mathtt{V}} : V_I \cap URI \rightarrow Res_I \cup Prop_I$.
- A property-truth extension mapping $PT_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I)$.
- A property-falsity extension mapping $PF_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I)$.
- A mapping $IL_I : V_I \cap \mathcal{TL} \rightarrow Res_I$.
- A set of literal values $LV_I \subseteq Res_I$, which contains $V \cap \mathcal{PL}$.

We define the mapping: $I : V_I \rightarrow Res_I \cup Prop_I$ such that: (i) $I(x) = I_{\mathtt{V}}(x),\ \forall x \in V_I \cap URI$, (ii) $I(x) = x,\ \forall\, x \in V_I \cap \mathcal{PL}$, and (iii) $I(x) = IL_I(x),\ \forall\, x \in V_I \cap \mathcal{TL}$. □

A partial interpretation I is *coherent* iff for all $x \in Prop_I$, $PT_I(x) \cap PF_I(x) = \emptyset$.

Convention: In the rest of the paper by $\mathcal{R}$, we will denote a modular ERDF ontology (MEO).

Let $O \in \mathcal{R}$. Below we define the dependencies of O w.r.t. $\mathcal{R}$. These are the ontologies in $\mathcal{R}$ on which O depends, either because O imports directly or indirectly information from them or because it queries directly or indirectly them.

Definition 7 (Dependencies of an ontology w.r.t. a MEO). Let $O \in \mathcal{R}$. The *dependencies* of O w.r.t. $\mathcal{R}$, denoted by $D_O^{\mathcal{R}}$, is the minimum set of ontologies s.t.: (i) $O \in D_O^{\mathcal{R}}$, (ii) if $O' \in D_O^{\mathcal{R}}$ and it exists $x \in Imported^{\mathtt{t}}_{O'}$, for $\mathtt{t} \in \{\mathtt{pr}, \mathtt{cl}\}$, s.t. $Nam_{O''} \in Import^{\mathtt{t}}_{O',\mathcal{R}}(x)$ then $O'' \in D_O^{\mathcal{R}}$, and (iii) if $O' \in D_O^{\mathcal{R}}$, $r \in P_{O'}$, and it exists a qualified ERDF formula $F@Nam_{O''}$ in $Cond(r)$ then $O'' \in D_O^{\mathcal{R}}$. □

Example 5. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. It holds: (i) $D_{O_1}^{\mathcal{R}} = \{O_1\}$, (ii) $D_{O_2}^{\mathcal{R}} = \{O_2, O_1\}$, and (iii) $D_{O_5}^{\mathcal{R}} = \{O_5, O_1, O_2, O_3, O_4\}$. Item (i) is true because ontology O_1 does not import any property or class from any other ontology, item (ii) is true because ontology O_2 imports class `geo:Country` from ontology O_1, and finally item (iii) is true because ontology O_5 imports class `geo:Europ_Country` and property `geo:capital` from ontology O_1, class `eu:CountryEU` from ontology O_2, and properties `vac:visit` and `vac:travel` from ontologies O_3 and O_4. Additionally, ontology O_2 uses ontologies O_3 and O_4 in the body of its rules. □

The vocabulary of RDF, $\mathcal{V}_{RDF}$, is a set of *URI* references in the *rdf*: namespace [42], and the vocabulary of RDFS, $\mathcal{V}_{RDFS}$, is a set of *URI* references in the *rdfs*: namespace [42]. Let $n \in I\!N$. We define $\mathcal{V}_{RDF}^{\#n} = \mathcal{V}_{RDF} - \{rdf{:}\_i \mid i > n\}$. The *vocabulary of ERDF* is defined as $\mathcal{V}_{ERDF} = \{erdf{:}TotalClass,\ erdf{:}TotalProperty\}$.

Let $O \in \mathcal{R}$. We define: (i) $N_O = 0$, if $(V_{G_O} \cup V_{P_O}) \cap \{rdf{:}\_i \mid i \geq 1\} = \emptyset$, and (ii) $N_O = max(\{i \in I\!N \mid rdf{:}\_i \in V_{G_O} \cup V_{P_O}\})$, otherwise. Further, we define: $n_O = max(\{N_{O'} \mid O' \in D_O^{\mathcal{R}}\} \cup \{1\})$. Intuitively, n_O is the largest i ($i \in I\!N$) such that $rdf{:}\_i$ appears in an $O' \in D_O^{\mathcal{R}}$. In the case that no such an $rdf{:}\_i$ exists then $n_O = 1$. Recall that the $rdf{:}\_i$ properties are used in RDF(S) [42] to express members of containers (i.e. bags, sequences, and alternatives), which are in practice finitely limited.

Let $O \in \mathcal{R}$, and let $n \in I\!N$. The *#n-vocabulary* of O is defined as: $V_O^{\#n} = V_{sk(G_O)} \cup V_{P_O} \cup Exported_O^{\mathtt{pr}} \cup Exported_O^{\mathtt{cl}} \cup Imported_O^{\mathtt{pr}} \cup Imported_O^{\mathtt{cl}} \cup V_{RDF}^{\#n} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$. Intuitively, if $L_O = \langle G, P \rangle$ then $V_O^{\#n}$ consists of the vocabulary of $sk(G)$ plus the vocabulary of P plus all classes or properties imported and exported by O plus the standard vocabularies $V_{RDF}^{\#n} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$. The *vocabulary* of O w.r.t. $\mathcal{R}$ is defined as: $V_{O,\mathcal{R}} = \cup\{V_{O'}^{\#n_O} \mid O' \in D_O^{\mathcal{R}}\}$. Intuitively, $V_{O,\mathcal{R}}$ corresponds to the local domain of O w.r.t. R.

Example 6. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. It holds that:
$V_{O_2,\mathcal{R}} = \{$`geo:Egypt,geo:Italy,geo:Croatia,geo:Cairo,geo:Zagreb,geo:Country, geo:Europ_Country,eu:CountryEU`$\} \cup V_{RDF}^{\#1} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$. □

Let $n \in I\!N$. We would like to define the modular ERDF interpretations of an ontology w.r.t. a modular ERDF ontology. Here, we use the definition of an *ERDF #n-interpretation* over a vocabulary V from [7], reviewed below for completeness. Intuitively, an *ERDF #n-interpretation* I of a vocabulary V is a partial interpretation of $V_I = V \cup \mathcal{V}_{RDF}^{\#n} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$ that assigns truth and falsity extensions to the classes and properties in V_I, satisfying: (i) all semantic conditions of an RDFS interpretation [42] of V, except these referring to $\{rdf{:}\_i \mid i > n\}$ terms, as well as (ii) new semantic conditions, particular to ERDF. Several examples and motivation of the semantic conditions of an ERDF $\#n$-interpretation can be found in [6].

Definition 8 (ERDF $\#n$-interpretation). An *ERDF $\#n$-interpretation* I of a vocabulary V is a coherent, partial interpretation of $V \cup \mathcal{V}_{RDF}^{\#n} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$, extended by the new ontological categories[15] $Cls_I \subseteq Res_I$ for classes, $TCls_I \subseteq Cls_I$ for total classes, and $TProp_I \subseteq Prop_I$ for total properties, as well as the class-truth extension mapping $CT_I : Cls_I \rightarrow \mathcal{P}(Res_I)$, and the class-falsity extension mapping $CF_I : Cls_I \rightarrow \mathcal{P}(Res_I)$, such that:

1. $x \in CT_I(y)$ iff $\langle x, y\rangle \in PT_I(I(\texttt{rdf:type}))$, and
   $x \in CF_I(y)$ iff $\langle x, y\rangle \in PF_I(I(\texttt{rdf:type}))$.
2. The ontological categories are defined as follows:
   $Prop_I = CT_I(I(\texttt{rdf:Property}))$ $\quad Cls_I = CT_I(I(\texttt{rdfs:Class}))$
   $Res_I = CT_I(I(\texttt{rdfs:Resource}))$ $\quad LV_I = CT_I(I(\texttt{rdfs:Literal}))$
   $TCls_I = CT_I(I(\texttt{erdf:TotalClass}))$ $\quad TProp_I = CT_I(I(\texttt{erdf:TotalProperty}))$.
3. If $\langle x, y\rangle \in PT_I(I(\texttt{rdfs:domain})$ and $\langle z, w\rangle \in PT_I(x)$ then $z \in CT_I(y)$.
4. If $\langle x, y\rangle \in PT_I(I(\texttt{rdfs:range}))$ and $\langle z, w\rangle \in PT_I(x)$ then $w \in CT_I(y)$.
5. If $x \in Cls_I$ then $\langle x, I(\texttt{rdfs:Resource})\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
6. If $\langle x, y\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$ then
   $x, y \in Cls_I$, $CT_I(x) \subseteq CT_I(y)$, and $CF_I(y) \subseteq CF_I(x)$.
7. $PT_I(I(\texttt{rdfs:subClassOf}))$ is a reflexive and transitive relation on Cls_I.
8. If $\langle x, y\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$ then
   $x, y \in Prop_I$, $PT_I(x) \subseteq PT_I(y)$, and $PF_I(y) \subseteq PF_I(x)$.
9. $PT_I(I(\texttt{rdfs:subPropertyOf}))$ is a reflexive and transitive relation on $Prop_I$.
10. If $x \in CT_I(I(\texttt{rdfs:Datatype}))$ then
    $\langle x, I(\texttt{rdfs:Literal})\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
11. If $x \in CT_I(I(\texttt{rdfs:ContainerMembershipProperty}))$ then
    $\langle x, I(\texttt{rdfs:member})\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$.
12. If $x \in TCls_I$ then $CT_I(x) \cup CF_I(x) = Res_I$.
13. If $x \in TProp_I$ then $PT_I(x) \cup PF_I(x) = Res_I \times Res_I$.
14. If "s"^^`rdf:XMLLiteral` $\in V$ and s is a well-typed XML literal string, then
    IL_I("s"^^`rdf:XMLLiteral`) is the XML value of s, and
    IL_I("s"^^`rdf:XMLLiteral`) $\in CT_I(I(\texttt{rdf:XMLLiteral}))$.
15. If "s"^^`rdf:XMLLiteral` $\in V$ and s is an ill-typed XML literal string then
    IL_I("s"^^`rdf:XMLLiteral`) $\in Res_I - LV_I$, and
    IL_I("s"^^`rdf:XMLLiteral`) $\in CF_I(I(\texttt{rdfs:Literal}))$.
16. I satisfies the RDF and RDFS axiomatic triples [42], that contain *URI* references in $\mathcal{V}_{RDF}^{\#n} \cup \mathcal{V}_{RDFS}$.
17. I satisfies the following triples, called *ERDF axiomatic triples*:
    `rdfs:subClassOf(erdf:TotalClass, rdfs:Class).`
    `rdfs:subClassOf(erdf:TotalProperty, rdfs:Class).` $\square$

A modular ERDF interpretation is defined, below, as a set ERDF $\#n_{O'}$-interpretations of $V_{O',\mathcal{R}}$, where $O' \in D_O^{\mathcal{R}}$, that respects importing of classes and properties.

Definition 9 (Modular ERDF interpretation). Let $O \in \mathcal{R}$. A *modular ERDF interpretation* of O w.r.t. $\mathcal{R}$ is a set $\mathsf{I} = \{I_{O'} \mid O' \in D_O^{\mathcal{R}}\}$, where $I_{O'}$ is an ERDF $\#n_{O'}$-interpretation of $V_{O',\mathcal{R}}$ and it holds that:

[15] An *ontological category* is a set of ontological items, like Res_I, $Prop_I$, and Cls_I, and LV_I (as defined in RDFS semantics). The ontological categories $TCls$ and $TProp_I$ where introduced here for uniformity reasons.

1. If $O' \in D^{\mathcal{R}}_{O}$, $p \in \mathit{Imported}^{\mathtt{pr}}_{O',\mathcal{R}}$, and $\mathit{Nam}_{O''} \in \mathit{Import}^{\mathtt{pr}}_{O',\mathcal{R}}(p)$ then for all $x, y \in V_{O'',\mathcal{R}}$,
   (a) if $\langle I_{O''}(x), I_{O''}(y)\rangle \in PT_{I_{O''}}(I_{O''}(p))$ then $\langle I_{O'}(x), I_{O'}(y)\rangle \in PT_{I_{O'}}(I_{O'}(p))$ and
   (b) if $\langle I_{O''}(x), I_{O''}(y)\rangle \in PF_{I_{O''}}(I_{O''}(p))$ then $\langle I_{O'}(x), I_{O'}(y)\rangle \in PF_{I_{O'}}(I_{O'}(p))$.
2. If $O' \in D^{\mathcal{R}}_{O}$, $c \in \mathit{Imported}^{\mathtt{cl}}_{O',\mathcal{R}}$, and $\mathit{Nam}_{O''} \in \mathit{Import}^{\mathtt{cl}}_{O',\mathcal{R}}(c)$ then for all $o \in V_{O'',\mathcal{R}}$,
   (a) if $\langle I_{O''}(o), I_{O''}(c)\rangle \in PT_{I_{O''}}(I_{O''}(\mathtt{rdf{:}type}))$ then $\langle I_{O'}(x), I_{O'}(c)\rangle \in PT_{I_{O'}}(I_{O'}(\mathtt{rdf{:}type}))$ and
   (b) if $\langle I_{O''}(o), I_{O''}(c)\rangle \in PF_{I_{O''}}(I_{O''}(\mathtt{rdf{:}type}))$ then $\langle I_{O'}(x), I_{O'}(c)\rangle \in PF_{I_{O'}}(I_{O'}(\mathtt{rdf{:}type}))$. □

The motivation behind Definition 9 is that if an ontology O' imports a property p from another ontology O'' then (i) if $p(x, y)$ is true in O'' then $p(x, y)$ should be true in O' and (ii) if $p(x, y)$ is explicitly false in O'' then $p(x, y)$ should be explicitly false in O'. Additionally, if an ontology O' imports a class c from another ontology O'' then (i) if $\mathtt{rdf{:}type}(o, c)$ is true in O'' then $\mathtt{rdf{:}type}(o, c)$ should be true in O' and (ii) if $\mathtt{rdf{:}type}(o, c)$ is explicitly false in O'' then $\mathtt{rdf{:}type}(o, c)$ should be explicitly false in O'.

As special cases, if an ontology O' imports property `rdf:type` from another ontology O'' then if $\mathtt{rdf{:}type}(o, c)$ is true in O'' then $\mathtt{rdf{:}type}(o, c)$ is also true in O'. If O' imports class `rdfs:Class` from O'' then if $\mathtt{rdf{:}type}(c, \mathtt{rdfs{:}Class})$ is true in O'' then $\mathtt{rdf{:}type}(c, \mathtt{rdfs{:}Class})$ is also true in O'. If O' imports class `rdf:Property` from O'' then if $\mathtt{rdf{:}type}(p, \mathtt{rdf{:}Property})$ is true in O'' then $\mathtt{rdf{:}type}(p, \mathtt{rdf{:}Property})$ is also true in O'.

Example 7. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. Let $\mathsf{I} = \{I_{O_1}, I_{O_2}, I_{O_3}, I_{O_4}, I_{O_5}\}$ be a modular ERDF interpretation of O_5 w.r.t. R. Since $O_2 \in D^{\mathcal{R}}_{O_5}$, $\mathtt{geo{:}Country} \in \mathit{Imported}^{\mathtt{cl}}_{O_2,\mathcal{R}}$ and $\langle \mathit{http://geography.int}\rangle \in \mathit{Import}^{\mathtt{cl}}_{O_2,\mathcal{R}}(\mathtt{geo{:}Country})$, it holds that: for all $o \in V_{O_1,\mathcal{R}}$, if $\langle I_{O_1}(o), I_{O_1}(\mathtt{geo{:}County})\rangle \in PT_{I_{O_1}}(I_{O_1}(\mathtt{rdf{:}type}))$ then $\langle I_{O_2}(o), I_{O_2}(\mathtt{geo{:}County})\rangle \in PT_{I_{O_2}}(I_{O_2}(\mathtt{rdf{:}type}))$ and if $\langle I_{O_1}(o), I_{O_1}(\mathtt{geo{:}County})\rangle \in PF_{I_{O_1}}(I_{O_1}(\mathtt{rdf{:}type}))$ then $\langle I_{O_2}(o), I_{O_2}(\mathtt{geo{:}County})\rangle \in PF_{I_{O_2}}(I_{O_2}(\mathtt{rdf{:}type}))$. □

Below, we define satisfaction of an `r`-ERDF formula w.r.t. a modular ERDF interpretation, an ontology, and a valuation. First, we provide an auxiliary definition. Let I be a partial interpretation of a vocabulary V_I, let Res be a set, and let v be a partial function $v : \mathit{Var} \rightarrow \mathit{Res}$ (called *valuation*). If $x \in \mathit{Var}$, we define $[I + v](x) = v(x)$. If $x \in V_I$, we define $[I + v](x) = I(x)$.

Definition 10. (Satisfaction of an r-ERDF formula w.r.t. a modular ERDF interpretation, an ontogoly, and a valuation) Let $O \in \mathcal{R}$ and let Res be a set. Let $\mathsf{I} = \{I_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. Additionally, let F be an `r`-ERDF formula over $\{\mathit{Nam}_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$. For each $O', O'' \in D^{\mathcal{R}}_{O}$ and for each mapping $v : \mathit{Var}(F) \rightarrow \mathit{Res}$:

– If $F = p(s, o)$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $p \in V_{I_{O'}} \cap \mathit{URI}$, $s, o \in V_{I_{O'}} \cup \mathit{Var}$, $I_{O'}(p) \in \mathit{Prop}_{I_{O'}}$, and $\langle [I_{O'} + v](s), [I_{O'} + v](o)\rangle \in PT_{I_{O'}}(I_{O'}(p))$.
– If $F = \neg p(s, o)$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $p \in V_{I_{O'}} \cap \mathit{URI}$, $s, o \in V_{I_{O'}} \cup \mathit{Var}$, $I_{O'}(p) \in \mathit{Prop}_{I_{O'}}$, and $\langle [I_{O'} + v](s), [I_{O'} + v](o)\rangle \in PF_{I_{O'}}(I_{O'}(p))$.
– If $F = {\sim}G$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $\langle \mathsf{I}, O', v\rangle \not\models G$.
– If $F = F_1 \wedge F_2$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $\langle \mathsf{I}, O', v\rangle \models F_1$ and $\langle \mathsf{I}, O', v\rangle \models F_2$.
– If $F = F_1 \vee F_2$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $\langle \mathsf{I}, O', v\rangle \models F_1$ or $\langle I, O', v\rangle \models F_2$.
– If $F = F_1 \supset F_2$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $\langle \mathsf{I}, O', v\rangle \models {\sim}F_1 \vee F_2$.
– If $F = \exists x\ G$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff there exists a mapping u such that $u(y) = v(y)$, $\forall y \in \mathit{Var}(G) - \{x\}$, $u(x) \in \mathit{Res}_{I_{O'}}$, and $\langle \mathsf{I}, O', u\rangle \models G$.
– If $F = \forall x\ G$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff for all mappings u such that $u(y) = v(y)$, $\forall y \in \mathit{Var}(G) - \{x\}$ and $u(x) \in \mathit{Res}_{I_{O'}}$, it holds that $\langle \mathsf{I}, O', u\rangle \models G$.

– If $F = G@Nam_{O''}$ then $\langle \mathsf{I}, O', v\rangle \models F$ iff $\langle \mathsf{I}, O'', v\rangle \models G$.
– All other cases of ERDF formulas are treated by the following DeMorgan-style rewrite rules expressing the falsification of compound ERDF formulas:
$\neg(F_1 \wedge F_2) \to \neg F_1 \vee \neg F_2$, $\neg(F_1 \vee F_2) \to \neg F_1 \wedge \neg F_2$, $\neg(\neg G) \to G$, $\neg({\sim} G) \to G$[16],
$\neg(\exists x\ G) \to \forall x\ \neg G$, $\neg(\forall x\ G) \to \exists x\ \neg G$, $\neg(F_1 \supset F_2) \to F_1 \wedge \neg F_2$,
$\neg(G@Nam_{O'}) \to (\neg G)@Nam_{O'}$. □

Note that the last item of the previous definition is exactly the same as that of the corresponding definition (Definition 3.4) in [6]. Let $O \in \mathcal{R}$, and let $O' \in D^{\mathcal{R}}_O$. Let I be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$ and let F be an r-ERDF formula. We define: $\langle \mathsf{I}, O'\rangle \models F$ iff for each mapping $v : Var(F) \to Res_{I_{O'}}$, it holds that $\langle \mathsf{I}, O', v\rangle \models F$. Additionally, let G be an ERDF graph. We define: $\langle \mathsf{I}, O'\rangle \models G$ iff $\langle \mathsf{I}, O'\rangle \models formula(G)$. Note that if $formula(G) = \exists x_1, ..., \exists x_k\ t_1 \wedge ... \wedge t_m$ then, based on Definition 10, $\langle \mathsf{I}, O'\rangle \models formula(G)$ iff there exists mapping $v : Var(F) \to Res_{I_{O'}}$ s.t. $\langle \mathsf{I}, O', v\rangle \models t_1 \wedge ... \wedge t_m$.

We assume that for every function $v : Var \to Res_{I_{O'}}$, it holds that $\langle \mathsf{I}, O', v\rangle \models \mathtt{true}$ and $\langle \mathsf{I}, O', v\rangle \not\models \mathtt{false}$.

Note that Definition 10 does not impose any restriction on $Res_{I_{O'}}$ and $Res_{I_{O''}}$. For example, let $F = p(?x, o) \wedge p'(?x, o')@Nam_{O''}$, where $p, o, p', o' \in URI$, $I_{O'}(p) \in Prop_{I_{O'}}$, and $I_{O''}(p') \in Prop_{I_{O''}}$. Let $v : \{?x\} \to Res_{I_{O'}} \cap Res_{I_{O''}}$ s.t. $\langle v(?x), I_{O'}(o)\rangle \in PT_{I_{O'}}(I_{O'}(p))$ and $\langle v(?x), I_{O''}(o')\rangle \in PT_{I_{O''}}(I_{O''}(p'))$. Then, $\langle \mathsf{I}, O', v\rangle \models F$. Now, let $u : \{?x\} \to Res_{I_{O'}} - Res_{I_{O''}}$ s.t. $\langle v(?x), I_{O'}(o)\rangle \in PT_{I_{O'}}(I_{O'}(p))$. Then, $\langle \mathsf{I}, O', v\rangle \models {\sim}F$, as $\langle \mathsf{I}, O', v\rangle \not\models F$. The latter is true because $u(?x) \notin Res_{I_{O''}}$. Thus, $\langle \mathsf{I}, O'', v\rangle \not\models F$. Note that in Definition 10 satisfaction of a formula $F = G@Nam_{O''}$ according to an r-RDF ontotology O' is translated to satisfaction of F according O''.

The following two propositions state equivalences on the satisfaction of qualified ERDF formulas.

Proposition 1. Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} |\ O' \in D^{\mathcal{R}}_O\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. Let F_1, F_2 be ERDF formulas and $O'' \in D^{\mathcal{R}}_O$. Assuming that $O' \in D^{\mathcal{R}}_O$, it holds that:

1. $\langle \mathsf{I}, O'\rangle \models (F_1 \wedge F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O'\rangle \models F_1@Nam_{O''} \wedge F_2@Nam_{O''}$.
2. $\langle \mathsf{I}, O'\rangle \models (F_1 \vee F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O'\rangle \models F_1@Nam_{O''} \vee F_2@Nam_{O''}$.

Proposition 2. Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} |\ O' \in D^{\mathcal{R}}_O\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. Additionally, let $O', O'' \in D^{\mathcal{R}}_O$. It holds that: $\langle \mathsf{I}, O'\rangle \models ({\sim}p(s, o))@Nam_{O''}$ iff $\langle \mathsf{I}, O'\rangle \models {\sim}p(s, o)@Nam_{O''}$.

Below, we define the modular models of an ontology w.r.t. a modular ERDF ontology.

Definition 11 (Modular ERDF model). Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} |\ O' \in D^{\mathcal{R}}_O\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$ and let $O' \in D^{\mathcal{R}}_O$:

– We say that $\langle \mathsf{I}, O'\rangle$ *satisfies* an r-ERDF rule r, denoted by $\langle \mathsf{I}, O'\rangle \models r$, iff it holds: For all mappings $v : Var(r) \to Res_{I_{O'}}$, if $\langle \mathsf{I}, O', v\rangle \models Cond(r)$ then $\langle \mathsf{I}, O', v\rangle \models Concl(r)$.
– We say that I is a *modular* (*ERDF*) *model* of O w.r.t. $\mathcal{R}$, denoted by $\mathsf{I} \models_{\mathcal{R}} O$, iff for all $O' \in D^{\mathcal{R}}_O$, $\langle \mathsf{I}, O'\rangle \models G_{O'}$ and $\langle \mathsf{I}, O'\rangle \models r$, for all $r \in P_{O'}$. □

Let $O \in \mathcal{R}$. We denote by $Res^{\mathtt{H}}_{O,\mathcal{R}}$ the union of $V_{O,\mathcal{R}}$ and the set of XML values of the well-typed XML literals in $V_{O,\mathcal{R}}$ minus the well-typed XML literals [17]. In other

[16] This transformation expresses that if it is *false* that G *does not hold* then G *holds*.
[17] Note that semantic condition 14 in Definition 8 maps well-typed XML literals to their XML values, similarly to RDFS semantics.

words, the set of Herbrand resources $Res^{\mathtt{H}}_{O,\mathcal{R}}$ is $V_{O,\mathcal{R}}$ with the well-typed XML literals substituted by their XML values. Below we define the modular Herbrand interpretations of O w.r.t. $\mathcal{R}$, extending the definition of a Herbrand interpretation of an ERDF ontology [6]. This definition is needed for defining the modular stable models of O w.r.t. R.

Definition 12 (Modular ERDF Herbrand interpretation). Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} | \ O' \in D^{\mathcal{R}}_{O}\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. We say that I is a *modular* (*ERDF*) *Herbrand interpretation* of O w.r.t. $\mathcal{R}$ iff for each $O' \in D^{\mathcal{R}}_{O}$:

- $Res_{I_{O'}} = Res^{\mathtt{H}}_{O',\mathcal{R}}$.
- $I_{O'_{\mathtt{V}}}(x) = x$, for all $x \in V_{O',\mathcal{R}} \cap URI$.
- $IL_{I_{O'}}(x) = x$, if[18] x is a typed literal in $V_{O',\mathcal{R}}$ other than a well-typed XML literal, and $IL_{I_{O'}}(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{O',\mathcal{R}}$.

We denote the set of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$ by $\mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. □

Note that different ontologies $O', O'' \in D^{\mathcal{R}}_{O}$ have different vocabularies but their common elements are mapped through $I_{O'}$ and $I_{O''}$, respectively, to the same resource. Additionally, note that we first defined the notion of a modular ERDF interpretation and then the notion of a modular ERDF Herbrand interpretation of an $O \in \mathcal{R}$, in order to faithfully extend the RDFS semantics [42]. As usual in the construction of Herbrand interpretations, we map every constant to itself except for the predefined XML literals that have a fixed interpretation in the original RDFS semantics. Accordingly, a modular ERDF Herbrand interpretation can be succinctly described by the property-truth and property-false extensions of the properties in $O' \in D^{\mathcal{R}}_{O}$.

Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} | \ O' \in D^{\mathcal{R}}_{O}\}$ be a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$. We say that I is a *modular* (*ERDF*) *Herbrand model* of O w.r.t. $\mathcal{R}$ iff for all $O' \in D^{\mathcal{R}}_{O}$, (i) $\langle \mathsf{I}, O' \rangle \models sk(G_{O'})$ and (ii) for all $r \in P_{O'}$, $\langle \mathsf{I}, O' \rangle \models r$. We denote the set of modular Herbrand models of O w.r.t. $\mathcal{R}$ by $\mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$.

Proposition 3. Let $O \in \mathcal{R}$. If M is a modular Herbrand model of O w.r.t. $\mathcal{R}$ then M is a modular model of O w.r.t. $\mathcal{R}$.

5 Modular ERDF Stable Model Semantics

In this Section, we define the modular stable models of an ontology w.r.t. a modular ERDF ontology, and provide some of their properties.

Let $\mathcal{R}$ be a modular ERDF ontology and let $O \in \mathcal{R}$. We proceed by defining a partial ordering on the modular Herbrand interpretations of O w.r.t. $\mathcal{R}$.

Definition 13 (Modular Herbrand interpretation ordering). Let $O \in \mathcal{R}$. Let $\mathsf{I}, \mathsf{J} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. We say that J *extends* I, denoted by $\mathsf{I} \leq \mathsf{J}$ (or $\mathsf{J} \geq \mathsf{I}$) iff: For all $O' \in D^{\mathcal{R}}_{O}$, it holds that (i) $Prop_{I_{O'}} \subseteq Prop_{J_{O'}}$, and for all $p \in Prop_{I_{O'}}$, it holds that $PT_{I_{O'}}(p) \subseteq PT_{J_{O'}}(p)$ and $PF_{I_{O'}}(p) \subseteq PF_{J_{O'}}(p)$. □

Let $O \in \mathcal{R}$. The intuition behind Definition 13 is that by extending a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$, we extend both the truth and falsity extension for all properties of $O' \in D^{\mathcal{R}}_{O}$, and thus (since *rdf:type* is a property), for all classes.

Let $O \in \mathcal{R}$. Let $\mathcal{I} \subseteq \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. We define $minimal(\mathcal{I}) = \{\mathsf{I} \in \mathcal{I} \mid$ there is no $\mathsf{J} \in \mathcal{I} : \ \mathsf{J} \neq \mathsf{I}$ and $\mathsf{J} \leq \mathsf{I}\}$. Let $\mathsf{I}, \mathsf{J} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. We define $[\mathsf{I}, \mathsf{J}]_{O,\mathcal{R}} = \{\mathsf{I}' \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}, \ \mathsf{I} \leq \mathsf{I}' \leq \mathsf{J}\}$.

[18] Note that this item is in accordance with semantic conditions 14 and 15 of Definition 8.

Let V be a vocabulary and let r be an r-ERDF rule. We denote by $[r]_V$ the set of rules that result from r if we replace each variable $x \in FVar(r)$ by $v(x)$, for all mappings $v : FVar(r) \to V$. Let P be an r-ERDF program. We define $[P]_V = \bigcup_{r \in P} [r]_V$.

In [7], we defined the #n-stable models of an ERDF ontology (for an $n \in \mathbb{N}$), based on the coherent stable models of partial logic [43] (which, on extended logic programs (ELPs), are equivalent [43] to Answer Sets [31]). Here, we extend this definition to modular stable models of an ontology w.r.t. a modular ERDF ontology. Note that in contrast to ELPs, we allow arbitrary formulas in the bodies of the rules.

Definition 14 (Modular ERDF stable model). Let $O \in \mathcal{R}$, and let $\mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. We say that M is a *modular* (*ERDF*) *stable model* of O w.r.t. $\mathcal{R}$ iff there is a chain of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_k$ such that $\mathsf{I}_{k-1} = \mathsf{I}_k = \mathsf{M}$ and:

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \ \forall O' \in D^{\mathcal{R}}_O, \ \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that:
   if $r \in [P_{O'}]_{V_{O',\mathcal{R}}}$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r), \ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$[19].

The set of modular stable models of O w.r.t. $\mathcal{R}$ is denoted by $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. $\forall O' \in D^{\mathcal{R}}_O$, $\langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$. □

Convention: In the sequel, to simplify notation, by $[P_{O'}]$ *we will denote* $[P_{O'}]_{V_{O',\mathcal{R}}}$.

Note that I_0 is a provenance Herbrand interpretation of O w.r.t. $\mathcal{R}$ that for all $O' \in D^{\mathcal{R}}_O$, $\langle \mathsf{I}, O' \rangle \models sk(G_{O'})$, while Herbrand interpretations $\mathsf{I}_1, ..., \mathsf{I}_{k+1}$ correspond to a stratified sequence of rule applications, where all applied rules remain applicable throughout the generation of a stable model M. In other words, a stable model is generated bottom-up by the iterative application of the rules in the programs $[P_{O'}]_{V_{O',\mathcal{R}}}$, where $O' \in D^{\mathcal{R}}_O$, starting from the information in I_0.

Example 8. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. For every $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}, O_1 \rangle \models \neg$ `rdf:type(geo:Egypt,geo:Europ_Country)`. This is due to the local CWA in P_{O_1}. Now, since O_5 imports class `geo:Europ_Country` from O_1, for every $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}, O_5 \rangle \models \neg$ `rdf:type(geo:Egypt,geo:Europ_Country)`. Therefore, due to the 2nd rule of P_{O_5}, for every $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, $\langle \mathsf{M}, O_5 \rangle \models$ `ann:choose_trav_package(pyr:package1,geo:Egypt)`.

Note the ERDF triple `rdf:type(eu:CountryEU,erdf:TotalClass)` in G_{O_2}, expressing a local OWA. Therefore, in some $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}, O_2 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`, while in the rest $\mathsf{M}' \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}', O_2 \rangle \models$ `rdf:type(geo:Croatia,eu:CountryEU)`. Now note that O_5 imports class `eu:CountryEU` from O_2. Thus, in some $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}, O_5 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`, while in the rest $\mathsf{M}' \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}', O_5 \rangle \models$ `rdf:type(geo:Croatia,eu:CountryEU)`. Reasoning now by cases and due to the 3rd and 4th rule of P_{O_5}, for every $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, it holds $\langle \mathsf{M}, O_5 \rangle \models$ `ann:choose_trav_package(trav:package2,geo:Croatia)`.

Let $\mathsf{M}_1 \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$ s.t. $\langle \mathsf{M}_1, O_5 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`. Then, M_1 is generated by a chain of modular Herbrand interpretations of O_5 w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq \mathsf{I}_1 \leq \mathsf{M}_1$ s.t.:

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O_5,\mathcal{R}} \mid \ \forall O' \in \{O_1, O_2, O_3, O_4, O_5\}, \ \langle \mathsf{I}, O' \rangle \models G_{O'}\})$.
2. $\mathsf{I}_1 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O_5,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_0$ and $\forall O' \in \{O_1, O_2, O_3, O_4, O_5\}$, it holds that:
   if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r), \ \forall \mathsf{J} \in [\mathsf{I}_0, \mathsf{M}_1]_{O_5,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

[19] This is somewhat similar to defining the stable models M of an ELP P through grounding of P using M (i.e. P/M).

3. $\mathsf{M}_1 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O_5,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_1$ and $\forall O' \in \{O_1, O_2, O_3, O_4, O_5\}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_1, \mathsf{M}_1]_{O_5,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.
4. $\mathsf{M}_1 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O_5,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{M}_1$ and $\forall O' \in \{O_1, O_2, O_3, O_4, O_5\}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{M}_1, \mathsf{M}_1]_{O_5,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

Modular Herbrand interpretation I_0

$\langle \mathsf{I}_0, O_1 \rangle \models p(s,o)$, for all $p(s,o) \in G_{O_1}$
$\langle \mathsf{I}_0, O_1 \rangle \models$ rdf:type(geo:Italy, geo:Country)
$\langle \mathsf{I}_0, O_1 \rangle \models$ rdf:type(geo:Croatia, geo:Country)
$\langle \mathsf{I}_0, O_2 \rangle \models p(s,o)$, for all $p(s,o) \in G_{O_2}$
$\langle \mathsf{I}_0, O_2 \rangle \models \neg$rdf:type(geo:Croatia, geo:CountryEU)
$\langle \mathsf{I}_0, O_2 \rangle \models$ rdf:type(geo:Egypt, geo:Country)
$\langle \mathsf{I}_0, O_2 \rangle \models$ rdf:type(geo:Italy, geo:Country)
$\langle \mathsf{I}_0, O_2 \rangle \models$ rdf:type(geo:Croatia, geo:Country)
$\langle \mathsf{I}_0, O_3 \rangle \models p(s,o)$, for all $p(s,o) \in G_{O_3}$
$\langle \mathsf{I}_0, O_4 \rangle \models p(s,o)$, for all $p(s,o) \in G_{O_4}$
$\langle \mathsf{I}_0, O_5 \rangle \models$ rdf:type(geo:Italy, geo:Europ_Country)
$\langle \mathsf{I}_0, O_5 \rangle \models$ rdf:type(geo:Croatia, geo:Europ_Country)
$\langle \mathsf{I}_0, O_5 \rangle \models$ geo:capital(geo:Cairo, geo:Egypt)
$\langle \mathsf{I}_0, O_5 \rangle \models$ geo:capital(geo:Zagreb, geo:Croatia)
$\langle \mathsf{I}_0, O_5 \rangle \models \neg$rdf:type(geo:Croatia, geo:CountryEU)
$\langle \mathsf{I}_0, O_5 \rangle \models p(s,o)$, for all $p(s,o) \in G_{O_3} \cup G_{O_4}$

Modular Herbrand interpretation I_1

$\langle \mathsf{I}_1, O_1 \rangle \models p(s,o)$, for all $\langle \mathsf{I}_0, O_0 \rangle \models p(s,o)$
$\langle \mathsf{I}_1, O_1 \rangle \models \neg$rdf:type(geo:Egypt, geo:Europ_Country)
$\langle \mathsf{I}_1, O_2 \rangle \models [\neg]p(s,o)$, for all $\langle \mathsf{I}_0, O_2 \rangle \models [\neg]p(s,o)$
$\langle \mathsf{I}_1, O_3 \rangle \models p(s,o)$, for all $\langle \mathsf{I}_0, O_3 \rangle \models p(s,o)$
$\langle \mathsf{I}_1, O_4 \rangle \models p(s,o)$, for all $\langle \mathsf{I}_0, O_4 \rangle \models p(s,o)$
$\langle \mathsf{I}_1, O_5 \rangle \models [\neg]p(s,o)$, for all $\langle \mathsf{I}_0, O_5 \rangle \models [\neg]p(s,o)$
$\langle \mathsf{I}_1, O_5 \rangle \models$ ann:choose_trav_package(trav:package2, geo:Croatia)
$\langle \mathsf{I}_1, O_5 \rangle \models \neg$rdf:type(geo:Egypt, geo:Europ_Country)

Modular Herbrand interpretation M_1

$\langle \mathsf{M}_1, O_1 \rangle \models [\neg]p(s,o)$, for all $\langle \mathsf{I}_1, O_1 \rangle \models [\neg]p(s,o)$
$\langle \mathsf{M}_1, O_2 \rangle \models [\neg]p(s,o)$, for all $\langle \mathsf{I}_1, O_2 \rangle \models [\neg]p(s,o)$
$\langle \mathsf{M}_1, O_3 \rangle \models p(s,o)$, for all $\langle \mathsf{I}_1, O_3 \rangle \models p(s,o)$
$\langle \mathsf{M}_1, O_4 \rangle \models p(s,o)$, for all $\langle \mathsf{I}_1, O_4 \rangle \models p(s,o)$
$\langle \mathsf{M}_1, O_5 \rangle \models [\neg]p(s,o)$, for all $\langle \mathsf{I}_1, O_5 \rangle \models [\neg]p(s,o)$
$\langle \mathsf{M}_0, O_5 \rangle \models$ ann:choose_trav_package(pyr:package1, geo:Egypt)

Fig. 2. ERDF triples satisfied by the modular Herbrand interpretations $\mathsf{I}_0, \mathsf{I}_1$, and M_1

In Figure 2, we show the ERDF triples satisfied by each modular Herbrand interpretation $\mathsf{I}_0, \mathsf{I}_1$, and M_1. Specifically, each $p(s,o) \in G_{O_1}$, it holds that $\langle \mathsf{I}_0, O_1 \rangle \models p(s,o)$. Additionally, since $\langle \mathsf{I}_0, O_1 \rangle \models$ `rdfs:subclass(geo:Europ_Country,geo:Country)`, it holds that $\langle \mathsf{I}_0, O_1 \rangle \models$ `rdf:type( geo:Italy,geo:Country)` and $\langle \mathsf{I}_0, O_1 \rangle \models$ `rdf:type(geo: Croatia,geo:Country)`. For each $p(s,o) \in G_{O_2}$, it holds that $\langle \mathsf{I}_0, O_2 \rangle \models p(s,o)$. Since $\langle \mathsf{I}_0, O_2 \rangle \models$ `rdf:type(eu:CountryEU,erdf:TotalClass)`, O_5 imports class `eu:CountryEU` from O_2, and $\langle \mathsf{M}_1, O_5 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`, it follows that $\langle \mathsf{I}_0, O_2 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`. Since O_2 imports class `geo:Country` from O_1, it holds that $\langle \mathsf{I}_0, O_2 \rangle \models$ `rdf:type(geo:Egypt,geo:Country)`, $\langle \mathsf{I}_0, O_2 \rangle \models$ `rdf: type(geo:Italy,geo:Country)`, and $\langle \mathsf{I}_0, O_2 \rangle \models$ `rdf:type(geo:Croatia,geo:Country)`.

For each $p(s,o) \in G_{O_3}$, it holds that $\langle \mathsf{I}_0, O_3 \rangle \models p(s,o)$ and for each $p(s,o) \in G_{O_4}$, it holds that $\langle \mathsf{I}_0, O_4 \rangle \models p(s,o)$.

Since O_5 imports class `geo:Europ_Country` from O_1, it holds that $\langle \mathsf{I}_0, O_5 \rangle \models$ `rdf:type(geo:Italy,geo:Europ_Country)` and $\langle \mathsf{I}_0, O_5 \rangle \models$ `rdf:type(geo:Croatia, geo:Europ_Country)`. Since O_5 imports property `geo:capital` from O_1, it holds that $\langle \mathsf{I}_0, O_5 \rangle \models$ `geo:capital(geo:Cairo,geo:Egypt)` and $\langle \mathsf{I}_0, O_5 \rangle \models$ `geo:capital(geo:Zagreb,geo:Croatia)`. Since O_5 imports class `eu:CountryEU` from O_2, it holds that $\langle \mathsf{I}_0, O_5 \rangle \models \neg$ `rdf:type(geo:Croatia,eu:CountryEU)`. Since O_5 imports properties `vac:travel` and `vac:visit` from O_3 and O_4, it follows that for each $p(s,o) \in G_{O_3} \cup G_{O_4}$, it holds that $\langle \mathsf{I}_0, O_5 \rangle \models p(s,o)$.

For each $p(s,o)$ s.t. $\langle \mathsf{I}_0, O_1 \rangle \models p(s,o)$, it holds that $\langle \mathsf{I}_1, O_1 \rangle \models p(s,o)$. Additionally, due to the single rule in P_{O_1}, it holds that $\langle \mathsf{I}_1, O_1 \rangle \models \neg$ `rdf:type(geo:Egypt,geo:Europ_Country)`. For each $[\neg]p(s,o)$ s.t. $\langle \mathsf{I}_0, O_2 \rangle \models [\neg]p(s,o)$, it holds that $\langle \mathsf{I}_1, O_2 \rangle \models [\neg]p(s,o)$. For each $p(s,o)$ s.t. $\langle \mathsf{I}_0, O_3 \rangle \models p(s,o)$, it holds that $\langle \mathsf{I}_1, O_3 \rangle \models p(s,o)$. For each $p(s,o)$ s.t. $\langle \mathsf{I}_0, O_4 \rangle \models p(s,o)$, it holds that $\langle \mathsf{I}_1, O_4 \rangle \models p(s,o)$. For each $p(s,o)$ s.t. $\langle \mathsf{I}_0, O_5 \rangle \models [\neg]p(s,o)$, it holds that $\langle \mathsf{I}_1, O_5 \rangle \models [\neg]p(s,o)$. Due to the fourth rule in P_{O_5}, it holds that $\langle \mathsf{I}_1, O_5 \rangle \models$ `ann:choose_trav_package(trav:package2,geo:Croatia)`, Additionally, since O_5 imports class `geo:Europ_Country` from O_1, it holds that $\langle \mathsf{I}_1, O_5 \rangle \models \neg$ `rdf:type(geo:Egypt,geo:Europ_Country)`.

For each $[\neg]p(s,o)$ s.t. $\langle \mathsf{I}_1, O_1 \rangle \models [\neg]p(s,o)$, it holds that $\langle \mathsf{M}_1, O_1 \rangle \models [\neg]p(s,o)$. For each $[\neg]p(s,o)$ s.t. $\langle \mathsf{I}_1, O_2 \rangle \models [\neg]p(s,o)$, it holds that $\langle \mathsf{M}_1, O_2 \rangle \models [\neg]p(s,o)$. For each $p(s,o)$ s.t. $\langle \mathsf{I}_1, O_3 \rangle \models p(s,o)$, it holds that $\langle \mathsf{M}_1, O_3 \rangle \models p(s,o)$. For each $p(s,o)$ s.t. $\langle \mathsf{I}_1, O_4 \rangle \models p(s,o)$, it holds that $\langle \mathsf{M}_1, O_4 \rangle \models p(s,o)$. For each $[\neg]p(s,o)$ s.t. $\langle \mathsf{I}_1, O_5 \rangle \models [\neg]p(s,o)$, it holds that $\langle \mathsf{M}_1, O_5 \rangle \models [\neg]p(s,o)$. Therefore, $\langle \mathsf{M}_1, O_5 \rangle \models$ `ann:choose_trav_package(trav:package2,geo:Croatia)`. Additionally, due to the second rule in P_{O_5}, it holds that $\langle \mathsf{M}_1, O_5 \rangle \models$ `ann:choose_trav_package(pyr:package1,geo:Egypt)`.

Let $O \in \mathcal{R}$. The following proposition shows that any modular stable model of O w.r.t. $\mathcal{R}$ is a modular Herbrand model of O w.r.t. $\mathcal{R}$.

Proposition 4. Let $O \in \mathcal{R}$. If $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ then $\mathsf{M} \in \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$.

On the other hand, if all properties of $O' \in D^{\mathcal{R}}_{O}$ are total, a modular Herbrand model M of O w.r.t. $\mathcal{R}$ is a modular stable model of O w.r.t. $\mathcal{R}$.

Proposition 5. Let $O \in \mathcal{R}$. If `rdfs:subclass(rdf:Property, erdf:TotalProperty)` $\in G_{O'}$, for all $O' \in D^{\mathcal{R}}_{O}$, then $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}} = \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$.

The following proposition relates the modular stable models of different ontologies w.r.t. a modular ERDF ontology.

Proposition 6. Let $O \in \mathcal{R}$, and let $O' \in D^{\mathcal{R}}_{O}$. Let $\mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$ and let $\mathsf{M}' = \{M_{O''} \in \mathsf{M} \mid O'' \in D^{\mathcal{R}}_{O'}\}$[20]. It holds that: If $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$.

Let $O \in \mathcal{R}$. We say that O is *inconsistent* under the modular stable model semantics w.r.t. $\mathcal{R}$ iff $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}} = \{\}$.

Let $O \in \mathcal{R}$, and let $O' \in D^{\mathcal{R}}_{O}$. It follows directly from Proposition 6 that if O' is inconsistent under the modular stable model semantics w.r.t. $\mathcal{R}$ then O is also inconsistent under the modular stable model semantics w.r.t. $\mathcal{R}$. Obviously, due to the definition of $D^{\mathcal{R}}_{O}$ in Definition 7, all other ontologies in $\mathcal{R}$ remain unaffected from the local inconsistency.

[20] Recall that $\mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$ is a set.

Example 9. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1 and assume that we add to ERDF graph G_{O_2} the ERDF triple $\neg$`rdf:type(geo:Greece,eu:CountryEU)`. Then, the ontologies O_2 and O_5 are inconsistent under the modular stable model semantics w.r.t. $\mathcal{R}$, while the ontologies O_1, O_3, and O_4 are not. □

Definition 15 (Modular ERDF stable model entailment). Let $O \in \mathcal{R}$. Additionally, let F be an `r`-ERDF formula. We say that O *entails* F w.r.t. $\mathcal{R}$ under the *modular ERDF stable model semantics*, denoted by $O \models^{\mathsf{st}}_{\mathcal{R}} F$ iff for all $\mathsf{M} \in \mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}}$, $\langle \mathsf{M}, O \rangle \models F$. □

Our framework supports directed semantic relations, as if an ontology $O' \in \mathcal{R}$ imports information from another ontology $O \in \mathcal{R}$ then this affects reasoning in O' but not in O. In particular, the following proposition holds.

Proposition 7. Let $O, O' \in \mathcal{R}$ s.t. $O' \notin D^{\mathcal{R}}_{O}$. Then, for any `r`-ERDF formula F, it holds that $O \models^{\mathsf{st}}_{\mathcal{R}} F$ iff $O \models^{\mathsf{st}}_{\mathcal{R}-\{O'\}} F$.

The proof of Proposition 7 is straightforward as $\mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}-\{O'\}}$.

Example 10. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. Note that $O_5 \notin D^{\mathcal{R}}_{x}$ for $x \in \{O1, O_2, O_3, O_4\}$. Therefore, $\mathcal{M}^{\mathsf{st}}_{x,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{x,\mathcal{R}-\{O_5\}}$, for $x \in \{O_1, O_2, O_3, O_4\}$. Additionally, note that $\mathcal{M}^{\mathsf{st}}_{O_1,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{O_1,\{O_1\}}$, $\mathcal{M}^{\mathsf{st}}_{O_2,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{O_2,\{O_1,O_2\}}$, $\mathcal{M}^{\mathsf{st}}_{O_3,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{O_3,\{O_3\}}$, and $\mathcal{M}^{\mathsf{st}}_{O_4,\mathcal{R}} = \mathcal{M}^{\mathsf{st}}_{O_4,\{O_4\}}$. This is because ontologies O_1, O_3, and O_4 do not import any property or class from another ontology, while ontology O_2 imports only from ontology O_1. □

The following proposition shows that modular stable model entailment extends $\#n$-stable model entailment [7, 5] (also reviewed in Appendix A) from ERDF ontologies to modular ERDF ontologies.

Proposition 8. Let $O = \langle G, P \rangle$ be an ERDF ontology and let O' be an ontology such that $G_{O'} = G$, $P_{O'} = P$, and $Int_{O'} = \{\}$. Additionally, let $\mathcal{R} = \{O'\}$ and let F be an ERDF formula over $V^{\#n_{O'}}_{O}$. It holds that: $O \models^{st_{\#n_{O'}}} F$ iff $O' \models^{\mathsf{st}}_{\mathcal{R}} F$.

Further, the following proposition shows that modular stable model entailment extends RDFS entailment from RDF graphs to modular ERDF ontologies.

Proposition 9. Let G, G' be RDF graphs such that $V_G \cap \mathcal{V}_{ERDF} = \emptyset$, $V_{G'} \cap \mathcal{V}_{ERDF} = \emptyset$, and $V_{G'} \cap sk_G(Var(G)) = \emptyset$. Let O be an ontology with $G_O = G$, $P_O = \{\}$, and $Int_O = \{\}$. If $max(\{i \in I\!N \mid rdf{:}\_i \in V_{G'}\}) \leq n_O$ then: $G \models^{RDFS} G'$ iff $O \models^{\mathsf{st}}_{\mathcal{R}} G'$, where $\mathcal{R} = \{O\}$.

The following corollary follows from Proposition 8.

Corollary 1. Let $O = \langle G, P \rangle$ be an ERDF ontology and let O' be an ontology such that $G_{O'} = G$, $P_{O'} = P$, and $Int_{O'} = \{\}$. It holds that O has an $\#n_{O'}$-stable model iff O' has a modular stable model w.r.t. $\mathcal{R}$, where $\mathcal{R} = \{O'\}$.

Let $O \in \mathcal{R}$. Additionally, let F be an `r`-ERDF formula. The *modular stable answers* of F w.r.t. O and $\mathcal{R}$ are defined as follows[21]:

$$Ans^{\mathsf{st}}_{O,\mathcal{R}}(F) = \begin{cases} \text{“yes”} & \text{if } FVar(F) = \emptyset \text{ and } O \models^{\mathsf{st}}_{\mathcal{R}} F \\ \text{“no”} & \text{if } FVar(F) = \emptyset \text{ and } O \not\models^{\mathsf{st}}_{\mathcal{R}} F \\ \{v : FVar(F) \to V_{O,\mathcal{R}} \mid O \models^{\mathsf{st}}_{\mathcal{R}} v(F)\}, & \text{if } FVar(F) \neq \emptyset \end{cases}$$

Example 11. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. Then: $Ans^{\mathsf{st}}_{O_5,\mathcal{R}}$(`ann:choose_trav_package`$(?x, ?y)) = \{\langle ?x =$`pyr:package1`$, ?y =$ `geo:Egypt`$\rangle$, $\langle ?x =$ `trav:package2`$, ?y =$ `geo:Croatia`$\rangle\}$. This follows from Figure 2. □

[21] $v(F)$ is the formula F after replacing all the free variables x in F by $v(x)$.

6 Query Answering on Simple Ontologies

In this Section, we consider the problem of query answering on simple ontologies w.r.t. a modular ERDF ontology under the modular ERDF stable model semantics and provide related complexity results.

Definition 16 (Simple r-ERDF formula, simple ontology). An r-ERDF formula is called *simple*, if it has the form: $t_1 \wedge ... \wedge t_k \wedge \sim t_{k+1} \wedge ... \wedge \sim t_n$, where each t_i is a normal or qualified ERDF triple (positive or negative). An r-ERDF program P is called *simple* if the body of each rule in P is simple, or true. Let $\mathcal{R}$ be a modular ERDF ontology and let $O \in \mathcal{R}$. O is called *simple* w.r.t. $\mathcal{R}$, if for each $O' \in D_O^{\mathcal{R}}$, $P_{O'}$ is simple. □

Let $O \in \mathcal{R}$ s.t. O is simple w.r.t. $\mathcal{R}$. We will show that the modular stable answers of a simple r-ERDF formula F w.r.t. O and $\mathcal{R}$ can be computed through Answer Set Programming [31] on an Extended Logic Program (ELP) $\Pi_{O,\mathcal{R}}$.

To give the precise definition of $\Pi_{O,\mathcal{R}}$, a few auxiliary definitions are needed[22]. Let $O' \in D_O^{\mathcal{R}}$. Let t be a normal or qualified ERDF triple (positive or negative), true, or false. We define:
(i) $L_t^{O'} = [\neg]H(Nam_{O'}, s, p, o)$, if $t = [\neg]p(s, o)$,
(ii) $L_t^{O'} = [\neg]H(Nam_{O''}, s, p, o)$, if $t = [\neg]p(s, o)@Nam_{O''}$, and
(iii) $L_t^{O'} = t$, if $t \in \{\texttt{true}, \texttt{false}\}$.
Note that $H(Nam_{O'}, s, p, o)$ is an ELP literal, indicating that the ERDF triple $p(s, o)$ holds in the ontology O'. In particular, the predicate H stands for *Holds*.

Let G be an ERDF graph. We define: $\Pi_G^{O'} = \{L_t^{O'} \leftarrow \texttt{true} \mid t \in sk(G)\}$.

Example 12. Consider the ontology O_2 of Example 1. It holds that:

$$\begin{aligned}\Pi_{G_{O_2}}^{O_2} = \{&H(Nam_{O_2}, \texttt{eu:CountryEU}, \texttt{rdf:type}, \texttt{erdf:TotalClass}) \leftarrow \texttt{true},\\ &H(Nam_{O_2}, \texttt{geo:Italy}, \texttt{rdf:type}, \texttt{eu:CountryEU}) \leftarrow \texttt{true},\\ &H(Nam_{O_2}, \texttt{geo:Greece}, \texttt{rdf:type}, \texttt{eu:CountryEU}) \leftarrow \texttt{true}, ...\}. \ \square\end{aligned}$$

Note that all ERDF triples of a ERDF ontology $sk(O)$ are considered true.

Let F be a simple r-ERDF formula of the form: $F = t_1 \wedge ... \wedge t_k \wedge \sim t_{k+1} \wedge ... \wedge \sim t_n$. We define: $L_F^{O'} = L_{t_1}^{O'} \wedge ... \wedge L_{t_k}^{O'} \wedge \sim L_{t_{k+1}}^{O'} \wedge ... \wedge \sim L_{t_n}^{O'}$.

Let P be a simple r-ERDF program. We define: $\Pi_P^{O'} = \{L_t^{O'} \leftarrow L_F^{O'} \mid t \leftarrow F \in P\}$.

Example 13. Let:
$P = \{\neg\texttt{ann:choose\_trav\_package}(?p, ?c) \leftarrow \sim\texttt{peter:likes\_package}(?p, ?c)@Nam_{O''}\}$.
Then,

$$\begin{aligned}\Pi_P^{O'} = \{&\neg H(Nam_{O'}, ?p, \texttt{ann:choose\_trav\_package}, ?c) \leftarrow\\ &\sim H(Nam_{O''}, ?p, \texttt{peter:likes\_package}, ?c). \} \ \square\end{aligned}$$

We denote by $\Pi_{O'}^{H_R}$ the ELP that consists of the following three sets of rules:

Importing Rules
If $p \in Imported_{O',\mathcal{R}}^{\texttt{pr}}$ and $Nam_{O''} \in Import_{O',\mathcal{R}}^{\texttt{pr}}(p)$ then:
$H(Nam_{O'}, ?x, p, ?y) \leftarrow H(Nam_{O''}, ?x, p, ?y)$.
$\neg H(Nam_{O'}, ?x, p, ?y) \leftarrow \neg H(Nam_{O''}, ?x, p, ?y)$.

If $c \in Imported_{O',\mathcal{R}}^{\texttt{cl}}$ and $Nam_{O''} \in Import_{O',\mathcal{R}}^{\texttt{cl}}(c)$ then:
$H(Nam_{O'}, ?x, \texttt{type}, c) \leftarrow H(Nam_{O''}, ?x, \texttt{type}, c)$.
$\neg H(Nam_{O'}, ?x, \texttt{type}, c) \leftarrow \neg H(Nam_{O''}, ?x, \texttt{type}, c)$.

[22] For simplicity, we have eliminated the namespace from the URIs in $\mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$.

Partial Interpretation Rules

$H(Nam_{O'}, ?z, \mathtt{type}, \mathtt{Property}) \leftarrow H(Nam_{O'}, ?x, ?z, ?y)$.
$H(Nam_{O'}, ?z, \mathtt{type}, \mathtt{Property}) \leftarrow \neg H(Nam_{O'}, ?x, ?z, ?y)$.
For all $x \in V_{O',\mathcal{R}}$: $H(Nam_{O'}, x, \mathtt{type}, \mathtt{Resource}) \leftarrow \mathtt{true}$.
For all $x \in V_{O',\mathcal{R}} \cap \mathcal{PL}$: $H(Nam_{O'}, x, \mathtt{type}, \mathtt{Literal}) \leftarrow \mathtt{true}$.

ERDF Interpretation Rules

$H(Nam_{O'}, ?z, \mathtt{type}, ?y) \leftarrow H(Nam_{O'}, ?x, \mathtt{domain}, ?y), H(Nam_{O'}, ?z, ?x, ?w)$.
$H(Nam_{O'}, ?w, \mathtt{type}, ?y) \leftarrow H(Nam_{O'}, ?x, \mathtt{range}, ?y), H(Nam_{O'}, ?z, ?x, ?w)$.
$H(Nam_{O'}, ?x, \mathtt{subClassOf}, \mathtt{Resource}) \leftarrow H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Class})$.

$H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Class}) \leftarrow H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?y)$.
$H(Nam_{O'}, ?y, \mathtt{type}, \mathtt{Class}) \leftarrow H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?y)$.
$H(Nam_{O'}, ?z, \mathtt{type}, ?y) \leftarrow H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?y), H(Nam_{O'}, ?z, \mathtt{type}, ?x)$.
$\neg H(Nam_{O'}, ?z, \mathtt{type}, ?x) \leftarrow H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?y), \neg H(Nam_{O'}, ?z, \mathtt{type}, ?y)$.
$H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?x) \leftarrow H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Class})$.
$H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?z) \leftarrow H(Nam_{O'}, ?x, \mathtt{subClassOf}, ?y),$
$H(Nam_{O'}, ?y, \mathtt{subClassOf}, ?\mathtt{z})$.

$H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Property}) \leftarrow H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?y)$.
$H(Nam_{O'}, ?y, \mathtt{type}, \mathtt{Property}) \leftarrow H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?y)$.
$H(Nam_{O'}, ?z_1, ?y, ?z_2) \leftarrow H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?y), H(Nam_{O'}, ?z_1, ?x, ?z_2)$.
$\neg H(Nam_{O'}, ?z_1, ?x, ?z_2) \leftarrow H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?y), \neg H(Nam_{O'}, ?z_1, ?y, ?z_2)$.
$H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?x) \leftarrow H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Property})$.
$H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?z) \leftarrow H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, ?y),$
$H(Nam_{O'}, ?y, \mathtt{subPropertyOf}, ?z)$.

$H(Nam_{O'}, ?x, \mathtt{subClassOf}, \mathtt{Literal}) \leftarrow H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{Datatype})$.
$H(Nam_{O'}, ?x, \mathtt{subPropertyOf}, \mathtt{member}) \leftarrow H(Nam_{O'}, ?x, \mathtt{type}, \mathtt{ContainerMembershipProperty})$.

$\neg H(Nam_{O'}, ?x, \mathtt{type}, ?c) \leftarrow H(Nam_{O'}, ?c, \mathtt{type}, \mathtt{TotalClass}), \sim H(Nam_{O'}, ?x, \mathtt{type}, ?c)$. $(\star)$
$H(Nam_{O'}, ?x, \mathtt{type}, ?c) \leftarrow H(Nam_{O'}, ?c, \mathtt{type}, \mathtt{TotalClass}), \sim \neg H(Nam_{O'}, ?x, \mathtt{type}, ?c)$.

$\neg H(Nam_{O'}, ?x, ?p, ?y) \leftarrow H(Nam_{O'}, ?p, \mathtt{type}, \mathtt{TotalProperty}), \sim H(Nam_{O'}, ?x, ?p, ?y)$. $(\star)$
$H(Nam_{O'}, ?x, ?p, ?y) \leftarrow H(Nam_{O'}, ?p, \mathtt{type}, \mathtt{TotalProperty}), \sim \neg H(Nam_{O'}, ?x, ?p, ?y)$.

For each "s"^^rdf:XMLLiteral $\in V_{O',\mathcal{R}}$ s.t. s is a well-typed XML literal string:
$H(Nam_{O'}, \text{“}s\text{”}$^^rdf:XMLLiteral$, \mathtt{type}, \mathtt{XMLLiteral}) \leftarrow \mathtt{true}$.
For each "s"^^rdf:XMLLiteral $\in V_{O',\mathcal{R}}$ s.t. s is not a well-typed XML literal string:
$\neg H(Nam_{O'}, \text{“}s\text{”}$^^rdf:XMLLiteral$, \mathtt{type}, \mathtt{Literal}) \leftarrow \mathtt{true}$.
For each RDF, RDFS, or ERDF axiomatic triple $p(s, o)$ s.t. $p, s, o \in V_{RDF}^{\#n_{O'}} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$:
$H(Nam_{O'}, s, p, o) \leftarrow \mathtt{true}$.

The importing rules are due to Definition 9. The partial interpretation rules are due to Definition 6 and item 2 of Definition 8. The ERDF interpretation rules are due to the rest of the semantic conditions in Definition 8. Let c be a total class and p be a total property of O'. Due to the ERDF Interpretation Rules, indicated by $(\star)$, the OWA applies to the local truth values of rdf:type(x, c) and $p(x, y)$. These correspond to semantic conditions 12 and 13 of Definition 8. For example, regarding semantic condition 12, if in a ontology O', $\mathtt{type}(c, \mathtt{TotalClass})$ holds (in an interpretation) and $\mathtt{type}(x, c)$ does not hold then $\neg\mathtt{type}(x, c)$ holds. Additionally, if $\mathtt{type}(c, \mathtt{TotalClass})$ holds (in an interpretation) and $\neg\mathtt{type}(x, c)$ does not hold then $\mathtt{type}(x, c)$ holds. Similarly, regarding semantic condition 13, if in a ontology O', $\mathtt{type}(p, \mathtt{TotalProperty})$ holds and $p(x, y)$ does not hold then $\neg p(x, y)$ holds. Additionally, if $\mathtt{type}(p, \mathtt{TotalProperty})$ holds and $\neg p(x, y)$ does not hold then $p(x, y)$ holds.

Finally, we define[23]:

$$\Pi_{O,\mathcal{R}} = \bigcup\{\Pi^{O'}_{G_{O'}} \cup [\Pi^{O'}_{P_{O'}}]_{V_{O',\mathcal{R}}} \cup [\Pi^{H_\mathcal{R}}_{O'}]_{V_{O',\mathcal{R}}} \mid O' \in D^{\mathcal{R}}_{O}\}.$$

Note that for each $O' \in D^{\mathcal{R}}_{O}$, the ELP $\Pi^{O'}_{P_{O'}} \cup \Pi^{H_\mathcal{R}}_{O'}$ is instantiated over the local domain $V_{O',\mathcal{R}}$. This is due to the fact that in Definition 14, item 2, we consider `r`-ERDF rules $r \in [P_{O'}]_{V_{O',\mathcal{R}}}$. The $[\Pi^{H_\mathcal{R}}_{O'}]_{V_{O',\mathcal{R}}}$ part is needed for inferences that correspond to the semantic conditions of a modular Herbrand interpretation which are found in Definitions 8, 9, and 12.

Example 14. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1. The importing rules of $\Pi^{H_\mathcal{R}}_{O_5}$ are the following:
$[\neg]H(Nam_{O_5}, ?x, \texttt{geo:capital}, ?y) \leftarrow [\neg]H(Nam_{O_1}, ?x, \texttt{geo:capital}, ?y)$.
$[\neg]H(Nam_{O_5}, ?x, \texttt{vac:travel}, ?y) \leftarrow [\neg]H(Nam_{O_3}, ?x, \texttt{vac:travel}, ?y)$.
$[\neg]H(Nam_{O_5}, ?x, \texttt{vac:travel}, ?y) \leftarrow [\neg]H(Nam_{O_4}, ?x, \texttt{vac:travel}, ?y)$.
$[\neg]H(Nam_{O_5}, ?x, \texttt{vac:visit}, ?y) \leftarrow [\neg]H(Nam_{O_3}, ?x, \texttt{vac:visit}, ?y)$.
$[\neg]H(Nam_{O_5}, ?x, \texttt{vac:visit}, ?y) \leftarrow [\neg]H(Nam_{O_4}, ?x, \texttt{vac:visit}, ?y)$.
$[\neg]H(Nam_{O_5}, ?x, \texttt{type}, \texttt{eu:CountryEU}) \leftarrow [\neg]H(Nam_{O_2}, ?x, \texttt{type}, \texttt{eu:CountryEU})$. □

Convention: In the sequel, to simplify notation, by $[\Pi^{O'}_{P_{O'}}]$ *we will denote* $[\Pi^{O'}_{P_{O'}}]_{V_{O',\mathcal{R}}}$ *and by* $[\Pi^{H_\mathcal{R}}_{O'}]$*, we will denote* $[\Pi^{H_\mathcal{R}}_{O'}]_{V_{O',\mathcal{R}}}$. Additionally, by $\mathtt{G}^{\mathcal{R}}_{O}$, we will denote $\cup_{O' \in D^{\mathcal{R}}_{O}} \Pi^{O'}_{G_{O'}}$ and by $\mathtt{H}^{\mathcal{R}}_{O}$, we will denote $\cup_{O' \in D^{\mathcal{R}}_{O}} [\Pi^{H_\mathcal{R}}_{O'}]$. Further, by $\mathtt{P}^{\mathcal{R}}_{O}$, we will denote $\cup_{O' \in D^{\mathcal{R}}_{O}} [\Pi^{O'}_{P_{O'}}]$.

To proceed in computing the modular stable models of an ontology w.r.t. a modular ERDF ontology, we need the following auxiliary definition. Intuitively, a modular semi-Herbrand interpretation of an ontology O w.r.t. a modular ERDF ontology $\mathcal{R}$ is a set of coherent, partial interpretations that satisfies the three conditions of a modular Herbrand interpretation (Definition 12), while the rest of the conditions are just definitions. The definition of a modular semi-Herbrand interpretation is needed for mapping modular Herbrand interpretations to consistent sets of ELP literals and vice-versa.

Definition 17 (Modular semi-Herbrand interpretation). Let $O \in \mathcal{R}$. A *modular semi-Herbrand interpretation* of O w.r.t. $\mathcal{R}$ is a set $\mathsf{I} = \{I_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$, where $I_{O'}$ is a coherent, partial interpretation of $V_{O',\mathcal{R}}$, extended by the ontological categories $Cls_{I_{O'}} \subseteq Res_{I_{O'}}$ for classes, $TCls_{I_{O'}} \subseteq Cls_{I_{O'}}$ for total classes, and $TProp_{I_{O'}} \subseteq Prop_{I_{O'}}$ for total properties, as well as the class-truth extension mapping $CT_{I_{O'}} : Cls_{I_{O'}} \rightarrow \mathcal{P}(Res_{I_{O'}})$, and the class-falsity extension mapping $CF_{I_{O'}} : Cls_{I_{O'}} \rightarrow \mathcal{P}(Res_{I_{O'}})$, such that:

1. $Res_{I_{O'}} = Res^{\mathtt{H}}_{O',\mathcal{R}}$.
2. $I_{O'\mathtt{V}}(x) = x$, for all $x \in V_{O',\mathcal{R}} \cap URI$.
3. $IL_{I_{O'}}(x) = x$, if x is a typed literal in $V_{O',\mathcal{R}}$ other than a well-typed XML literal, and $IL_{I_{O'}}(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{O',\mathcal{R}}$.
4. $x \in CT_{I_{O'}}(y)$ iff $\langle x, y\rangle \in PT_{I_{O'}}(rdf{:}type)$, and
   $x \in CF_{I_{O'}}(y)$ iff $\langle x, y\rangle \in PF_{I_{O'}}(rdf{:}type)$.
5. Additionally:
   $Prop_{I_{O'}} = CT_{I_{O'}}(rdf{:}Property)$ $\quad Cls_{I_{O'}} = CT_{I_{O'}}(rdfs{:}Class)$
   $Res_{I_{O'}} = CT_{I_{O'}}(rdfs{:}Resource)$ $\quad LV_{I_{O'}} = CT_{I_{O'}}(rdfs{:}Literal)$
   $TCls_{I_{O'}} = CT_{I_{O'}}(erdf{:}TotalClass)$ $\quad TProp_{I_{O'}} = CT_{I_{O'}}(erdf{:}TotalProperty)$. □

[23] Let Π be an ELP and let C be a set of constants. By $[\Pi]_C$, we denote the instantiation of Π w.r.t. the constants appearing in C.

Let $O \in \mathcal{R}$. Below, we show how a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$ can be translated to a consistent Herbrand interpretation of $\Pi_{O,\mathcal{R}}$, denoted by $ELP(\mathsf{I})$. Let I be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$. We define:

$$
\begin{aligned}
ELP(\mathsf{I}) = \; & \{H(Nam_{O'}, s, p, o) \mid s, p, o \in V_{O',\mathcal{R}} \text{ and } \langle s, o\rangle \in PT_{I_{O'}}(p), \\
& \qquad \text{for } O' \in D_O^{\mathcal{R}}\} \cup \\
& \{\neg H(Nam_{O'}, s, p, o) \mid s, p, o \in V_{O',\mathcal{R}} \text{ and } \langle s, o\rangle \in PF_{I_{O'}}(p), \\
& \qquad \text{for } O' \in D_O^{\mathcal{R}}\}
\end{aligned}
$$

Note that the function $ELP(.)$ from the set of modular semi-Herbrand interpretations of O w.r.t. $\mathcal{R}$ to the set of consistent sets of ELP literals $H(Nam_{O'}, s, p, o)$ and $\neg H(Nam_{O'}, s, p, o)$, where $O' \in D_O^{\mathcal{R}}$ and $s, p, o \in V_{O',\mathcal{R}}$, is a bijective mapping.

Now, we give a few definitions that will be used throughout several proofs. Let Π be an ELP. By $[\Pi]$, we denote the grounded version of Π and by $EHB(\Pi)$, we denote the *Extended Herbrand Base*[24] of Π. Let Π be a ground ELP. Let $r = L_0 \leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n \in \Pi$. We define: $Head(r) = L_0$, $Body(r)^+ = \{L_1, ..., L_m\}$ and $Body(r)^- = \{L_{m+1}, ..., L_n\}$.. Let $N \subseteq EHB(\Pi)$. We define: $\Pi^N = \{Head(r) \leftarrow Body(r)^+ \mid r \in \Pi \text{ and } Body(r)^- \cap N = \emptyset\}$.

Let Π be a ground ELP such that for any $r \in \Pi$, it holds $Body(r)^- = \emptyset$. We define the mapping $T_\Pi : \mathcal{P}(EHB(\Pi)) \to \mathcal{P}(EHB(\Pi))$, where $T_\Pi(N) = N \cup \{Head(r) \mid r \in \Pi$ and $Body^+(r) \subseteq N\}$.

Proposition 10. Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. Let M be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$. It holds that: $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ iff $ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$.

Based on Proposition 10, we present the Algorithm 1 *All-ModularStableModelsSimple*($\mathcal{R}, O$) that takes as input a modular ERDF ontology $\mathcal{R}$ and a simple ERDF ontology $O \in \mathcal{R}$ and computes the set of all modular stable models of O w.r.t. $\mathcal{R}$.

Algorithm 1 *All-ModularStableModelsSimple*($\mathcal{R}, O$)
Input: (i) a modular ERDF ontology $\mathcal{R}$ and (ii) a simple ontology $O \in \mathcal{R}$ w.r.t. $\mathcal{R}$
Output: the set of all modular stable models of O w.r.t. $\mathcal{R}$

(1) $S = \{ELP^{-1}(N) \mid N$ is a consistent answer set of $\Pi_{O,\mathcal{R}}\}$;
(2) *return*(S);

Using Proposition 10, we can show that the modular stable answers of a simple r-ERDF formula F w.r.t. a simple ERDF ontology O w.r.t. $\mathcal{R}$ can be computed through Answer Set Programming [31] on $\Pi_{O,\mathcal{R}}$. First, we provide a few definitions. Let Π be an ELP and let F be a query of the form: $L_1 \wedge ... \wedge L_k \wedge \sim L_{k+1} \wedge ... \wedge \sim L_m$, where $L_i, i = 1, ..., m$, is an ELP literal. We will denote by $Ans^{AS}_\Pi(F)$ the (skeptical) answers of F w.r.t. Π according to answer set semantics [31].

Proposition 11. Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. Additionally, let F be a simple r-ERDF formula over $\{Nam_{O'} \mid O' \in D_O^{\mathcal{R}}\}$. Then,
(i) if $\Pi_{O,\mathcal{R}}$ is a non-contradictory ELP [31] then $Ans^{\mathtt{st}}_{O,\mathcal{R}}(F) = Ans^{AS}_{\Pi_{O,\mathcal{R}}}(L^O_F)$, and
(ii) $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}} = \emptyset$, otherwise.

Below, we state several complexity results of the modular ERDF stable model semantics. We define[25] $size(O, \mathcal{R}) = \sum\{\text{size of } (G_{O'} \cup P_{O'}) \mid O' \in D_O^{\mathcal{R}}\}$.

[24] The *Extended Herbrand Base* of an ELP Π is the set of ELP literals $p(\overline{c})$ and $\neg p(\overline{c})$ that can be formed using the language of Π.

[25] The *size* of a ground logic program is the summation of the size of its rules.

Proposition 12. Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. The problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$.

Below, we present Algorithm 2 $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F)$ that takes as input a modular ERDF ontology $\mathcal{R}$, an ontology $O \in \mathcal{R}$, a modular ERDF interpretation I of O w.r.t. $\mathcal{R}$, an ontology $O' \in D_O^{\mathcal{R}}$, and an r-ERDF formula F over $\{Nam_{O'} \mid O' \in D_O^{\mathcal{R}}\}$ with $FVar(F) = \emptyset$. This algorithm returns TRUE iff $\langle \mathsf{I}, O' \rangle \models F$. The algorithm and Proposition 12 will be used to provide the complexity of the query answering problem on a simple ontology w.r.t. the modular stable model semantics. Note that the algorithm is derived directly from the items of Definition 10.

Algorithm 2 $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F)$
Input: a modular ERDF ontology $\mathcal{R}$, an ontology $O \in \mathcal{R}$,
a modular ERDF interpretation I of O w.r.t. $\mathcal{R}$,
an ontology $O' \in D_O^{\mathcal{R}}$,
and an r-ERDF formula F over $\{Nam_{O'} \mid O' \in D_O^{\mathcal{R}}\}$ with $FVar(F) = \emptyset$.
Output: TRUE, if $\langle \mathsf{I}, O' \rangle \models F$, and FALSE, otherwise

(1) case(F) {
(2) $p(s, o)$: If $p \in V_{I_{O'}} \cap URI$, $s, o \in V_{I_{O'}}$, $p \in Prop_{I_{O'}}$, and $\langle I_{O'}(s), I_{O'}(o) \rangle \in PT_{I_{O'}}(p)$ then return(TRUE);
(3) $\neg p(s, o)$: If $p \in V_{I_{O'}} \cap URI$, $s, o \in V_{I_{O'}}$, $p \in Prop_{I_{O'}}$, and $\langle I_{O'}(s), I_{O'}(o) \rangle \in PF_{I_{O'}}(p)$ then return(TRUE);
(4) $\sim G$: If $Satisfies(\mathcal{R}, O, \mathsf{I}, O', G)$=FALSE then return(TRUE);
(5) $F_1 \wedge F_2$: If $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F_1)$=TRUE and $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F_2)$=TRUE then return(TRUE);
(6) $F_1 \vee F_2$: If $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F_1)$=TRUE or $Satisfies(\mathcal{R}, O, \mathsf{I}, O', F_2)$=TRUE then return(TRUE);
(7) $F_1 \supset F_2$: return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', \sim F_1 \vee F_2)$);
(8) $\forall x G$: If for all $u : \{x\} \to V_{I_{O'}}$, it holds that $Satisfies(\mathcal{R}, O, \mathsf{I}, O', u(G))$=TRUE then return(TRUE);
(9) $\exists x G$: If it exists $u : \{x\} \to V_{I_{O'}}$ such that $Satisfies(\mathcal{R}, O, \mathsf{I}, O', u(G))$=TRUE then return(TRUE);
(10) $G@Nam_{O''}$: return($Satisfies(\mathcal{R}, O, \mathsf{I}, O'', G)$);
(11) $\neg(F_1 \wedge F_2)$: return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', \neg F_1 \vee \neg F_2)$);
(12) $\neg(F_1 \vee F_2)$: return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', \neg F_1 \wedge \neg F_2)$);
(13) $\neg(\neg G)$: return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', G)$);
(14) $\neg(\sim G)$ return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', G)$);
(15) $\neg(\exists x\, G)$ return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', \forall x\, \neg G)$);
(16) $\neg(\forall x\, G)$ return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', \exists x\, \neg G)$);
(17) $\neg(F_1 \supset F_2)$ return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', F_1 \wedge \neg F_2)$);
(18) $\neg(G@Nam_{O''})$ return($Satisfies(\mathcal{R}, O, \mathsf{I}, O', (\neg G)@Nam_{O''})$);
}
(19) return(FALSE);

Example 15. Consider the modular rule base $\mathcal{R}$ of Example 1. Additionally, consider the modular stable model M_1 of O_5 w.r.t. $\mathcal{R}$ of Example 8. Let $F = \forall$`?city'` `vac:visit(pyr:package1, ?city')`@⟨`http://www.pyramis.gr`⟩ ⊃`eq:id(geo:Cairo, ?city')`. The formula F expresses that the only city visited by package `pr:package1` from the travel agency `pyramis` is `geo:Cairo`. It holds that $Satisfies(\mathcal{R}, O_5, \mathsf{M}_1, O_5, F)$=TRUE. Note that for all `city'` $\in V_{O_5,\mathcal{R}}$, it holds that $\langle \mathsf{M}_1, O_5 \rangle \models \sim$`vac:visit(pyr:package1, city')`@⟨`http://www.pyramis.gr`⟩∨ `eq:id(geo: Cairo, city')`. □

Proposition 13. Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$ and let F be an $\mathtt{r}$-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \rightarrow V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. The problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is co-NP$^{\text{NP}}$-complete w.r.t. $size(O, \mathcal{R})$.

7 Computing the Modular Stable Models of General Ontologies

In this Section, we compute the modular stable models of a general ontology w.r.t. a modular ERDF ontology and provide related complexity results.

First, we present Algorithm 3 *All-ModularStableModelsGeneral*$(\mathcal{R}, O)$ that takes as input a modular ERDF ontology $\mathcal{R}$ and an ontology $O \in \mathcal{R}$ and computes the set of all modular stable models of O w.r.t. $\mathcal{R}$. This algorithm calls Algorithm 4 *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ that takes as input a modular ERDF ontology $\mathcal{R}$, an ontology $O \in \mathcal{R}$, and a modular semi-Herbrand interpretation M of O w.r.t. $\mathcal{R}$ and returns TRUE, if M is a modular stable model of O w.r.t. $\mathcal{R}$, and FALSE, otherwise. In particular, Algorithm *All-ModularStableModelsGeneral*$(\mathcal{R}, O)$ generates all modular semi-Herbrand interpretations M of O w.r.t. $\mathcal{R}$ and if *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ returns TRUE then M is a modular stable model of O w.r.t. $\mathcal{R}$, which is added to a set S that is finally returned.

Algorithm 3 *All-ModularStableModelsGeneral*$(\mathcal{R}, O)$
Input: (i) a modular ERDF ontology $\mathcal{R}$ and (ii) an ontology $O \in \mathcal{R}$
Output: the set of all modular stable models of O w.r.t. $\mathcal{R}$

(1) S={};
(2) For all modular semi-Herbrand interpretations M of O w.r.t. $\mathcal{R}$ do
(3) If *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$=TRUE then $S = S \cup \{\mathsf{M}\}$;
(4) Endfor
(5) *return*(S);

Algorithm *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ calls Algorithm 5 *MiddleNotSatisfies*$(\mathcal{R}, O, O', \mathsf{I}, \mathsf{M}, F)$ which takes as input (i) a modular ERDF ontology $\mathcal{R}$, (ii) an ontology $O \in \mathcal{R}$, (iii) an ontology $O' \in D^{\mathcal{R}}_O$, (iv) two modular ERDF interpretations I, M of O w.r.t. $\mathcal{R}$, and (v) an formula F over $\{Nam_{O'} \mid O' \in D^{\mathcal{R}}_O\}$ with $FVar(F) = \emptyset$ and returns TRUE, if it exists $\mathsf{J} \in [\mathsf{I}, \mathsf{M}]_{O,\mathcal{R}}$ and $\langle \mathsf{J}, O' \rangle \not\models F$, and FALSE, otherwise. In particular, Algorithm *MiddleNotSatisfies*$(\mathcal{R}, O, O', \mathsf{I}, \mathsf{M}, F)$ checks if it exists a modular Herbrand interpretation J of O w.r.t. $\mathcal{R}$ (line 2) s.t. $\mathsf{I} \leq \mathsf{J} \leq \mathsf{M}$ (line 3) and $\langle \mathsf{J}, O' \rangle \not\models F$ (line 4). If there is such a J, it returns TRUE else it returns FALSE.

Algorithm 4 *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$
Input: (i) a modular ERDF ontology $\mathcal{R}$, (ii) an ontology $O \in \mathcal{R}$, and
(iii) a modular semi-Herbrand interpretation M of O w.r.t. $\mathcal{R}$
Output: TRUE, if M is a modular stable model of O w.r.t. $\mathcal{R}$, and FALSE, otherwise

(1) Let $N = ELP(\mathsf{M})$;
(2) If $T_{(\mathtt{H}^{\mathcal{R}}_O)^N}(N) \not\subseteq N$ then return(FALSE);
/* In this case, M is not a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$ */

(3) Let $N_0 = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_{O})^N}(T_{\mathtt{G}^{\mathcal{R}}_{O}}(\emptyset))$;
/* Note that $ELP^{-1}(N_0)$ is a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$ */
(4) If $N \subset N_0$ then *return*(FALSE);
(5) Let $\alpha = 0$;
(6) While $N_\alpha \subseteq N$ do
{
(7) $S = N_\alpha$;
(8) $\mathsf{I} = ELP^{-1}(N_\alpha)$;
(9) For each $O' \in D^{\mathcal{R}}_{O}$ do
(10) For each $r \in [P_{O'}]$ do
{
(11) If *MiddleNotSatisfies*$(\mathcal{R}, O, O', \mathsf{I}, \mathsf{M}, Cond(r))$=FALSE then
$S = S \cup \{L^{O'}_{Concl(r)}\}$;
}
(12) $\alpha = \alpha + 1$;
(13) Let $N_\alpha = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_{O})^N}(S)$;
/* Note that $ELP^{-1}(N_\alpha)$ is a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$ */
(14) If $N_\alpha = N_{\alpha-1}$ then
(15) If $N_\alpha = N$ then *return*(TRUE);
(16) else *return*(FALSE);
}
(17) *return*(FALSE);

Algorithm 5 *MiddleNotSatisfies*$(\mathcal{R}, O, O', \mathsf{I}, \mathsf{M}, F)$
Input: (i) a modular ERDF ontology $\mathcal{R}$, (ii) an ontology $O \in \mathcal{R}$,
(iii) an ontology $O' \in D^{\mathcal{R}}_{O}$,
(iii) two modular ERDF interpretations I, M of O w.r.t. $\mathcal{R}$, and
(iv) an r-ERDF formula F over $\{Nam_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$ with $FVar(F) = \emptyset$
Output: TRUE, if it exists $\mathsf{J} \in [\mathsf{I}, \mathsf{M}]_{O,\mathcal{R}}$ and $\langle \mathsf{J}, O' \rangle \not\models F$, and FALSE, otherwise.

(1) If a modular semi-Herbrand interptetation J of O w.r.t. $\mathcal{R}$ exists s.t.
(2) (i) $T_{(\mathtt{H}^{\mathcal{R}}_{O})^N}(N) \subseteq N$, where $N = ELP(\mathsf{J})$,
/* i.e. J is a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$ */
(3) (ii) $\mathsf{I} \leq \mathsf{J} \leq \mathsf{M}$, and
(4) (iii) *Satisfies*$(\mathcal{R}, O, \mathsf{J}, O', F)$=FALSE
(5) then *return*(TRUE);
(6) else *return*(FALSE);

Algorithm *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ follows the steps of Definition 14. In particular, in line (2), it checks if M is a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$, and if it is not then it returns FALSE. In line (3), it computes $N_0 = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_{O})^N}(T_{\mathtt{G}^{\mathcal{R}}_{O}}(\emptyset))$. Thus, $ELP^{-1}(N_0) \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D^{\mathcal{R}}_{O}, \ \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$. In line (4), it checks if $\mathsf{M} < ELP^{-1}(N_0)$ (or, equivalently $N \subset N_0$), and if this is the case then it returns FALSE. Then, in line (6), it enters into a loop for $\alpha \in \{0, 1, ...\}$, where in line (13) it computes $N_{\alpha+1} = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_{O})^N}(S)$, for $S = N_\alpha \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_{O},\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_\alpha), \mathsf{M}]_{O,\mathcal{R}}\}$, until $ELP^{-1}(N_\alpha) = ELP^{-1}(\ N_{\alpha+1})$ or $ELP^{-1}(N_\alpha) > \mathsf{M}$. Thus, $ELP^{-1}(N_{\alpha+1}) \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq ELP^{-1}(N_\alpha)$ and $\forall O' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in$

$[ELP^{-1}(N_\alpha), \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$. If there is $\alpha \in \{0, 1, ...\}$ such that $ELP^{-1}(N_\alpha) = ELP^{-1}(N_{\alpha+1}) = \mathsf{M}$ then *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ returns TRUE else it returns FALSE.

The following proposition proves formally correctness of Algorithm *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$.

Proposition 14. Let $O \in \mathcal{R}$. Let M be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$. It holds that: $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ iff *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$=TRUE.

Example 16. Consider the modular ERDF ontology $\mathcal{R}$ of Example 1 and $\mathsf{M}_1 \in \mathcal{M}^{\mathtt{st}}_{O_5,\mathcal{R}}$, as defined in Example 8. Assume that we call *Is-ModularStableModelGeneral*$(\mathcal{R}, O_5, \mathsf{M}_1)$. Then, in line (3) of the Algorithm, it holds that $N_0 = ELP(\mathsf{I}_0)$, where I_0 is defined in Example 8. In line (13) of the Algorithm, for $\alpha = 1$, $S = N_0 \cup \{\neg H(Nam_{O_1}$,`geo:Egypt`,`rdf:type`,`geo:Europ_Country`$)$, $H(Nam_{O_5}$,`trav:package2`,`ann:choose_trav_package`,`geo:Croatia`$)\}$, and $N_1 = S \cup \neg H(Nam_{O_5}$, `geo:Egypt`,`rdf:type`,`geo:Europ_Country`$)\}$. Note that $N_1 = ELP(\mathsf{I}_1)$, where I_1 is defined in Example 8. Additionally, note that (i) $\neg H(Nam_{O_1}$, `geo:Egypt`,`rdf:type`,`geo:Europ_Country`$) \in S$ due to the single rule in P_{O_1}, (ii) $H(Nam_{O_5}$,`trav:package2`,`ann:choose_trav_package`,`geo:Croatia`$) \in S$ due to the fourth rule in P_{O_5}, and (iii) $\neg H(Nam_{O_5}$, `geo:Egypt`,`rdf:type`,`geo:Europ_Country`$) \in N_1$, because O_5 imports class `geo:Europ_Country` from O_1. For $\alpha = 2$, $S = N_1 \cup \{H(Nam_{O_5}$, `pyr:package1`,`ann:choose_trav_package`,`geo:Egypt`$)\}$ and $N_2 = S$. Note that $H(Nam_{O_5}$, `pyr:package1`,`ann:choose_trav_package`,`geo:Egypt`$) \in S$ due to the second rule in P_{O_5}. Since $N_3 = N_2 = ELP(\mathsf{M}_1)$, *Is-ModularStableModelGeneral*$(\mathcal{R}, O_5, \mathsf{M}_1)$ returns TRUE.

The following proposition provides the complexity of the modular stable model semantics, in the case that ontologies do not contain quantifiers.

Proposition 15. Let $O \in \mathcal{R}$ s.t. for all $O' \in D^{\mathcal{R}}_O$, no quantifiers $\forall, \exists$ appear in $P_{O'}$ and let F be an `r`-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \rightarrow V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. Then: (i) the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\Sigma^P_2 = \mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$, and (ii) the problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is $\Pi^P_2 = \text{co-}\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$.

The following proposition provides the complexity of the modular stable model semantics, in the general case.

Proposition 16. Let $O \in \mathcal{R}$ and let F be an `r`-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \rightarrow V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. Then: (i) the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is PSPACE-complete w.r.t. $size(O, \mathcal{R})$, and (ii) the problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is PSPACE-complete w.r.t. $size(O, \mathcal{R})$.

8 Related Work

Below, we review related work and compare it with our framework on modular ERDF ontologies.

[*RDF Modularity Frameworks*]

N3Logic [12] allows rules to be integrated with RDF. Indeed, part of the RDFS semantics is represented by program rules. Additionally, it provides quoted N3 formulas and certain built-in functions that allow information from the Web to be accessed and reasoned over. For example, N3Logic allows to check whether a Web resource, identified by a URI, N3-derives or *not* a set of RDF triples. Yet, this supported form of negation as failure, expressed through the built-in `log:notincludes`, is limited. Additionally, N3Logic does not have a model-theoretic semantics that faithfully extends RDFS semantics [42] [26], does not support explicit negation and general formulas in the body of the rules, and ignores visibility issues.

A modularity framework for RDF rule bases (without blank nodes) is proposed in [60]. There, RDFS semantics are partially represented through a normal logic program, associated with a special context/module c_{RDFS}. The *contextually closed* `AS` and *contextually closed* `WFS` semantics of such a modular RDF rule base $\mathcal{R}$ are defined, through the `AS` [31] and `WFS` [29] semantics of a normal logic program $\mathcal{R}_{CC}$, generated from $\mathcal{R}$, respectively. In particular, all simple atoms appearing in a rule base s are qualified by the name of s. The resulting rules union the original rules of each rule base $s \in \mathcal{R}$ form the logic program $\mathcal{R}_{CC}$. Yet, this framework does not have a model-theoretic semantics that faithfully extends RDFS semantics [42], does not support explicit negation and general formulas in the body of the rules, and ignores visibility issues.

TRIPLE [67] is a rule language for the Semantic Web that supports modules (called, *models* there), qualified literals, and dynamic module transformation. Its syntax is based on F-Logic [48]. Arbitrary formulas can be used in the body of a rule, handled through the Lloyd-Topor transformations [54]. Module indicators in qualified literals can be module names, variables, or skolem functions, as well as conjunction and difference of module indicators. However, the latter two cases do not add expressive power, as they can be defined, equivalently, through qualified literal conjunctions and the use of weak negation. Part of the semantics of the RDF(S) vocabulary is represented as pre-defined rules (and not as semantic conditions on interpretations), which are grouped together in a module. The semantics of a modular rule base is defined, based on the *well-founded semantics* (`WFS`) [29] of an equivalent logic program. Yet, the model-theoretic semantics of TRIPLE [68] does not faithfully extend RDFS semantics [42] and is not, in general, equivalent to its transformational semantics. Additionally, TRIPLE does not support explicit negation and ignores visibility issues.

[*F-Logic Modularity Framework*]

Flora-2 [72] is a rule-based object-oriented knowledge base system for reasoning with semantic information on the Web. It is based on F-logic [48] and supports metaprogramming, non-monotonic multiple inheritance, logical database updates, encapsulation, modules with dynamically assigned content, and qualified literals. Module indicators in qualified literals can be module names or variables that get bound to a module name at run time. In Flora-2, strong negation is not supported. Simple literals appearing in a file, that is loaded to a module, are assumed to be qualified by the module name. The semantics of a modular rule base $\mathcal{R}$ is defined, based on the F-logic semantics [48] of an equivalent rule base with no modules. In particular, each qualified atom $subject[predicate \rightarrow object]@Nam_s$ (where Nam_s is a module name) is translated to $subject[predicate\#Nam_s \rightarrow object]$, where $predicate\#Nam_s$ is a new predicate name [46]. However, this work does not have a model-theoretic semantics that faithfully ex-

[26] Our theory faithfully extends RDFS semantics, as it extends the model theory of RDFS and Proposition 9 holds.

tends RDFS semantics. Additionally, it does not support general formulas and qualified ERDF formulas in the bodies of the rules.

[*Modular Web Rule Bases*]

In [2, 4], we presented a principled framework for modular web rule bases, called `MWeb`. According to this framework, users or applications import knowledge about predicates (available on rule bases over the web) into their own rule base. For each predicate defined in a rule base s, four reasoning modes, `definite`, `open`, `closed`, and `normal` are considered, which indicate, respectively, that weak negation is not accepted at all, only open-world assumptions are accepted, both closed-world and open-world assumptions are accepted, and weak negation is fully accepted. When a user or application imports a predicate p, he/she may express that certain reasoning modes on p are not allowed. The producer of a rule base s might also express that a predicate defined in s is (i) allowed to be redefined by other rule bases, (ii) allowed only to be used but not redefined by other rule bases, or (iii) is invisible from other rule bases. We call these predicates `global`, `local`, or `internal` to s, respectively. In summary, each predicate defined in a rule base is characterized by its defining reasoning mode, scope, and exporting rule base list. Each predicate used in a rule base is characterized by its requesting reasoning mode and importing rule base list. For legal `MWeb` modular rule bases $\mathcal{R}$, the `MWebAS` and `MWebWFS` semantics of each rule base $s \in \mathcal{R}$ w.r.t. $\mathcal{R}$ are defined model-theoretically. These semantics extend the *answer set semantics* (`AS`) and the *well-founded semantics with explicit negation* (`WFSX`) on ELPs, respectively, keeping all of their semantical and computational characteristics. The `MWeb` framework supports: (i) local semantics and different points of view, (ii) local closed-world and open-world assumptions, (iii) scoped negation-as-failure, and (iv) restricted propagation of local inconsistencies. However, this work does not have a model-theoretic semantics that faithfully extends RDFS semantics and does not consider ERDF ontologies. Additionally, it does not support general formulas and qualified ERDF formulas in the bodies of the rules. Moreover, the framework described in [2, 4], cannot be used directly for implementing simple modular ERDF ontologies due to the fact that an ERDF triple $p(s, o)$ is represented in logic programming by the atom $H(s, p, o)$. Therefore, the `MWeb` framework has been extended such that it implements simple modular ERDF ontologies and allows (i) the definition of properties and classes in `normal` and `global` mode and (ii) importing/exporting of properties and classes. The exact implementation has been detailed in [22, 21]. More information on this matter is provided in the concluding section.

[*Networked Graphs*]

Networked Graphs [65] are an extension to RDF and to named graphs [19] allowing to assign names to graphs and define other graphs via SPARQL CONSTRUCT queries, which can be recursive. The authors provide a well-founded based semantics to these recursive graphs, and share some ideas with our approach. However, visibility issues are not addressed, recursive views may require the creation of artificial named graphs, and no strong negation is provided. Regarding expressivity, we base our approach in the Answer Set Semantics while Networked graphs resort to Well-founded Semantics. There is a distributed implementation reported in the paper.

[*Description Logic Modularity Frameworks*]

A related direction of research is on modularity frameworks for Description Logic (DL) ontologies [9] (for an overview and qualitative comparison of these frameworks, see [37, 10]). In *E-connections* for DLs [50, 38, 39], a set of DL ontologies $s_1, ..., s_n$ with disjoint vocabularies is connected via a set of *link relations* $\mathcal{E}=\bigcup_{i,j\leq n} E_{i,j}$. Link relations in $E_{i,j}$ are used in s_i for forming the concepts: $\exists E_{i,j}.C_j$, $\forall E_{i,j}.C_j$, $\leq E_{i,j}.C_j$, and $\geq E_{i,j}.C_j$, where C_j is a concept of s_j. However, in E-connections a concept cannot be declared as a subclass of another concept in a foreign module and a property cannot be declared as a sub-relation of a foreign property. Additionally, foreign classes and foreign properties cannot be instantiated. Further, E-connected ontologies do not allow the same term to be used both as a link name and as a local role name, neither allow role inclusions between links and roles.

In *Distributed Description Logics* (DDLs) [16, 66], a set of DL ontologies $s_1, ..., s_n$ with disjoint vocabularies is connected via (i) a set of *bridge rules* of the form $C_i \overset{\sqsubseteq}{\rightarrow} C_j$ or $C_i \overset{\sqsupseteq}{\rightarrow} C_j$, where C_i and C_j are concepts of s_i and s_j, respectively, (ii) a set of *partial individual correspondences* $a_i \mapsto b_j$, where a_i and b_j are objects of s_i and s_j, respectively, and (iii) a set of *complete individual correspondences* $a_i \overset{=}{\mapsto} \{b_j^1, ..., b_j^n\}$, where a_i is an object of s_i and $b_j^1, ..., b_j^n$ are objects of s_j. Subsequent extensions to DDL include homogeneous mappings between roles and heterogeneous mappings between roles and concepts [32, 34, 33, 35]. Yet bridge rules have semantic differences with respect to concept inclusion and while subset relations are transitive, bridge rules cannot be transitively reused by multiple modules. Similarly, to our framework DDLs support directed semantic relations.

In *Package-based Description Logics* (P-DL) [11], a set of DL ontologies $s_1, ..., s_n$ (called *packages*) is connected through the use of common terms. Each term and axiom is associated with a single *home package*. A package s_i may use *foreign terms*, that is terms whose *home* is another package s_j. P-DL uses importing relations to connect local modules. The P-DL importing relation is partial in that only commonly shared terms are interpreted in the overlapping part of local models. The *image domain relation* between local interpretations of different packages (interpreting importing relations) is one-to-one and compositional consistent, ensuring that the parts of local domains connected by the domain relations match perfectly. Additionally, consistency between the interpretations of concepts and roles in their home package and the interpretations in the packages that import them is ensured. P-DL supports inter-module concept inclusion and concept construction using foreign concepts, roles, and nominals, but does not support inter-module role inclusion. Similarly, to our framework P-DL supports directed semantic relations.

In *Integrated Distributed Description Logics* (IDDL) [73] a set of DL ontologies is interconnected by ontology alignments. An ontology alignment is composed of a set of correspondences. A correspondence between two ontologies i and j is one of the following formulas: (i) $i : C \overset{\sqsubseteq}{\longleftrightarrow} j : D$ is a *cross-ontology concept subsumption* (ii) $i : R \overset{\sqsubseteq}{\longleftrightarrow} j : S$ is a *cross-ontology role subsumption* (iii) $i : C \overset{\perp}{\longleftrightarrow} j : D$ is a *cross-ontology concept disjunction* (iv) $i : R \overset{\perp}{\longleftrightarrow} j : S$ is a *cross-ontology role disjunction* (v) $i : a \overset{\in}{\longleftrightarrow} j : C$ is a *cross-ontology membership* (vi) $i : a \overset{=}{\longleftrightarrow} j : b$ is a *cross-ontology identity*. A distributed interpretation assigns a standard DL interpretation to each ontology in the system, as well as an equalizing function that correlate local knowledge into a global domain of interpretation. IDDL has the advantage of being able to compose correspondences. However, in IDDL, mappings are bi-directional, hence if an ontology O_1 wishes to reuse ontology O_2 then O_2 is also affected by O_1.

In [26], an interface-based formalism for modular ontologies is introduced. An interface is a set of concept and role names and their inclusion axioms. A module is an ontology in any description logic language, which can utilize or realize a set of interfaces. A module utilizes an interface if it uses the concepts and roles of the interface in its local knowledge base, according to the TBox axioms of that interface. A module realizes an interface when it provides definitions and more specialized inclusion axioms as well as appropriate ABox assertions for the interface concepts and roles. Different modules and interfaces can be configured to form a meaningful modular ontology, in the sense that for every module which uses an interface, a realizer module is assigned to it. This formalism provides a black-box behavior so that each module which uses an interface makes the assumption that the current knowledge of the realizer module is complete enough for reasoning, hence there is a closed world assumption between different modules in a modular ontology. The formalism uses epistemic queries to allow a utilizer module employ the local knowledge base of the others.

Despite their differences, modular DL ontologies and modular ERDF ontologies share some common goals, such as encapsulation, localized semantics, partial knowledge reuse, and directed semantic relations. Yet, there exist some major differences between the two approaches, as modular DL ontologies deal with the interconnection of DL ontologies, whereas modular ERDF ontologies deal with the interconnection of ERDF ontologies. Although DL ontologies are more expressive than rules in certain aspects [40], rules provide for more expressive forms of module interconnection. Moreover, modular ERDF ontologies provide full support for negation (weak and strong) and all other connectives.

A different kind of integration of extended logic programming with ontologies is achieved in [25]. In this work, a $\mathcal{SHOIN}(\mathbf{D})$ or $\mathcal{SHIF}(\mathbf{D})$ knowledge base L communicates with an extended logic program P (possibly with weak and strong negation), only through DL-query atoms in the body of the rules. In particular, the description logic component L is used for answering the augmented, with input from the logic program, queries appearing in the (possibly weakly negated) DL-query atoms, thus allowing flow of knowledge from P to L and vice-versa. The answer set semantics of $\langle L, P \rangle$ is defined, as a generalization of the answer set semantics [31] on ordinary extended logic programs. Obviously, in this work, derived information concerns only non DL-atoms (that can be possibly used as input to DL-query atoms) and DL ontologies are viewed as black boxes. In contrast, in our work, properties and classes can freely appear in the heads and bodies of the rules, allowing even the derivation of metalevel statements such as subclass and subproperty relationships, property and class totalness. Additionally, in our work, queries to ontologies are not made in a black box fashion but at the model derivation level.

[*Multi-context Reasoning*]

Finally, we would like to mention a general framework for multi-context reasoning, proposed in [17], that allows to combine arbitrary monotonic and non-monotonic logics. Information flow between the different contexts is achieved through a set of non-monotonic bridge rules. In particular, in [17], knowledge bases of different logics (such as Default Logic, normal logic programs under answer set semantics, propositional logics under closed world semantics) are associated with a set of bridge rules that possibly contain weakly negated elements qualified over other knowledge bases. Then, the ground equilibriums for such a system are defined. Though, [17] considers knowledge bases of different logics, our bridge rules (defined as the set of rules of an ontology that contain qualified formulas) are more general since they can contain variables and all connectives $\sim$, $\neg$, $\supset$, $\wedge$, $\vee$, $\forall$, $\exists$. The following definition relates our work with that of [17].

Proposition 17. Let $\mathcal{R}$ be a simple modular ERDF ontology consisting of ontologies that do not contain strong negation and import statements, and export all properties and classes. Let $O \in \mathcal{R}$ and let $D_O^{\mathcal{R}} = \{O_1, ..., O_m\}$. Let KB_i be the knowledge base consisting of (i) the ERDF triples and the `r`-ERDF rules in O_i except these containing qualified ERDF triples and (ii) the partial and ERDF interpretation rules without the first argument in the literals of the rules and instantiated over $V_{O_i,\mathcal{R}}$. Let the bridge rules br_i of KB_i be the rules in O_i that contain qualified ERDF triples instantiated over $V_{O_i,\mathcal{R}}$. We should mention that all ERDF triples $p(s,o)$ in KB_i and br_i should be replaced by the literal $H(s,p,o)$. Let the *multi-context system* $W = (C_1, ..., C_m)$, where $C_i = (NLP, KB_i, br_i)$ and NLP is the logic of normal logic programs under stable model semantics [30]. Then, the following hold:

1. If $S = (S_1, ..., S_m)$ is a *belief state* of W according to [17] then
$$ELP^{-1}(\{H(Nam_{O_i}, s, p, o) \mid H(s,p,o) \in S_i \text{ and } i \in \{1, ..., m\}\}) \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}.$$
2. If $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ then $S = (S_1, ..., S_m)$ is a *belief state* of W according to [17], where
$$S_i = \{H(s,p,o) \mid H(Nam_{O_i}, s, p, o) \in ELP(\mathsf{M})\}.$$

The proof of Proposition 17 is based on Proposition 10.

9 Conclusions

In this paper, we extended ERDF ontologies [6], and thus RDF graphs to ontologies. In particular, an ontology is an ERDF ontology that (i) is associated with a set of export and import statements, and (ii) interacts with other ontologies (through qualified ERDF formulas in the program rules). Further, we defined a modular ERDF ontology as a set of ontologies and defined its modular stable model semantics, model-theoretically, based on partial logic [43]. We showed that modular stable model entailment on modular ERDF ontologies extends $\#n$-stable model entailment on ERDF ontologies [7], and thus it also extends RDFS entailment on RDF graphs [42].

Our modular ERDF framework enables collaborative reasoning over a modular ERDF ontology $\mathcal{R}$, while support for hidden knowledge is also provided. Additionally, it supports:

- local semantics and different points of view, as different ontologies have their own semantics and can entail even contradictory facts. When knowledge from one ontology is imported to another ontology, the interpretation of the reused knowledge is constrained by the ontology in which the knowledge is being reused,
- local closed-world assumptions, as closed-world assumptions of the form $\neg p(?x, ?y) \leftarrow F \wedge \sim p(?x,\ ?y)$ and $\neg$ `rdf:type`$(?x, c) \leftarrow F \wedge \sim$ `rdf:type`$(?x,\ c)$, where F is an formula, can appear inside an ontology O and thus applied to terms in $V_{O,\mathcal{R}}$,
- local open-world assumptions, as open-world assumptions of the form `rdf:type`(p, `erdf:TotalProperty`) and `rdf:type`(c,`erdf:TotalClass`) can appear inside an ontology O and thus applied to terms in $V_{O,\mathcal{R}}$,
- scoped negation-as-failure, as a formula $\sim F@Nam_O$ can appear in the body of a rule of an ontology O', where F is an ERDF formula and O is an ontology. Thus, the formula $\sim F@Nam_O$ succeeds if F cannot be derived from the ontology O,
- directed semantic relations, as if an ontology $O \in \mathcal{R}$ imports information from another ontology $O' \in \mathcal{R}$ then this affects reasoning in O but not in O', and
- restricted propagation of local inconsistencies, as the semantics of an ontology $O \in \mathcal{R}$ depends only on the ontologies in $D^{\mathcal{R}}_O$ and not on all ontologies in $\mathcal{R}$.

A mapping from simple modular ERDF ontologies[27] into the `MWeb` system [2, 4] has been detailed in [22, 21]. The semantics of simple modular ERDF ontologies is totally captured by `MWeb` rule bases using a syntax aligned with the recent W3C recommendation Rule Interchange Format [13, 24]. By expressing simple modular ERDF ontologies in the

[27] A *simple modular ERDF ontology* is a modular ERDF ontology $\mathcal{R}$ s.t. for all $O \in \mathcal{R}$, it holds that P_O is simple.

MWeb system, the existing MWeb implementation[28] can be used to perform query answering on simple ontologies w.r.t. a modular ERDF ontology. The existing implementation compiles simple modular ERDF ontologies into XSB Prolog's modules [63] supporting both the well-founded and answer-set based semantics. The answer-set based semantics relies on XSB's library interface to the answer-set solver Smodels [55]. Moreover, preliminary testing indicates that this is one of the fastest implementations for memory-based (E)RDF inferencing due to its use of (subsumptive) tabling features of the underlying XSB Prolog engine; these good practical results are also supported by the underlying good performance of XSB, as described in [53]. In particular, we have performed a first comparison using the W3C Wine ontology, determining CPU times for RDFS inference in XSB 3.2 without equality reasoning (discarding loading and compile times). Briefly, Jena's (2.6.2) inbuilt RDFS Reasoner was 2 times slower than our MWeb implementation, while Jena's Generic reasoner was 100 times slower. Euler Yap (Eye 3414) shown to be 4 times slower and CWM-1.2.1 was 100 times slower.

The modular ERDF framework proposes the complete separation of the interface part, which can be freely exchanged in the Web, containing the `export` and `import` declarations, and the associated logic program, containing the predicate definitions, which might be private (sensible data, etc.). However, it is outside the scope of the paper, how these mechanisms must be implemented in practice with the full generality required by the Semantic Web. In fact, trust and authorization could be much improved by security languages, such as the PeerTrust language [28].

Due to the Semantic Web's decentralized and distributed nature, contradictory information is frequent [64]. Future work concerns the extension of the modular stable model semantics such that meaning is assigned to inconsistent ontologies of a modular ERDF ontology.

Our future work also concerns the support of datatypes, including XSD datatypes, and the extension of the predefined ERDF vocabulary by adding other useful constructs, possibly in accordance with the extensions of ter Horst [70].

Acknowledgments: The authors would like to thank the anonymous referees for their valuable comments.

[28] Available at `http://centria.di.fct.unl.pt/~cd/mweb`

Appendix A: Review of the ERDF $\#n$-Stable Model Semantics

In this appendix, we review for self-containment the basic definitions of the ERDF $\#n$-stable model semantics, provided in [7, 5].

An *ERDF rule* r over a vocabulary V is an expression of the form: $Concl(r) \leftarrow Cond(r)$, where $Cond(r) \in L(V) \cup \{true\}$ and $Concl(r)$ is an ERDF triple or *false*. We assume that no bound variable in $Cond(r)$ appears free in $Concl(r)$. We denote the set of variables and the set of free variables of r by $Var(r)$ and $FVar(r)$[29], respectively. An *ERDF program* P is a set of ERDF rules. We denote the set of URI references and literals appearing in P by V_P.

An *ERDF ontology* is a pair $O = \langle G, P \rangle$, where G is an ERDF graph and P is an ERDF program.

Definition 18. (Satisfaction of an ERDF formula w.r.t. a partial interpretation and a valuation) Let F, G be ERDF formulas and let I be a partial interpretation of a vocabulary V. Additionally, let v be a mapping $v : Var(F) \rightarrow Res_I$.

- If $F = p(s, o)$ then $I, v \models F$ iff $p \in V \cap URI$, $s, o \in V \cup Var$, $I(p) \in Prop_I$, and $\langle [I + v](s), [I + v](o) \rangle \in PT_I(I(p))$.
- If $F = \neg p(s, o)$ then $I, v \models F$ iff $p \in V \cap URI$, $s, o \in V \cup Var$, $I(p) \in Prop_I$, and $\langle [I + v](s), [I + v](o) \rangle \in PF_I(I(p))$.
- If $F = {\sim}G$ then $I, v \models F$ iff $V_G \subseteq V$ and $I, v \not\models G$.
- If $F = F_1 \wedge F_2$ then $I, v \models F$ iff $I, v \models F_1$ and $I, v \models F_2$.
- If $F = F_1 \vee F_2$ then $I, v \models F$ iff $I, v \models F_1$ or $I, v \models F_2$.
- If $F = F_1 \supset F_2$ then $I, v \models F$ iff $I, v \models {\sim}F_1 \vee F_2$.
- If $F = \exists x\ G$ then $I, v \models F$ iff there exists mapping $u : Var(G) \rightarrow Res_I$ such that $u(y) = v(y)$, $\forall y \in Var(G) - \{x\}$, and $I, u \models G$.
- If $F = \forall x\ G$ then $I, v \models F$ iff for all mappings $u : Var(G) \rightarrow Res_I$ such that $u(y) = v(y)$, $\forall y \in Var(G) - \{x\}$, it holds $I, u \models G$.
- All other cases of ERDF formulas are treated by the following DeMorgan-style rewrite rules expressing the falsification of compound ERDF formulas: $\neg(F_1 \wedge F_2) \rightarrow \neg F_1 \vee \neg F_2$, $\neg(F_1 \vee F_2) \rightarrow \neg F_1 \wedge \neg F_2$, $\neg(\neg G) \rightarrow G$, $\neg({\sim}\, G) \rightarrow G$ $\neg(\exists x\ G) \rightarrow \forall x\ \neg G$, $\neg(\forall x\ G) \rightarrow \exists x\ \neg G$, $\neg(F_1 \supset F_2) \rightarrow F_1 \wedge \neg F_2$. $\square$

Let F be an ERDF formula, let G be an ERDF graph, and let I be a partial interpretation of a vocabulary V. We define: $I \models F$ iff for each mapping $v : Var(F) \rightarrow Res_I$, it holds that $I, v \models F$. Additionally, we define: $I \models G$ iff $I \models formula(G)$.

We assume that for every partial interpretation I, it holds that $I \models true$ and $I \not\models false$.

The *$\#n$-vocabulary* of an ERDF ontology O is defined as $V_O^{\#n} = V_{sk(G)} \cup V_P \cup \mathcal{V}_{RDF}^{\#n} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$. Additionally, we denote by $Res_O^{H_{\#n}}$ the union of $V_O^{\#n}$ and the set of XML values of the well-typed XML literals in $V_O^{\#n}$ minus the well-typed XML literals.

Definition 19 ($\#n$-Herbrand interpretation). Let $O = \langle G, P \rangle$ be an ERDF ontology. An *$\#n$-Herbrand interpretation* I of O is an ERDF $\#n$-interpretation of $V_O^{\#n}$ such that: (i) $Res_I = Res_O^{H_{\#n}}$, (ii) $I_V(x) = x$, for all $x \in V_O^{\#n} \cap URI$, (iii) $IL_I(x) = x$, if x is a typed literal in $V_O^{\#n}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_O^{\#n}$. We denote the set of $\#n$-Herbrand interpretations of O by $\mathcal{I}^{H_{\#n}}(O)$. $\square$

[29] $FVar(r) = FVar(F) \cup FVar(G)$.

Let $I, J \in \mathcal{I}^{H_{\#n}}(O)$. We say that J *extends* I, denoted by $I \leq J$, iff $Prop_I \subseteq Prop_J$, and $\forall\ p \in Prop_I,\ \ PT_I(p) \subseteq PT_J(p)$ and $PF_I(p) \subseteq PF_J(p)$. Let $\mathcal{I} \subseteq \mathcal{I}^{H_{\#n}}(O)$. We define $minimal(\mathcal{I}) = \{I \in \mathcal{I} \mid\ \nexists J \in \mathcal{I}:\ J \neq I$ and $J \leq I\}$. Let $I, J \in \mathcal{I}^{H_{\#n}}(O)$. We define $[I, J]_O^{\#n} = \{I' \in \mathcal{I}^{H_{\#n}}(O) \mid I \leq I' \leq J\}$.

Let V be a vocabulary and let r be an ERDF rule. We denote by $[r]_V$ the set of rules that result from r if we replace each variable $x \in FVar(r)$ by $v(x)$, for all mappings $v : FVar(r) \rightarrow V$. Let P be an ERDF program. We define $[P]_V = \bigcup_{r \in P} [r]_V$.

Below, we define the stable models of an ERDF ontology, based on the coherent stable models of Partial Logic [43].

Definition 20 (ERDF $\#n$-stable model). Let $O = \langle G, P \rangle$ be an ERDF ontology and let $M \in \mathcal{I}^{H_{\#n}}(O)$. We say that M is an *(ERDF) $\#n$-stable model* of O iff there is a chain of $\#n$-Herbrand interpretations of O, $I_0 \leq ... \leq I_{k+1}$, such that $I_k = I_{k+1} = M$ and:

1. $I_0 \in minimal(\{I \in \mathcal{I}^{H_{\#n}}(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k+1$:
   $I_\alpha \in minimal(\{I \in \mathcal{I}^{H_{\#n}}(O) \mid I \geq I_{\alpha-1}$ and it holds that:
   $\forall\ r \in [P]_{V_O^{\#n}}$, if $J \models Cond(r),\ \forall J \in [I_{\alpha-1}, M]_O^{\#n}$, then $I \models Concl(r)\})$.

The set of stable models of O is denoted by $\mathcal{M}^{st_{\#n}}(O)$. $\square$

Let $O = \langle G, P \rangle$ be an ERDF ontology and let F be an ERDF formula or ERDF graph. Let $n \in I\!N$. We say that O *entails* F under the *(ERDF) $\#n$-stable model semantics*, denoted by $O \models^{st_{\#n}} F$ iff for all $M \in \mathcal{M}^{st_{\#n}}(O)$, $\ M \models F$.

Appendix B: Proofs

In this Appendix, we provide proofs for all propositions presented in the main paper. To reduce the size of the proofs, we have eliminated the namespace from the URIs in $\mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$.

Proposition 1 Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} \mid O' \in D_O^{\mathcal{R}}\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. Let F_1, F_2 be ERDF formulas and $O'' \in D_O^{\mathcal{R}}$. It holds that:

1. $\langle \mathsf{I}, O' \rangle \models (F_1 \wedge F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O' \rangle \models F_1@Nam_{O''} \wedge F_2@Nam_{O''}$.
2. $\langle \mathsf{I}, O' \rangle \models (F_1 \vee F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O' \rangle \models F_1@Nam_{O''} \vee F_2@Nam_{O''}$.

Proof:
1. Let mapping $v : Var(F_1 \wedge F_2) \rightarrow Res_{I_{O'}}$. $\langle \mathsf{I}, O', v \rangle \models (F_1 \wedge F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O'', v \rangle \models F_1 \wedge F_2$ iff $\langle \mathsf{I}, O'', v \rangle \models F_1$ and $\langle \mathsf{I}, O'', v \rangle \models F_2$ iff $\langle \mathsf{I}, O', v \rangle \models F_1@Nam_{O''}$ and $\langle \mathsf{I}, O', v \rangle \models F_2@Nam_{O''}$ iff $\langle \mathsf{I}, O', v \rangle \models F_1@Nam_{O''} \wedge F_2@Nam_{O''}$.

2. Let mapping $v : Var(F_1 \vee F_2) \rightarrow Res_{I_{O'}}$. $\langle \mathsf{I}, O', v \rangle \models (F_1 \vee F_2)@Nam_{O''}$ iff $\langle \mathsf{I}, O'', v \rangle \models F_1 \vee F_2$ iff $\langle \mathsf{I}, O'', v \rangle \models F_1$ or $\langle \mathsf{I}, O'', v \rangle \models F_2$ iff $\langle \mathsf{I}, O', v \rangle \models F_1@Nam_{O''}$ or $\langle \mathsf{I}, O', v \rangle \models F_2@Nam_{O''}$ iff $\langle \mathsf{I}, O', v \rangle \models F_1@Nam_{O''} \vee F_2@Nam_{O''}$. $\square$

Proposition 2 Let $O \in \mathcal{R}$. Let $\mathsf{I} = \{I_{O'} \mid O' \in D_O^{\mathcal{R}}\}$ be a modular ERDF interpretation of O w.r.t. $\mathcal{R}$. Additionally, let $O', O'' \in D_O^{\mathcal{R}}$. It holds that: $\langle \mathsf{I}, O' \rangle \models (\sim p(s,o))@Nam_{O''}$ iff $\langle \mathsf{I}, O' \rangle \models \sim p(s,o)@Nam_{O''}$.

Proof:
Let $v : \{\} \rightarrow Res_{I_{O'}}$. $\langle \mathsf{I}, O', v \rangle \models (\sim p(s,o))@Nam_{O''}$ iff $\langle \mathsf{I}, O'', v \rangle \models \sim p(s,o)$ iff $\langle \mathsf{I}, O'', v \rangle \not\models p(s,o)$ iff $\langle \mathsf{I}, O', v \rangle \not\models p(s,o)@Nam_{O''}$ iff $\langle \mathsf{I}, O', v \rangle \models \sim p(s,o)@Nam_{O''}$. $\square$

Proposition 3 Let $O \in \mathcal{R}$. If M is a modular Herbrand model of O w.r.t. $\mathcal{R}$ then M is a modular model of O w.r.t. $\mathcal{R}$.

Proof: Since M is a modular Herbrand model of O w.r.t. $\mathcal{R}$, it follows that for all $O' \in D_O^{\mathcal{R}}$, $\langle \mathsf{I}, O' \rangle \models sk(G_{O'})$. It is enough to show that for all $O' \in D_O^{\mathcal{R}}$, $\langle \mathsf{I}, O' \rangle \models G_{O'}$. Let $O' \in D_O^{\mathcal{R}}$. Let $formula(G_{O'}) = \exists ?x_1, ..., \exists ?x_k \; t_1 \wedge ... \wedge t_m$. Additionally, we define a total function $u : Var(G_{O'}) \rightarrow Res^{\mathtt{H}}_{O',\mathcal{R}}$ s.t. $u(x) = sk_{G_{O'}}(x), \forall x \in Var(G_{O'})$. Moreover, we define a total function $u' : V_{O',\mathcal{R}} \cup Var(G_{O'}) \rightarrow V_{O',\mathcal{R}}$ s.t. $u'(x) = sk_{G_{O'}}(x)$, if $x \in Var(G_{O'})$ and $u'(x) = x$, otherwise. We will show that $\langle \mathsf{I}, O', u \rangle \models t_1 \wedge ... \wedge t_m$. Let $p(s, o) \in G_{O'}$. Then, $p \in V_{O',\mathcal{R}}$, $s, o \in V_{O',\mathcal{R}} \cup Var$, and $I_{O'}(p) \in Prop_{I_{O'}}$. It holds: $\langle [I_{O'} + u](s), [I_{O'} + u](o) \rangle \in PT_{I_{O'}}(I_{O'}(p))$ iff $\langle I_{O'}(u'(s)), I_{O'}(u'(o)) \rangle \in Prop_{I_{O'}}(I_{O'}(p))$, which is true, since $p(u'(s), u'(o)) \in sk(G_{O'})$ and $\langle \mathsf{I}, O' \rangle \models sk(G_{O'})$. Thus, $\langle \mathsf{I}, O', u \rangle \models t_1 \wedge ... \wedge t_m$.

We want to show that $\langle \mathsf{I}, O' \rangle \models G_{O'}$. Let $v : Var(formula(G_{O'})) \rightarrow Res^{\mathtt{H}}_{O',\mathcal{R}}$. It is enough to show that $\langle \mathsf{I}, O', v \rangle \models formula(G_{O'})$.

It holds that $\langle \mathsf{I}, O', v \rangle \models \exists ?x_1, ..., \exists ?x_k \; t_1 \wedge ... \wedge t_m$ iff there exists a mapping $u_1 : Var(t_1 \wedge ... \wedge t_m) \rightarrow Res^{\mathtt{H}}_{O',\mathcal{R}}$ such that $u_1(y) = v(y)$, $\forall y \in Var(t_1 \wedge ... \wedge t_m) - \{x_1\}$, and $\langle \mathsf{I}, O', u_1 \rangle \models \exists ?x_2, ..., \exists ?x_k \; t_1 \wedge ... \wedge t_m$ iff ... iff there exists a mapping $u_k : Var(t_1 \wedge ... \wedge t_m) \rightarrow Res^{\mathtt{H}}_{O',\mathcal{R}}$ such that $u_k(y) = u_{k-1}(y)$, $\forall y \in Var(t_1 \wedge ... \wedge t_m) - \{x_k\}$, and $\langle \mathsf{I}, O', u_k \rangle \models t_1 \wedge ... \wedge t_m$. Let $u_i(x_i) = sk_{G_{O'}}(x_i)$, for $i = 1, ...k$. Then, $u_k = u$ and the last statement holds. Thus, $\langle \mathsf{I}, O', v \rangle \models formula(G_{O'})$. Therefore, $\langle \mathsf{I}, O' \rangle \models G_{O'}$. $\Box$

Proposition 4 Let $O \in \mathcal{R}$. If $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ then $\mathsf{M} \in \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$.

Proof: Let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. Obviously, $\mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$ and $\forall O' \in D_O^{\mathcal{R}}$, $\langle \mathsf{M}, O' \rangle \models sk(G_{O'})$. We will show that for all $O' \in D_O^{\mathcal{R}}$, and for all $r \in P_{O'}$, $\langle \mathsf{M}, O' \rangle \models r$. Let $O' \in D_O^{\mathcal{R}}$ and let $r \in P_{O'}$. Let v be a mapping $v : Var(r) \rightarrow Res^{\mathtt{H}}_{O',\mathcal{R}}$ s.t. $\langle \mathsf{M}, O', v \rangle \models Cond(r)$. It is enough to show that $\langle \mathsf{M}, O', v \rangle \models Concl(r)$.

Let Res be a set. For any mapping $u : X \rightarrow Res$, where $X \subseteq Var$, we define the mapping $u^* : X \rightarrow Res \cup \mathcal{PL} \cup \mathcal{TL}$ as follows:

$$u^*(x) = \begin{cases} u(x) & \text{if } u(x) \text{ is not the xml value of a well-typed XML literal} \\ t & \text{if } u(x) \text{ is the xml value of a well-typed XML literal } t \end{cases}$$

Let $x \in V_{O',\mathcal{R}}$, we define $x^{u^*} = x$. Let $x \in X$, we define $x^{u^*} = u^*(x)$. Let F be an $\mathtt{r}$-ERDF formula over $V_{O',\mathcal{R}}$ and $\{Nam_{O''} | \; O'' \in D_{O'}^{\mathcal{R}}\}$, $\mathtt{true}$, or $\mathtt{false}$ such that $FVar(F) \subseteq X$, we define F^{u^*} to be the formula that results from F after replacing each free variable x of F by $u^*(x)$. It is easy to see that it holds: $Concl(r)^{v^*} \leftarrow Cond(r)^{v^*} \in [r]_{V_{O',\mathcal{R}}} \subseteq [P_{O'}]_{V_{O',\mathcal{R}}}$.

Lemma: Let $O' \in D_O^{\mathcal{R}}$ and let Res be a set. Let F be an $\mathtt{r}$-ERDF formula over $V_{O',\mathcal{R}}$ and $\{Nam_{O''} | \; O'' \in D_{O'}^{\mathcal{R}}\}$ and let u be a mapping $u : Var(F) \rightarrow Res$. It holds that: $\langle \mathsf{M}, O', u \rangle \models F$ iff $\langle \mathsf{M}, O', u \rangle \models F^{u^*}$.
Proof: We prove the lemma by induction on the length of the formula F. Without loss of generality, we assume that $\neg$ appears only in front of positive ERDF triples. Otherwise we apply the transformation rules of Definition 10, to get an equivalent formula that satisfies the assumption.

Let $F = p(s, o)$. It holds that: $\langle \mathsf{M}, O', u \rangle \models F$ iff $\langle \mathsf{M}, O', u \rangle \models p(s, o)$ iff $\langle [M_{O'} + u](s), [M_{O'} + u](o) \rangle \in PT_{M_{O'}}(M_{O'}(p))$ iff $\langle [M_{O'} + u](s^{u^*}), [M_{O'} + u](o^{u^*}) \rangle \in PT_{M_{O'}}(M_{O'}(p))$ iff $\langle \mathsf{M}, O', u \rangle \models p(s, o)^{u^*}$.

Let $F = \neg p(s, o)$. It holds that: $\langle \mathsf{M}, O', u \rangle \models F$ iff $\langle \mathsf{M}, O', u \rangle \models p(s, o)$ iff $\langle [M_{O'} + u](s), [M_{O'} + u](o) \rangle \in PF_{M_{O'}}(M_{O'}(p))$ iff $\langle [M_{O'} + u](s^{u^*}), [M_{O'} + u](o^{u^*}) \rangle \in PF_{M_{O'}}(M_{O'}(p))$ iff $\langle \mathsf{M}, O', u \rangle \models (\neg p(s, o))^{u^*}$.
Assumption: Assume that the lemma holds for the $\mathtt{r}$-ERDF formulas of length smaller than F.

We will show that the lemma holds also for F.

Let $F = \sim G$. It holds that: $\langle \mathsf{M}, O', u\rangle \models F$ iff $\langle \mathsf{M}, O', u\rangle \models \sim G$ iff $\langle \mathsf{M}, O', u\rangle \not\models G$ iff $\langle \mathsf{M}, O', u\rangle \not\models G^{u^*}$ iff $\langle \mathsf{M}, O', u\rangle \models \sim G^{u^*}$ iff $\langle \mathsf{M}, O', u\rangle \models F^{u^*}$.

Let $F = F_1 \wedge F_2$. It holds that: $\langle \mathsf{M}, O', u\rangle \models F$ iff $\langle \mathsf{M}, O', u\rangle \models F_1 \wedge F_2$ iff $\langle \mathsf{M}, O', u\rangle \models F_1$ and $\langle \mathsf{M}, O', u\rangle \models F_2$ iff $\langle \mathsf{M}, O', u\rangle \models F_1^{u^*}$ and $\langle \mathsf{M}, O', u\rangle \models F_2^{u^*}$ iff $\langle \mathsf{M}, O', u\rangle \models (F_1 \wedge F_2)^{u^*}$ iff $\langle \mathsf{M}, O', u\rangle \models F^{u^*}$.

Let $F = \exists x G$. It holds that: $\langle \mathsf{M}, O', u\rangle \models F \Rightarrow \langle \mathsf{M}, O', u\rangle \models \exists x G \Rightarrow$ there exists a mapping u_1 s.t. $u_1(y) = u(y),\ \forall y \in \mathit{Var}(G) - \{x\}$, $u_1(x) \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$, and $\langle \mathsf{M}, O', u_1\rangle \models G$ $\Rightarrow$ there exists a mapping u_1 s.t. $u_1(y) = u(y),\ \forall y \in \mathit{Var}(G) - \{x\}$, $u_1(x) \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$, $\langle \mathsf{M}, O', u_1\rangle \models G^{u_1^*} \Rightarrow \langle \mathsf{M}, O', u\rangle \models (\exists x G)^{u_1^*} \Rightarrow$ (since $u_1^*(y) = u^*(y),\ \forall y \in \mathit{FVar}(\exists x G)$ $\langle \mathsf{M}, O', u\rangle \models (\exists x G)^{u^*} \Rightarrow \langle \mathsf{M}, O', u\rangle \models F^{u^*}$.

It holds that: $\langle \mathsf{M}, O', u\rangle \models F^{u^*} \Rightarrow \langle \mathsf{M}, O', u\rangle \models (\exists x G)^{u^*} \Rightarrow$ there exists a mapping u_1 s.t. $u_1(y) = u(y),\ \forall y \in \mathit{Var}(G) - \{x\}$, $u_1(x) \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$, $\langle \mathsf{M}, O', u_1\rangle \models G^{u_1^*} \Rightarrow$ there exists a mapping u_1 s.t. $u_1(y) = u(y),\ \forall y \in \mathit{Var}(G) - \{x\}$, $u_1(x) \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$, $\langle \mathsf{M}, O', u_1\rangle \models G$ $\Rightarrow \langle \mathsf{M}, O', u\rangle \models \exists x G \Rightarrow \langle \mathsf{M}, O', u\rangle \models F$.

Let $F = F'@\mathit{Nam}_{O''}$. It holds that: $\langle \mathsf{M}, O', u\rangle \models F$ iff $\langle \mathsf{M}, O', u\rangle \models F'@\mathit{Nam}_{O''}$ iff $\langle \mathsf{M}, O'', u\rangle \models F'$ iff $\langle \mathsf{M}, O'', u\rangle \models (F')^{u*}$ iff $\langle \mathsf{M}, O', u\rangle \models (F')^{u*}@\mathit{Nam}_{O''}$ iff $\langle \mathsf{M}, O', u\rangle \models F^{u^*}$.

Let $F = F_1 \vee F_2$ or $F = F_1 \supset F_2$ or $F = \forall x G$. We can prove, similarly to the above cases, that $\langle \mathsf{M}, O', u\rangle \models F$ iff $\langle \mathsf{M}, O', u\rangle \models F^{u^*}$.

End of Lemma

First assume that $\mathit{Cond}(r) \neq \mathtt{true}$. Then, $\mathit{Cond}(r)$ is an r-ERDF formula over $V_{O',\mathcal{R}}$ and $\{\mathit{Nam}_{O''} |\ O'' \in D^{\mathcal{R}}_{O'}\}$. Since $\langle \mathsf{M}, O', v\rangle \models \mathit{Cond}(r)$, it follows from Lemma that $\langle \mathsf{M}, O', v\rangle \models \mathit{Cond}(r)^{v^*}$. Now since $\mathit{FVar}(\mathit{Cond}(r)^{v^*}) = \emptyset$, it follows that $\langle \mathsf{M}, O'\rangle \models \mathit{Cond}(r)^{v^*}$. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$, it follows that $\langle \mathsf{M}, O'\rangle \models \mathit{Concl}(r)^{v^*}$. Thus, $\mathit{Concl}(r) \neq \mathtt{false}$ and $\mathit{Concl}(r)$ is an ERDF triple over $V_{O',\mathcal{R}}$. Now since $\mathit{FVar}(\mathit{Concl}(r)^{v^*}) = \emptyset$, it follows that $\langle \mathsf{M}, O', v\rangle \models \mathit{Concl}(r)^{v^*}$. It now follows from Lemma that $\langle \mathsf{M}, O', v\rangle \models \mathit{Concl}(r)$.

Assume now that $\mathit{Cond}(r) = \mathtt{true}$. Then, $\langle \mathsf{M}, O'\rangle \models \mathit{Cond}(r)^{v^*}$. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$, it follows that $\langle \mathsf{M}, O'\rangle \models \mathit{Concl}(r)^{v^*}$. Therefore, $\mathit{Concl}(r) \neq \mathtt{false}$, and we can prove as above that $\langle \mathsf{M}, O', v\rangle \models \mathit{Concl}(r)$. Therefore, for all $r \in P_{O'}$, $\langle \mathsf{M}, O'\rangle \models r$. $\Box$

Proposition 5 Let $O \in \mathcal{R}$. If `rdfs:subclass(rdf:Property, erdf:TotalProperty)` $\in G_{O'}$, for all $O' \in D^{\mathcal{R}}_{O}$, then $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}} = \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$.

Proof: From Proposition 4, it follows that $\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}} \subseteq \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$. We will show that $\mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}} \subseteq \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. Let $\mathsf{M} \in \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$. It follows that $\forall O' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{M}, O'\rangle \models \mathit{sk}(G_{O'})$. We will show that $\mathsf{M} \in \mathit{minimal}(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid\ \forall O' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{I}, O'\rangle \models \mathit{sk}(G_{O'})\})$.

Let $\mathsf{J} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$ s.t. $\forall O' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{J}, O'\rangle \models \mathit{sk}(G_{O'})$ and $\mathsf{J} \leq \mathsf{M}$. We will show that $\mathsf{J} = \mathsf{M}$. Since $\mathsf{J} \leq \mathsf{M}$, it follows that $\forall O' \in D^{\mathcal{R}}_{O}$, $\mathit{Prop}_{J_{O'}} \subseteq \mathit{Prop}_{M_{O'}}$ and for all $p \in \mathit{Prop}_{J_{O'}}$, it holds $\mathit{PT}_{J_{O'}}(p) \subseteq \mathit{PT}_{M_{O'}}(p)$ and $\mathit{PF}_{J_{O'}}(p) \subseteq \mathit{PF}_{M_{O'}}(p)$. Let $O' \in D^{\mathcal{R}}_{O}$ and let $p \in \mathit{Prop}_{J_{O'}}$. Since $\langle \mathsf{J}, O'\rangle \models \mathit{sk}(G_{O'})$, it follows that $\mathit{Prop}_{J_{O'}} \subseteq \mathit{TProp}_{J_{O'}}$. Thus, $p \in \mathit{TProp}_{J_{O'}}$. Assume that $\mathit{PT}_{J_{O'}}(p) \neq \mathit{PT}_{M_{O'}}(p)$. Then, there is $\langle x, y\rangle \in \mathit{PT}_{M_{O'}}(p)$ s.t. $\langle x, y\rangle \notin \mathit{PT}_{J_{O'}}(p)$. Then, $\langle x, y\rangle \in \mathit{PF}_{J_{O'}}(p)$. Thus, $\langle x, y\rangle \in \mathit{PF}_{M_{O'}}(p)$, which is impossible, since $\langle x, y\rangle \in \mathit{PT}_{M_{O'}}(p)$. Thus, $\mathit{PT}_{J_{O'}}(p) = \mathit{PT}_{M_{O'}}(p)$. Similarly, we can prove that $\mathit{PF}_{J_{O'}}(p) = \mathit{PF}_{M_{O'}}(p)$. Therefore, for all $p \in \mathit{Prop}_{J_{O'}}$, it holds $\mathit{PT}_{J_{O'}}(p) = \mathit{PT}_{M_{O'}}(p)$ and $\mathit{PF}_{J_{O'}}(p) = \mathit{PF}_{M_{O'}}(p)$. We will now show that $\mathit{Prop}_{J_{O'}} = \mathit{Prop}_{M_{O'}}$. It holds $\mathit{Prop}_{J_{O'}} = \{x \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}} \mid \langle x, \mathtt{Property}\rangle \in \mathit{PT}_{J_{O'}}(\mathtt{type})\} = \{x \in \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}} \mid \langle x, \mathtt{Property}\rangle \in \mathit{PT}_{M_{O'}}(\mathtt{type})\} = \mathit{Prop}_{M_{O'}}$. Based on these results and the fact that $\mathsf{J}, \mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $\mathsf{J} = \mathsf{M}$. Therefore, $\mathsf{M} \in \mathit{minimal}(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{I}, O'\rangle \models \mathit{sk}(G_{O'})\})$.

We will now show that $\mathsf{M} \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{M}$ and $\forall O' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{M}, O' \rangle \models Cond(r)$ then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $\mathsf{M} \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{M}$ and $\forall O' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{M}, O' \rangle \models Cond(r)$ then $\langle \mathsf{I}, O' \rangle \models Concl(r)\}$. Let $\mathsf{J} \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{M}$ and $\forall O' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{M}, O' \rangle \models Cond(r)$ then $\langle \mathsf{I}, O' \rangle \models Concl(r)\}$ and $\mathsf{J} \leq \mathsf{M}$. Since $\mathsf{J} \geq \mathsf{M}$, it follows that $\forall O' \in D^{\mathcal{R}}_{O}$, $Prop_{M_{O'}} \subseteq Prop_{J_{O'}}$, and for all $p \in Prop_{M_{O'}}$, it holds $PT_{M_{O'}}(p) \subseteq PT_{J_{O'}}(p)$ and $PF_{M_{O'}}(p) \subseteq PF_{J_{O'}}(p)$. Since $\mathsf{J} \leq \mathsf{M}$, it follows that $Prop_{J_{O'}} \subseteq Prop_{M_{O'}}$, and for all $p \in Prop_{J_{O'}}$, it holds $PT_{J_{O'}}(p) \subseteq PT_{M_{O'}}(p)$ and $PF_{J_{O'}}(p) \subseteq PF_{M_{O'}}(p)$. Therefore, it follows that $\forall O' \in D^{\mathcal{R}}_{O}$, $Prop_{M_{O'}} = Prop_{J_{O'}}$, and for all $p \in Prop_{M_{O'}}$, it holds $PT_{M_{O'}}(p) = PT_{J_{O'}}(p)$ and $PF_{M_{O'}}(p) = PF_{J_{O'}}(p)$. Based on this result and the fact that $\mathsf{J}, \mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $\mathsf{J} = \mathsf{M}$.

Thus, $M \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{M}$ and $\forall O' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{M}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

Since M satisfies the conditions of Definition 14 (Modular Stable Model), it follows that $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. Thus, it holds $\mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}} \subseteq \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$.

Therefore, $\mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}} = \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. □

Proposition 6 Let $O \in \mathcal{R}$, and let $O' \in D^{\mathcal{R}}_{O}$. Let $\mathsf{M} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$ and let $\mathsf{M}' = \{M_{O''} \in \mathsf{M} \mid O'' \in D^{\mathcal{R}}_{O'}\}$. It holds that: If $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$.

Proof: Let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. Then, there is a chain of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_k$ such that $\mathsf{I}_{k-1} = \mathsf{I}_k = \mathsf{M}$ and:

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O}$, it holds that:
   if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\})$.

We define: $\mathsf{I}'_\alpha = \{I_{\alpha O''} \in \mathsf{I}_\alpha \mid O'' \in D^{\mathcal{R}}_{O'}\}$.

Lemma: It holds that:

1. $\mathsf{I}'_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O'},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}'_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}'_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O'}$, it holds that:
   if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\})$.

Proof of Lemma: It holds that $\mathsf{I}'_0 \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O'},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\}$, we will show that $\mathsf{I}'_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O'},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\})$. Assume that it exists $\mathsf{Z} \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O'},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\}$ and $\mathsf{Z} < \mathsf{I}'_0$. Then, let $\mathsf{Z}' = \{Z_{O''} \in \mathsf{Z} \mid O'' \in D^{\mathcal{R}}_{O'}\} \cup \{I_{0O''} \in \mathsf{I}_0 \mid O'' \in D^{\mathcal{R}}_{O} - D^{\mathcal{R}}_{O'}\}$. Then, $\mathsf{Z}' \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\})$ and $\mathsf{Z}' < \mathsf{I}_0$, which is impossible. Thus, $\mathsf{I}'_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \forall O'' \in D^{\mathcal{R}}_{O'},\ \langle \mathsf{I}, O'' \rangle \models sk(G_{O''})\})$.

Assumption: Assume that the Lemma holds for $k < \alpha$, we will show that the Lemma holds for $k = \alpha$.

It holds $\mathsf{I}'_\alpha \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}}$, $\mathsf{I}'_\alpha \geq \mathsf{I}'_{\alpha-1}$. Let $O'' \in D^{\mathcal{R}}_{O'}$ and let $r \in [P_{O''}]$. If $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$ then, since $O' \in D^{\mathcal{R}}_{O}$, $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$. Therefore, $\langle \mathsf{I}_\alpha, O'' \rangle \models Concl(r)$. Thus, $\langle \mathsf{I}'_\alpha, O'' \rangle \models Concl(r)$. Thus, $\mathsf{I}'_\alpha \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}'_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O'}$, it holds that: if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\}$. We will show that $\mathsf{I}'_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}'_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O'}$, it holds that: if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\})$.

Assume that $\mathsf{Z} \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}'_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O'}$, it holds that: if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\}$ and $\mathsf{Z} < \mathsf{I}'_{\alpha}$. Then, let $\mathsf{Z}' = \{Z_{O''} \in \mathsf{Z} \mid O'' \in D^{\mathcal{R}}_{O'}\} \cup \{I_{\alpha O''} \in \mathsf{I}_{\alpha} \mid O'' \in D^{\mathcal{R}}_{O} - D^{\mathcal{R}}_{O'}\}$.

It holds $\mathsf{Z}' \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, $\mathsf{Z}' \geq \mathsf{I}_{\alpha-1}$. Let $O'' \in D^{\mathcal{R}}_{O} - D^{\mathcal{R}}_{O'}$ and let $r \in [P_{O''}]$. If $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$ then $\langle \mathsf{I}_{\alpha}, O'' \rangle \models Concl(r)$. Thus, $\langle \mathsf{Z}', O'' \rangle \models Concl(r)$. Let $O'' \in D^{\mathcal{R}}_{O'}$ and let $r \in [P_{O''}]$. If $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$ then $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$. Therefore, $\langle \mathsf{Z}, O'' \rangle \models Concl(r)$. Thus, $\langle \mathsf{Z}', O'' \rangle \models Concl(r)$. Thus, $\mathsf{Z}' \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O}$, it holds that: if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\}$ and $\mathsf{Z}' < \mathsf{I}_{\alpha}$, which is impossible.

Therefore, $\mathsf{I}'_{\alpha} \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}'_{\alpha-1}$ and $\forall O'' \in D^{\mathcal{R}}_{O'}$, it holds that: if $r \in [P_{O''}]$ s.t. $\langle \mathsf{J}, O'' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}'_{\alpha-1}, \mathsf{M}']_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'' \rangle \models Concl(r)\})$.

End of proof of Lemma

Now note that ${\mathsf{I}'}_0 \leq ... \leq {\mathsf{I}'}_k$ and that ${\mathsf{I}'}_{k-1} = {\mathsf{I}'}_k = \mathsf{M}'$. Thus, $\mathsf{M}' \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$. □

Proposition 8 Let $O = \langle G, P \rangle$ be an ERDF ontology and let O' be an ontology such that $G_{O'} = G$, $P_{O'} = P$, and $Int_{O'} = \{\}$. Additionally, let $\mathcal{R} = \{O'\}$ and let F be an ERDF formula over $V_O^{\#n_{O'}}$. It holds that: $O \models^{st\#n_{O'}} F$ iff $O' \models^{\mathtt{st}}_{\mathcal{R}} F$.

Proof:

$\Leftarrow$) Let $O' \models^{\mathtt{st}}_{\mathcal{R}} F$ and let $M \in \mathcal{M}^{st\#n_{O'}}(O)$ (see Definition 20 of Appendix A). We will show that $M \models F$. Since $M \in \mathcal{M}^{st\#n_{O'}}(O)$, there is a chain of $\#n_{O'}$-Herbrand interpretations of O, $I_0 \leq ... \leq I_{k+1}$, such that $I_k = I_{k+1} = M$ and:

1. $I_0 \in minimal(\{I \in \mathcal{I}^{H\#n_{O'}}(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k+1$:
   $I_\alpha \in minimal(\{I \in \mathcal{I}^{H\#n_{O'}}(O) \mid I \geq I_{\alpha-1}$ and it holds that:
   $\forall\, r \in [P]_{V_O^{\#n_{O'}}}$, if $J \models Cond(r)$, $\forall J \in [I_{\alpha-1}, M]_O^{\#n_{O'}}$, then $I \models Concl(r)\}$.

For each I_α, $\alpha \in \{0, ..., k+1\}$, we define a modular Herbrand interpretation I_α of O' w.r.t. $\mathcal{R}$ s.t. $\mathsf{I}_\alpha = \{I_\alpha\}$. It is easy to see that a chain of modular Herbrand interpretations of O' w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_{k+1}$, is formed such that $\mathsf{I}_k = \mathsf{I}_{k+1} = \mathsf{M} = \{M\}$ and

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \langle \mathsf{I}, O' \rangle \models sk(G)\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and it holds that: if $r \in [P]_{V_{O',\mathcal{R}}}$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

Therefore, $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$. Since $O' \models^{\mathtt{st}}_{\mathcal{R}} F$, it follows that $\langle \mathsf{M}, O' \rangle \models F$.

Lemma: Let F be an ERDF formula. It holds that $\langle \mathsf{M}, O' \rangle \models F$ iff $M \models F$.

Proof: We will prove the lemma by induction (see Definition 18 of Appendix A). Without loss of generality, we assume that $\neg$ appears only in front of positive ERDF triples. Otherwise we apply the transformation rules of Definition 18, to get an equivalent formula that satisfies the assumption.

Let $v : Var(F) \rightarrow Res_O^{H\#n_{O'}}$. Note that $Res_O^{H\#n_{O'}} = Res^{\mathtt{H}}_{O',\mathcal{R}}$.

Let $F = p(s,o)$. It holds that: $\langle \mathsf{M}, O', v \rangle \models F$ iff $\langle \mathsf{M}, O', v \rangle \models p(s,o)$ iff $p \in V_M \cap URI$, $s, o \in V_M \cup Var$, $M(p) \in Prop_M$, and $\langle [M+v](s), [M+v](o) \rangle \in PT_M(M(p))$ iff $M, v \models p(s,o)$ iff $M, v \models F$.

Let $F = \neg p(s,o)$. It holds that: $\langle \mathsf{M}, O', v \rangle \models F$ iff $\langle \mathsf{M}, O', v \rangle \models \neg p(s,o)$ iff $p \in V_M \cap URI$, $s, o \in V_M \cup Var$, $M(p) \in Prop_M$, and $\langle [M+v](s), [M+v](o) \rangle \in PF_M(M(p))$ iff $M, v \models \neg p(s,o)$ iff $M, v \models F$.

Assumption: Assume that the lemma holds for the subformulas of F.

We will show that the lemma holds also for F.

Let $F = {\sim}G$. It holds that: $\langle \mathsf{M}, O', v\rangle \models F$ iff $\langle \mathsf{M}, O', v\rangle \models {\sim}G$ iff $\langle \mathsf{M}, O', v\rangle \not\models G$ iff $V_G \subseteq V_M$ and $M, v \not\models G$ iff $M, v \models {\sim}G$ iff $M, v \models F$.

Let $F = F_1 \wedge F_2$. It holds that: $\langle \mathsf{M}, O', v\rangle \models F$ iff $\langle \mathsf{M}, O', v\rangle \models F_1 \wedge F_2$ iff $\langle \mathsf{M}, O', v\rangle \models F_1$ and $\langle \mathsf{M}, O', v\rangle \models F_2$ iff $M, v \models F_1$ and $M, v \models F_2$ iff $M, v \models F_1 \wedge F_2$ iff $M, v \models F$.

Let $F = \exists x G$. It holds: $\langle \mathsf{M}, O', v\rangle \models F$ iff $\langle \mathsf{M}, O', v\rangle \models \exists x G$ iff there exists a mapping $u : \mathit{Var}(G) \rightarrow \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$ s.t. $u(y) = v(y),\ \ \forall y \in \mathit{Var}(G) - \{x\}$ s.t. $\langle \mathsf{M}, O', u\rangle \models G$ iff there exists a mapping $u : \mathit{Var}(G) \rightarrow \mathit{Res}_O^{H_{\#n_{O'}}}$ s.t. $u(y) = v(y),\ \ \forall y \in \mathit{Var}(G) - \{x\}$ s.t. $M, u \models G$ iff $M, v \models \exists x G$ iff $M, v \models F$.

Let $F = F_1 \vee F_2$ or $F = F_1 \supset F_2$ or $F = \forall x G$. We can prove, similarly to the above cases, that $\langle \mathsf{M}, O', v\rangle \models F$ iff $M, v \models F$.
End of Lemma

Since $\langle \mathsf{M}, O'\rangle \models F$ and $\mathsf{M} = \{M\}$, it follows from Lemma that $M \models F$.

$\Rightarrow$) Let $O \models^{st_{\#n_{O'}}} F$ and let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$. We will show that $\langle \mathsf{M}, O'\rangle \models F$. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$, it follows that there is a chain of modular Herbrand interpretations of O' w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_{k+1}$, such that $\mathsf{I}_k = \mathsf{I}_{k+1} = \mathsf{M} = \{M\}$ and

1. $\mathsf{I}_0 \in \mathit{minimal}(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid\ \langle \mathsf{I}, O'\rangle \models sk(G)\})$.
2. For $0 < \alpha \leq k$:
 $\mathsf{I}_\alpha \in \mathit{minimal}(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O',\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and it holds that: if $r \in [P]_{V_{O',\mathcal{R}}}$ s.t. $\langle \mathsf{J}, O'\rangle \models \mathit{Cond}(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O',\mathcal{R}}$, then $\langle \mathsf{I}, O'\rangle \models \mathit{Concl}(r)\})$.

For each I_α, $\alpha \in \{0, ..., k+1\}$, we define an $\#n_{O'}$-Herbrand interpretation I_α of O s.t. $\mathsf{I}_\alpha = \{I_\alpha\}$. It is easy to see that a chain of $\#n_{O'}$-Herbrand interpretations of O, $I_0 \leq ... \leq I_{k+1}$, is formed such that $I_k = I_{k+1} = M$ and:

1. $I_0 \in \mathit{minimal}(\{I \in \mathcal{I}^{H_{\#n_{O'}}}(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k+1$:
 $I_\alpha \in \mathit{minimal}(\{I \in \mathcal{I}^{H_{\#n_{O'}}}(O) \mid I \geq I_{\alpha-1}$ and it holds that:
 $\forall\, r \in [P]_{V_O^{\#n_{O'}}}$, if $J \models \mathit{Cond}(r),\ \forall J \in [I_{\alpha-1}, M]_O^{\#n_{O'}}$, then $I \models \mathit{Concl}(r)\}$.

Thus, $M \in \mathcal{M}^{st_{\#n_{O'}}}(O)$. As $O \models^{st_{\#n_{O'}}} F$, it follows that $M \models F$. Since $\mathsf{M} = \{M\}$ and F is an ERDF formula, it follows from Lemma that $\langle \mathsf{M}, O'\rangle \models F$. $\Box$

Proposition 9 Let G, G' be RDF graphs such that $V_G \cap \mathcal{V}_{ERDF} = \emptyset$, $V_{G'} \cap \mathcal{V}_{ERDF} = \emptyset$, and $V_{G'} \cap sk_G(\mathit{Var}(G)) = \emptyset$. Let O be an ontology with $G_O = G$, $P_O = \{\}$, and $\mathit{Int}_O = \{\}$. If $\mathit{max}(\{i \in I\!N \mid \mathit{rdf}\!:\!\_i \in V_{G'}\}) \leq n_O$ then: $G \models^{RDFS} G'$ iff $O \models^{\mathtt{st}}_{\mathcal{R}} G'$, where $\mathcal{R} = \{O\}$.

Proof: First, we will prove the following Lemma:
Lemma: Let $O = \langle G, P\rangle$ be an ERDF ontology and let $F = t_1 \wedge ... \wedge t_n$, where t_i is an ERDF triple. Additionally, let O' be an ontology such that $G_{O'} = G$, $P_{O'} = P$, and $\mathit{Int}_{O'} = \{\}$. It holds that: $O \models^{st_{\#n_{O'}}} F$ iff $O' \models^{\mathtt{st}}_{\mathcal{R}} F$, where $\mathcal{R} = \{O'\}$.
Proof:
$\Leftarrow$) Let $O' \models^{\mathtt{st}}_{\mathcal{R}} F$, let $M \in \mathcal{M}^{st_{\#n_{O'}}}(O)$ (see Definition 20 of Appendix A), and let $\mathsf{M} = \{M\}$. It is proved exactly as in the proof of Proposition 8 that $\langle \mathsf{M}, O'\rangle \models F$.

Sublemma: It holds that: $\langle \mathsf{M}, O'\rangle \models F$ iff $M \models F$.
Proof: Let $v : \mathit{Var}(F) \rightarrow \mathit{Res}_O^{H_{\#n_{O'}}}$. Note that $\mathit{Res}_O^{H_{\#n_{O'}}} = \mathit{Res}^{\mathtt{H}}_{O',\mathcal{R}}$.

Let $F = p(s, o)$. It holds that: $\langle \mathsf{M}, O', v\rangle \models F$ iff $\langle \mathsf{M}, O', v\rangle \models p(s, o)$ iff $p \in V_M \cap \mathit{URI}$, $s, o \in V_M \cup \mathit{Var}$, $M(p) \in \mathit{Prop}_M$, and $\langle [M+v](s), [M+v](o)\rangle \in PT_M(M(p))$ iff $M, v \models p(s, o)$ iff $M, v \models F$.

Let $F = \neg p(s, o)$. It holds that: $\langle \mathsf{M}, O', v\rangle \models F$ iff $\langle \mathsf{M}, O', v\rangle \models \neg p(s, o)$ iff $p \in V_M \cap \mathit{URI}$, $s, o \in V_M \cup \mathit{Var}$, $M(p) \in \mathit{Prop}_M$, and $\langle [M+v](s), [M+v](o)\rangle \in PF_M(M(p))$ iff $M, v \models \neg p(s, o)$ iff $M, v \models F$.

Assumption: Assume that the sublemma holds for the subformulas of F.

We will show that the sublemma holds also for F.

Let $F = F_1 \wedge F_2$. It holds that: $\langle \mathsf{M}, O', v \rangle \models F$ iff $\langle \mathsf{M}, O', v \rangle \models F_1 \wedge F_2$ iff $\langle \mathsf{M}, O', v \rangle \models F_1$ and $\langle \mathsf{M}, O', v \rangle \models F_2$ iff $M, v \models F_1$ and $M, v \models F_2$ iff $M, v \models F_1 \wedge F_2$ iff $M, v \models F$.

End of Sublemma

Since $\langle \mathsf{M}, O' \rangle \models F$ and $\mathsf{M} = \{M\}$, it follows from sublemma that $M \models F$.

$\Rightarrow$) Let $O \models^{st_{\#n_{O'}}} F$ and let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O',\mathcal{R}}$. It is proved exactly as in the proof of Proposition 8 that $M \in \mathcal{M}^{st_{\#n_{O'}}}(O)$. As $O \models^{st_{\#n_{O'}}} F$, it follows that $M \models F$. Since $\mathsf{M} = \{M\}$ and $F = t_1 \wedge ... \wedge t_n$, where t_i is an ERDF triple, it follows from sublemma that $\langle \mathsf{M}, O' \rangle \models F$.

End of Lemma

The Proposition follows directly from the above Lemma and Proposition 3 in [7] (or equivalently Proposition 5 in [5]). □

Corollary 1 Let $O = \langle G, P \rangle$ be an ERDF ontology and let O' be an ontology such that $G_{O'} = G$, $P_{O'} = P$, and $Int_{O'} = \{\}$. It holds that O has an $\#n_{O'}$-stable model iff O' has a modular stable model w.r.t. $\mathcal{R}$, where $\mathcal{R} = \{O'\}$.

Proof: Let $F = p(x, y) \wedge \neg p(x, y)$, for $p, x, y \in V_O^{\#n_{O'}}$. It follows from Proposition 8 that $O \models^{st_{\#n_{O'}}} F$ iff $O' \models^{\mathtt{st}}_{\mathcal{R}} F$. Thus, O does not have an $\#n_{O'}$-stable model iff O' does not have a modular stable model w.r.t. $\mathcal{R}$. Thus, O has an $\#n_{O'}$-stable model iff O' has a modular stable model w.r.t. $\mathcal{R}$. □

Proposition 10 Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. Let M be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$. It holds that: $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ iff $ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$.

Proof:

$\Rightarrow$) Let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. We will show that $N = ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$, it follows that there is a chain of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_k$ such that $\mathsf{I}_{k-1} = \mathsf{I}_k = \mathsf{M}$ and:

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \ \forall O' \in D^{\mathcal{R}}_O, \ \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that:
   if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

Now, we define a sequence $N_0 \subseteq .... \subseteq N_{k+1} \subseteq EHB(\Pi_{O,\mathcal{R}})$, as follows:

$N_0 = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset))$.

$N_\alpha = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{(\mathtt{P}^{\mathcal{R}}_O)^N}(N_{\alpha-1}))$, where $1 \leq \alpha \leq k+1$.

Lemma: It holds $N_\alpha = ELP(\mathsf{I}_\alpha)$, for $\alpha = 0, ..., k+1$.

Proof: We will prove the Lemma, by induction.

First, we will show that $N_0 \subseteq ELP(\mathsf{I}_0)$. Since $\langle \mathsf{I}_0, O' \rangle \models sk(G_{O'})$, for all $O' \in D^{\mathcal{R}}_O$, it follows that $T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset) \subseteq ELP(\mathsf{I}_0)$. Since $\mathsf{I}_0 \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $ELP(\mathsf{I}_0)$ satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. As $ELP(\mathsf{I}_0) \subseteq N$, it follows that $ELP(\mathsf{I}_0)$ satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$. As $T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset) \subseteq ELP(\mathsf{I}_0)$, it follows that $T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset)) \subseteq ELP(\mathsf{I}_0)$. Therefore, $N_0 \subseteq ELP(\mathsf{I}_0)$.

Let $O' \in D^{\mathcal{R}}_O$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_0$, for $p \in V_{O',\mathcal{R}}$. As $N_0 \subseteq ELP(\mathsf{I}_0) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_0$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_0$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_0$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above and as N_0 satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$, it follows that N_0 satisfies all rules in $\mathtt{H}_O^{\mathcal{R}}$. Therefore, $ELP^{-1}(N_0)$ satisfies all semantic conditions of a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$. Thus, $ELP^{-1}(N_0) \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. Moreover, $\langle ELP^{-1}(N_0), O' \rangle \models sk(G_{O'})$, for all $O' \in D_O^{\mathcal{R}}$. Therefore, $ELP^{-1}(N_0) \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D_O^{\mathcal{R}}, \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\}$. Now as $I_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D_O^{\mathcal{R}}, \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$ and $N_0 \subseteq ELP(\mathsf{I}_0)$, it follows that $N_0 = ELP(\mathsf{I}_0)$.

Assumption: We assume that $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$, for an $\alpha \leq k$.

We will show that $N_\alpha = ELP(\mathsf{I}_\alpha)$. First, we will show that $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$. Due to assumption $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$ and the fact $\mathsf{I}_{\alpha-1} \leq \mathsf{I}_\alpha$, it follows that $N_{\alpha-1} \subseteq ELP(\mathsf{I}_\alpha)$. Based on (i) the assumption $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$ and (ii) the fact $\langle \mathsf{I}_\alpha, O' \rangle \models Concl(r)$, for all $O' \in D_O^{\mathcal{R}}$ and for all $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, it follows that $T_{(\mathtt{P}_O^{\mathcal{R}})^N}(N_{\alpha-1}) \subseteq ELP(I_\alpha)$.

Since $\mathsf{I}_\alpha \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $ELP(\mathsf{I}_\alpha)$ satisfies all rules in $\mathtt{H}_O^{\mathcal{R}}$. As $ELP(\mathsf{I}_\alpha) \subseteq N$, it follows that $ELP(\mathsf{I}_\alpha)$ satisfies all rules in $(\mathtt{H}_O^{\mathcal{R}})^N$. As $T_{(\mathtt{P}_O^{\mathcal{R}})^N}(N_{\alpha-1}) \subseteq ELP(I_\alpha)$, it follows that $T^{\uparrow\omega}_{(\mathtt{H}_O^{\mathcal{R}})^N}(T_{(\mathtt{P}_O^{\mathcal{R}})^N}(N_{\alpha-1})) \subseteq ELP(I_\alpha)$. Therefore, $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$.

Let $O' \in D_O^{\mathcal{R}}$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_\alpha$, for $p \in V_{O',\mathcal{R}}$. As $N_\alpha \subseteq ELP(\mathsf{I}_\alpha) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_\alpha$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above and as N_α satisfies all rules in $(\mathtt{H}_O^{\mathcal{R}})^N$, it follows that N_α satisfies all rules in $\mathtt{H}_O^{\mathcal{R}}$. Therefore, $ELP^{-1}(N_\alpha)$ satisfies all semantic conditions of a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$. Thus, $ELP^{-1}(N_\alpha) \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. Moreover, $ELP^{-1}(N_\alpha) \geq ELP^{-1}(N_{\alpha-1}) = \mathsf{I}_{\alpha-1}$. Now, based on the assumption that $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1}) \subseteq N$ and the fact that $T_{(\mathtt{P}_O^{\mathcal{R}})^N}(N_{\alpha-1}) \subseteq N_\alpha$, it follows that $\langle ELP^{-1}(N_\alpha), O' \rangle \models Concl(r)$, for all $O' \in D_O^{\mathcal{R}}$, for all $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$. Therefore, $ELP^{-1}(N_\alpha) \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D_O^{\mathcal{R}}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\}$. Now as $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D_O^{\mathcal{R}}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r)$, $\forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$ and $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$, it follows that $N_\alpha = ELP(\mathsf{I}_\alpha)$.
End of Lemma

Therefore, $N_k = N_{k+1} = ELP(\mathsf{M})$. Since M is a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$, it follows that $N = ELP(\mathsf{M})$ is consistent. Moreover, since $[\Pi_{O,\mathcal{R}}]^N = \bigcup\{\Pi^{O'}_{G_{O'}} \cup [\Pi^{O'}_{P_{O'}}]^N \cup [\Pi^{H_\mathcal{R}}_{O'}]^N \mid O' \in D_O^{\mathcal{R}}\}$, it follows that $T^{\uparrow\omega}_{[\Pi_{O,\mathcal{R}}]^N}(\emptyset) = N$. Therefore, $N = ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$.

$\Leftarrow$) Let $N = ELP(\mathsf{M})$ be a non-contradictory answer set of $\Pi_{O,\mathcal{R}}$. We will show that $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. Since N is a non-contradictory answer set of $\Pi_{O,\mathcal{R}}$, it follows that $T^{\uparrow\omega}_{[\Pi_{O,\mathcal{R}}]^N}(\emptyset) = N$. We define a sequence $N_\alpha \subseteq EHB(\Pi_{O,\mathcal{R}})$, $\alpha \in \{0, 1, ...\}$, as follows:

$N_0 = T^{\uparrow\omega}_{(\mathtt{H}_O^{\mathcal{R}})^N}(T_{\mathtt{G}_O^{\mathcal{R}}}(\emptyset))$.

$N_\alpha = T^{\uparrow\omega}_{(\mathtt{H}_O^{\mathcal{R}})^N}(T_{(\mathtt{P}_O^{\mathcal{R}})^N}(N_{\alpha-1}))$, where $1 \leq \alpha \leq k+1$.

Since $EHB(\Pi_{O,\mathcal{R}})$ is a finite set and $[\Pi_{O,\mathcal{R}}]^N = \bigcup\{\Pi^{O'}_{G_{O'}} \cup [\Pi^{O'}_{P_{O'}}]^N \cup [\Pi^{H_\mathcal{R}}_{O'}]^N \mid O' \in D_O^{\mathcal{R}}\}$, it follows that there is $k \in \{0, 1, ...\}$ such that $N_k = N_{k+1} = N$.

Let $O' \in D_O^{\mathcal{R}}$. Let $H(Nam_{O'}, p, \texttt{type}, \texttt{TotalProperty}) \in N_0$, for $p \in V_{O',\mathcal{R}}$. As $N_0 \subseteq ELP(\mathsf{I}_0) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \texttt{type}, \texttt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_0$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \texttt{type}, \texttt{TotalClass}) \in N_0$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \texttt{type}, \texttt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \texttt{type}, c) \in N_0$ iff $[\neg]H(Nam_{O'}, x, \texttt{type}, c) \in N$. Now, from the above and as N_0 is the smallest subset of $EHB(\Pi_{O,\mathcal{R}})$ that satisfies all rules in $\mathtt{G}_O^{\mathcal{R}} \bigcup (\mathtt{H}_O^{\mathcal{R}})^N$, it follows that N_0 satisfies all rules in $\mathtt{H}_O^{\mathcal{R}}$. Therefore, N_0 is a minimal subset of $EHB(\Pi_{O,\mathcal{R}})$ that satisfies all rules $\mathtt{G}_O^{\mathcal{R}} \bigcup \mathtt{H}_O^{\mathcal{R}}$. Thus, $ELP^{-1}(N_0) \in minimal(\{\mathsf{I} \in \mathcal{I}_{O,\mathcal{R}}^{\mathtt{H}} \mid \forall O' \in D_O^{\mathcal{R}},\ \langle \mathsf{I}, O'\rangle \models sk(G_{O'})\})$.

Let α such that $1 \leq \alpha \leq k+1$. Note that N_α is the smallest subset of $EHB(\Pi_{O,\mathcal{R}})$ such that (i) $N_\alpha \supseteq N_{\alpha-1}$, (ii) satisfies all rules in $(\mathtt{H}_O^{\mathcal{R}})^N$, and (iii) $Head(r) \in N_\alpha$, for all $r \in (\mathtt{P}_O^{\mathcal{R}})^N$ such that $Body(r) \subseteq N_{\alpha-1}$. Let $O' \in D_O^{\mathcal{R}}$. Let $H(Nam_{O'}, p, \texttt{type}, \mathit{TotalProperty}) \in N_\alpha$, for $p \in V_{O',\mathcal{R}}$. As $N_\alpha \subseteq ELP(\mathsf{I}_\alpha) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \texttt{type}, \texttt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \texttt{type}, \texttt{TotalClass}) \in N_\alpha$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \texttt{type}, \texttt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \texttt{type}, c) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, \texttt{type}, c) \in N$. Now, from the above, it follows that N_α is a minimal subset of $EHB(\Pi_{O,\mathcal{R}})$ such that (i) $N_\alpha \supseteq N_{\alpha-1}$, (ii) satisfies all rules in $\mathtt{H}_O^{\mathcal{R}}$, and (iii) $Head(r) \in N_\alpha$, for all $r \in (\mathtt{P}_O^{\mathcal{R}})^N$ such that $Body(r) \subseteq N_{\alpha-1}$. Therefore, as $N_{\alpha-1} \subseteq N$, it follows that $ELP^{-1}(N_\alpha) \in minimal(\{\mathsf{I} \in \mathcal{I}_{O,\mathcal{R}}^{\mathtt{H}} \mid \mathsf{I} \geq ELP^{-1}(N_{\alpha-1})$ and $\forall O' \in D_O^{\mathcal{R}}$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O'\rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), ELP^{-1}(N)]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O'\rangle \models Concl(r)\})$.

Now as $ELP^{-1}(N_0) \leq ... \leq ELP^{-1}(N_{k+1})$ and $ELP^{-1}(N_k) = ELP^{-1}(N_{k+1}) = ELP^{-1}(N) = \mathsf{M}$, it follows that $\mathsf{M} \in \mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}}$. $\Box$

Proposition 11 Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. Additionally, let F be a simple $\mathtt{r}$-ERDF formula over $\{Nam_{O'} \mid O' \in D_O^{\mathcal{R}}\}$. Then,
(i) if $\Pi_{O,\mathcal{R}}$ is a non-contradictory ELP [31] then $Ans_{O,\mathcal{R}}^{\mathtt{st}}(F) = Ans_{\Pi_{O,\mathcal{R}}}^{AS}(L_F^O)$, and
(ii) $\mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}} = \emptyset$, otherwise.

Proof:

(i)
$\Rightarrow$) Let $v \in Ans_{O,\mathcal{R}}^{\mathtt{st}}(F)$ and let $v(F) = t_1 \wedge ... \wedge t_k \wedge {\sim} t_{k+1} \wedge ... \wedge {\sim} t_n$, where each t_i is a normal or qualified ERDF triple (positive or negative). We will show that $v \in Ans_{\Pi_{O,\mathcal{R}}}^{AS}(L_F^O)$. Let N be a consistent answer set of $\Pi_{O,\mathcal{R}}$. Then, based on Proposition 10, $ELP^{-1}(N) \in \mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}}$. Thus, $\langle ELP^{-1}(N), O\rangle \models v(F)$. Then, (i) $\langle ELP^{-1}(N), O\rangle \models t_i$, for $i = 1, ..., k$ and (ii) $\langle ELP^{-1}(N), O\rangle \models {\sim} t_i$, , for $i = k+1, ..., n$. It follows from this that (i) $N \models L_{t_i}^O$, for $i = 1, ..., k$ and (ii) $N \models {\sim} L_{t_i}^O$, for $i = k+1, ..., n$. Therefore, $N \models v(L_F^O)$. Therefore, $v \in Ans_{\Pi_{O,\mathcal{R}}}^{AS}(L_F^O)$.

$\Leftarrow$) Let $v \in Ans_{\Pi_{O,\mathcal{R}}}^{AS}(L_F^O)$ and let $v(F) = t_1 \wedge ... t_k \wedge {\sim} t_{k+1} \wedge ... \wedge {\sim} t_n$, where each t_i is a normal or qualified ERDF triple (positive or negative). We will show that $v \in Ans_{O,\mathcal{R}}^{\mathtt{st}}(F)$. Let $\mathsf{M} \in \mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}}$. Then, based on Proposition 10, $ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$. Then, $ELP(\mathsf{M}) \models v(L_F^O)$. Thus, (i) $ELP(\mathsf{M}) \models L_{t_i}^O$, for $i = 1, ..., k$ and (ii) $ELP(\mathsf{M}) \models {\sim} L_{t_i}^O$, for $i = k+1, ..., n$. Therefore, (i) $\langle \mathsf{M}, O\rangle \models t_i$, for $i = 1, ..., k$ and (ii) $\langle \mathsf{M}, O\rangle \models {\sim} t_i$, , for $i = k+1, ..., n$. Therefore, $\langle \mathsf{M}, O\rangle \models v(F)$. Thus, $v \in Ans_{O,\mathcal{R}}^{\mathtt{st}}(F)$.

(ii) If $\Pi_{O,\mathcal{R}}$ is a contradictory ELP then there is no consistent answer set of $\Pi_{O,\mathcal{R}}$. Assume now that there is an $\mathsf{M} \in \mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}}$. Then, based on Proposition 10, $ELP(\mathsf{M})$ is a consistent answer set of $\Pi_{O,\mathcal{R}}$, which is impossible. Therefore, $\mathcal{M}_{O,\mathcal{R}}^{\mathtt{st}} = \emptyset$. $\Box$

Proposition 12 Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$. The problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$.

Proof:

Hardness) Let $O' = \langle G', P' \rangle$ be a simple ERDF ontology and let O'' be an ontology such that $G_{O''} = G'$, $P_{O''} = P'$, and $Int_{O''} = \{\}$. Let $\mathcal{R}'' = \{O''\}$. In Proposition 13 of [5], we show that the problem of establishing whether O' has an $\#n_{O''}$-stable model is $\mathrm{NP}^{\mathrm{NP}}$-hard w.r.t. the size of O'. In fact, this hardness result is based on a reduction from the *2-quantified boolean formula* (*2-QBF*) problem, which is a $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$-complete problem [69, 57]. It follows from Corollary 1 that O' has an $\#n_{O''}$-stable model iff O'' has a modular stable model w.r.t. $\mathcal{R}''$. Note that to generate O'' and R'' from O', it takes polynomial time and that $size(O'', \mathcal{R}'')$ is polynomial w.r.t. the size of O'. Therefore, the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\mathrm{NP}^{\mathrm{NP}}$-hard w.r.t. $size(O, \mathcal{R})$.

Membership) It follows directly from the membership part of Proposition 15(i) that the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is in $\mathrm{NP}^{\mathrm{NP}}$ w.r.t. $size(O, \mathcal{R})$.

It follows from the hardness and membership steps, above, that the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$. $\square$

Proposition 13 Let $O \in \mathcal{R}$ s.t. O is a simple ontology w.r.t. $\mathcal{R}$ and let F be an $\mathtt{r}$-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \to V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. The problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is co-$\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$.

Proof:

Hardness) Let $F = p(s, o) \wedge \neg p(s, o)$. Then, $O \models^{\mathtt{st}}_{\mathcal{R}} F$ iff O has no modular stable model w.r.t. $\mathcal{R}$. It follows from Proposition 12 that the complexity of deciding whether O has a modular stable model w.r.t. $\mathcal{R}$ is $\mathrm{NP}^{\mathrm{NP}}$-hard w.r.t. $size(O, \mathcal{R})$. Therefore, the complexity of deciding whether $O \models^{\mathtt{st}}_{\mathcal{R}} F$ is co-$\mathrm{NP}^{\mathrm{NP}}$-hard w.r.t. $size(O, \mathcal{R})$.

Membership) It follows directly from the membership part of Proposition 15(ii).

From the hardness and membership steps, above, it follows that the problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is co-$\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. $size(O, \mathcal{R})$. $\square$

Proposition 14 Let $O \in \mathcal{R}$. Let M be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$. It holds that: $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ iff *Is-ModularStableModelGeneral*($\mathcal{R}, O, \mathsf{M}$)=TRUE.

Proof:

$\Rightarrow$) Let $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. We will show that *Is-ModularStableModelGeneral*($\mathcal{R}, O, \mathsf{M}$)=TRUE. Since $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$, it follows that there is a chain of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$, $\mathsf{I}_0 \leq ... \leq \mathsf{I}_k$ such that $\mathsf{I}_{k-1} = \mathsf{I}_k = \mathsf{M}$ and:

1. $\mathsf{I}_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D^{\mathcal{R}}_O,\ \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$.
2. For $0 < \alpha \leq k$:
   $\mathsf{I}_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that:
   if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$.

Now, we define a sequence $N_0 \subseteq .... \subseteq N_{k+1} \subseteq EHB(\Pi_{O,\mathcal{R}})$, as follows:

$N_0 = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset))$.

$N_\alpha = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}\})$, where $1 \leq \alpha \leq k+1$.

Lemma: It holds that $N_\alpha = ELP(\mathsf{I}_\alpha)$, for $\alpha = 0, ..., k+1$.
Proof: We will prove the Lemma, by induction.

First, we will show that $N_0 \subseteq ELP(\mathsf{I}_0)$. Since $\langle \mathsf{I}_0, O' \rangle \models sk(G_{O'})$, for all $O' \in D^{\mathcal{R}}_O$, it follows that $T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset) \subseteq ELP(\mathsf{I}_0)$. Since $\mathsf{I}_0 \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $ELP(\mathsf{I}_0)$ satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. As $ELP(\mathsf{I}_0) \subseteq N$, it follows that $ELP(\mathsf{I}_0)$ satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$. As $T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset) \subseteq ELP(\mathsf{I}_0)$, it follows that $T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset)) \subseteq ELP(\mathsf{I}_0)$. Therefore, $N_0 \subseteq ELP(\mathsf{I}_0)$.

Let $O' \in D^{\mathcal{R}}_O$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_0$, for $p \in V_{O',\mathcal{R}}$. As $N_0 \subseteq ELP(\mathsf{I}_0) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_0$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_0$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_0$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above and as N_0 satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$, it follows that N_0 satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. Therefore, $ELP^{-1}(N_0)$ satisfies all semantic conditions of a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$. Thus, $ELP^{-1}(N_0) \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. Moreover, $\langle ELP^{-1}(N_0), O' \rangle \models sk(G_{O'})$ for all $O' \in D^{\mathcal{R}}_O$. Therefore, $ELP^{-1}(N_0) \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D^{\mathcal{R}}_O, \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\}$. Now as $I_0 \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \forall O' \in D^{\mathcal{R}}_O, \langle \mathsf{I}, O' \rangle \models sk(G_{O'})\})$ and $N_0 \subseteq ELP(\mathsf{I}_0)$, it follows that $N_0 = ELP(\mathsf{I}_0)$.

Assumption: We assume that $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$, for an $\alpha \leq k$.

We will show that $N_\alpha = ELP(\mathsf{I}_\alpha)$. First, we will show that $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$. Due to assumption $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$ and the fact $\mathsf{I}_{\alpha-1} \leq \mathsf{I}_\alpha$, it follows that $N_{\alpha-1} \subseteq ELP(\mathsf{I}_\alpha)$. Based on (i) the assumption $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1})$ and (ii) the fact $\langle \mathsf{I}_\alpha, O' \rangle \models Concl(r)$, for all $O' \in D^{\mathcal{R}}_O$, for all $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, it follows that $N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}\} \subseteq ELP(I_\alpha)$.

Since $\mathsf{I}_\alpha \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$, it follows that $ELP(\mathsf{I}_\alpha)$ satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. As $ELP(\mathsf{I}_\alpha) \subseteq N$, it follows that $ELP(\mathsf{I}_\alpha)$ satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$. As $N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}\} \subseteq ELP(I_\alpha)$, it follows that $T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}\}) \subseteq ELP(I_\alpha)$. Therefore, $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$.

Let $O' \in D^{\mathcal{R}}_O$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_\alpha$, for $p \in V_{O',\mathcal{R}}$. As $N_\alpha \subseteq ELP(\mathsf{I}_\alpha) \subseteq ELP(\mathsf{M}) = N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_\alpha$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above and as N_α satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$, it follows that N_α satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. Therefore, $ELP^{-1}(N_\alpha)$ satisfies all semantic conditions of a modular Herbrand interpretation of O w.r.t. $\mathcal{R}$. Thus, $ELP^{-1}(N_\alpha) \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$. Moreover, $ELP^{-1}(N_\alpha) \geq ELP^{-1}(N_{\alpha-1}) = \mathsf{I}_{\alpha-1}$. Now, based on the assumption that $N_{\alpha-1} = ELP(\mathsf{I}_{\alpha-1}) \subseteq N$ and the fact that $N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}\} \subseteq N_\alpha$, it follows that $\langle ELP^{-1}(N_\alpha), O' \rangle \models Concl(r)$, for all $O' \in D^{\mathcal{R}}_O$, for all $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$. Therefore, $ELP^{-1}(N_\alpha) \in \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\}$. Now as $I_\alpha \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq \mathsf{I}_{\alpha-1}$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O' \rangle \models Cond(r),\ \forall \mathsf{J} \in [\mathsf{I}_{\alpha-1}, \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O' \rangle \models Concl(r)\})$ and $N_\alpha \subseteq ELP(\mathsf{I}_\alpha)$, it

follows that $N_\alpha = ELP(\mathsf{I}_\alpha)$.
End of Lemma

Therefore, $N_k = N_{k+1} = ELP(\mathsf{M})$. Thus, *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$ =TRUE.

$\Leftarrow$) Let M be a modular semi-Herbrand interpretation of O w.r.t. $\mathcal{R}$ s.t. *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$=TRUE. We will show that $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$.

Now, we define a sequence $N_\alpha \subseteq EHB(\Pi_{O,\mathcal{R}})$, $\alpha \in \{0, 1, ...\}$, as follows:

$N_0 = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(T_{\mathtt{G}^{\mathcal{R}}_O}(\emptyset))$.

$N_\alpha = T^{\uparrow\omega}_{(\mathtt{H}^{\mathcal{R}}_O)^N}(N_{\alpha-1} \cup \{L^{O'}_{Concl(r)} \mid O' \in D^{\mathcal{R}}_O,\ r \in [P_{O'}]$, and $\langle \mathsf{J}, O'\rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}\})$, where $1 \leq \alpha$.

Since *Is-ModularStableModelGeneral*$(\mathcal{R}, O, \mathsf{M})$=TRUE, it follows that there is $k \in \{0, 1, ...\}$ such that $N_k = N_{k+1} = ELP(\mathsf{M})$. Let $N = ELP(\mathsf{M})$.

Let $O' \in D^{\mathcal{R}}_O$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_0$, for $p \in V_{O',\mathcal{R}}$. As $N_0 \subseteq N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_0$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_0$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_0$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above and as N_0 is the smallest subset of $EHB(\Pi_{O,\mathcal{R}})$ that satisfies all rules in $\mathtt{G}^{\mathcal{R}}_O \bigcup (\mathtt{H}^{\mathcal{R}}_O)^N$, it follows that N_0 satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$. Therefore, N_0 is a minimal subset of $EHB(\Pi_{O,\mathcal{R}})$ that satisfies all rules in $\mathtt{G}^{\mathcal{R}}_O \bigcup \mathtt{H}^{\mathcal{R}}_O$. Thus, $ELP^{-1}(N_0) \in minimal(\{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid\ \forall O' \in D^{\mathcal{R}}_O,\ \langle \mathsf{I}, O'\rangle \models sk(G_{O'})\})$.

Let α such that $1 \leq \alpha \leq k+1$. Note that N_α is the smallest subset of $EHB(\Pi_{O,\mathcal{R}})$ such that (i) $N_\alpha \supseteq N_{\alpha-1}$, (ii) satisfies all rules in $(\mathtt{H}^{\mathcal{R}}_O)^N$, and (iii) $L^{O'}_{Concl(r)} \in N_\alpha$, for all $O' \in D^{\mathcal{R}}_O$, for all $r \in [P_{O'}]$, and $\langle \mathsf{J}, O'\rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}$. Let $O' \in D^{\mathcal{R}}_O$. Let $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N_\alpha$, for $p \in V_{O',\mathcal{R}}$. As $N_\alpha \subseteq N$, it follows that $H(Nam_{O'}, p, \mathtt{type}, \mathtt{TotalProperty}) \in N$. Therefore, for all $x, y \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, p, y) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, p, y) \in N$. Similarly, let $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N_\alpha$, for $c \in V_{O',\mathcal{R}}$. Then, $H(Nam_{O'}, c, \mathtt{type}, \mathtt{TotalClass}) \in N$. Therefore, for all $x \in V_{O',\mathcal{R}}$, $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N_\alpha$ iff $[\neg]H(Nam_{O'}, x, \mathtt{type}, c) \in N$. Now, from the above, it follows that N_α is a minimal subset of $EHB(\Pi_{O,\mathcal{R}})$ such that (i) $N_\alpha \supseteq N_{\alpha-1}$, (ii) satisfies all rules in $\mathtt{H}^{\mathcal{R}}_O$, and (iii) $L^{O'}_{Concl(r)} \in N_\alpha$, for all $O' \in D^{\mathcal{R}}_O$, for all $r \in [P_{O'}]$, and $\langle \mathsf{J}, O'\rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}$. Therefore, it follows that $ELP^{-1}(N_\alpha) \in minimal(\ \{\mathsf{I} \in \mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}} \mid \mathsf{I} \geq ELP^{-1}(N_{\alpha-1})$ and $\forall O' \in D^{\mathcal{R}}_O$, it holds that: if $r \in [P_{O'}]$ s.t. $\langle \mathsf{J}, O'\rangle \models Cond(r),\ \forall \mathsf{J} \in [ELP^{-1}(N_{\alpha-1}), \mathsf{M}]_{O,\mathcal{R}}$, then $\langle \mathsf{I}, O'\rangle \models Concl(r)\})$.

Now as $ELP^{-1}(N_0) \leq ... \leq ELP^{-1}(N_{k+1})$ and $ELP^{-1}(N_k) = ELP^{-1}(N_{k+1}) = \mathsf{M}$, it follows that $\mathsf{M} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$. $\square$

Proposition 15 Let $O \in \mathcal{R}$ s.t. for all $O' \in D^{\mathcal{R}}_O$, no quantifiers $\forall, \exists$ appear in $P_{O'}$ and let F be an $\mathtt{r}$-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \to V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. Then: (i) the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is Σ^P_2 =NP$^{\text{NP}}$-complete w.r.t. *size*$(O, \mathcal{R})$, and (ii) the problem of establishing whether $v \in Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$ is Π^P_2 = co-NP$^{\text{NP}}$-complete w.r.t. *size*$(O, \mathcal{R})$.

Proof:
(i)
Hardness) It follows directly from the hardness part of Proposition 12.
Membership)
Guess now a modular semi-Herbrand interpretation $\mathcal{M}$ of O w.r.t. $\mathcal{R}$. It is the case that $\mathsf{M} \in \mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}}$ iff the following Condition 1 and Condition 2 hold.
Condition 1) The following should hold:

1. For all $t \in sk(G_{O'})$, for $O' \in D^{\mathcal{R}}_O$, it is the case that $\mathit{Satisfies}(\mathcal{R}, O, \mathcal{M}, O', t)$=TRUE.
2. For all $r \in [P_{O'}]$, for $O' \in D^{\mathcal{R}}_O$, it is the case that if $\mathit{Satisfies}(\mathcal{R}, O, \mathcal{M}, O', \mathit{Cond}(r))$=TRUE then $\mathit{Satisfies}(\mathcal{R}, O, \mathcal{M}, O', \mathit{Concl}(r))$=TRUE.
3. For all $r \in [\Pi^{H_\mathcal{R}}_{O'}]$, for $O' \in D^{\mathcal{R}}_O$, it is the case that $\mathit{ELP}(\mathcal{M})$ satisfies r.

Condition 2) For each $L \in \mathit{ELP}(\mathcal{M})$, there is a founded proof $r_1\theta_1$, ...,$r_k\theta_k$ of L, where $r_i \in \Pi^{O'}_{G_{O'}} \cup P_{O'} \cup \Pi^{H_\mathcal{R}}_{O'}$, for $O' \in D^{\mathcal{R}}_O$, and θ_i is a substitution of the free variables of r_i over $V_{O',\mathcal{R}}$. We define:

$$L_i = \begin{cases} \mathit{Head}(r_i) & \text{if } r_i \in \Pi^{O'}_{G_{O'}}, \text{ for } O' \in D^{\mathcal{R}}_O \\ L_{\mathit{Concl}(r_i\theta_i)} & \text{if } r_i \in P_{O'}, \text{ for } O' \in D^{\mathcal{R}}_O \\ \mathit{Head}(r_i\theta_i) & \text{if } r_i \in \Pi^{H_\mathcal{R}}_{O'}, \text{ for } O' \in D^{\mathcal{R}}_O \end{cases}$$

Additionally, we define:

$$\mathcal{I}_i = \begin{cases} \mathit{ELP}^{-1}(\{L_1, ..., L_{i-1}\}) & \text{if } r_i \in P_{O'}, \text{ for } O' \in D^{\mathcal{R}}_O \\ \{L_1, ..., L_{i-1}\} & \text{if } r_i \in \Pi^{H_\mathcal{R}}_{O'}, \text{ for } O' \in D^{\mathcal{R}}_O \end{cases}$$

It should hold:

1. $L_k = L$.
2. If $r_i \in P_{O'}$, for $O' \in D^{\mathcal{R}}_O$, then $\mathit{MiddleNotSatisfies}(\mathcal{R}, O, O', \mathcal{I}_i, \mathcal{M}, \mathit{Cond}(r_i\theta_i))$=FALSE.
3. If $r_i \in \Pi^{H_\mathcal{R}}_{O'}$, for $O' \in D^{\mathcal{R}}_O$, then $\mathit{Body}^+(r_i\theta_i) \subseteq \mathcal{I}_i$ and $\mathit{Body}^-(r_i\theta_i) \cap \mathit{ELP}(\mathcal{M}) = \emptyset$.

First, we consider Condition 1. The complexity of Condition 1.1 is in P. To check the complement of Condition 1.2, guess an $r \in P_{O'}$, for $O' \in D^{\mathcal{R}}_O$, and a substitution θ of the free variables of r over $V_{O',\mathcal{R}}$. Then, check if $\mathit{Satisfies}(\mathcal{R}, O, \mathcal{M}, O', \mathit{Cond}(r\theta))$=TRUE and $\mathit{Satisfies}(\mathcal{R}, O, \mathcal{M}, O', \mathit{Concl}(r\theta))$=FALSE. The complexity of this ckeck is in P. Thus, the complexity of the complement of Condition 1.1. is in NP. Therefore, the complexity of Condition 1.2. is in co-NP. To check the complement of Condition 1.3, guess an $r \in \Pi^{H_\mathcal{R}}_{O'}$, for $O' \in D^{\mathcal{R}}_O$, and a substitution θ of the free variables of r over $V_{O',\mathcal{R}}$. Then, check if $\mathit{ELP}(\mathcal{M})$ does not satisfy $r\theta$. The complexity of this check is in P. Thus, the complexity of the complement of Condition 1.3 is in NP. Therefore, the complexity of Condition 1.3 is in co-NP.

Now, we consider Condition 2. Note that the size of $\mathit{ELP}(\mathcal{M})$ is in P w.r.t. $\mathit{size}(O, \mathcal{R})$. Additionally, note that the size of each founded proof of an $L \in \mathit{ELP}(\mathcal{M})$ is in P w.r.t. $\mathit{size}(O, \mathcal{R})$. The complexity of Condition 2.2 is in NP. This is because the complexity of the steps (2-4) of Algorithm $\mathit{MiddleNotSatisfies}(\mathcal{R}, O, O', \mathcal{I}_i, \mathcal{M}, \mathit{Cond}(r_i\theta_i))$ is in P. The complexity of Condition 2.3 is in P. Therefore, the complexity of Condition 2 is in P^{NP}.

Thus, Conditions 1 and 2 can be solved by P, NP, and P^{NP} oracle calls. Note that $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{P}^{\mathrm{NP}}$. Thus, the complexity of the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is in $\mathrm{NP}^{\mathrm{P}^{\mathrm{NP}}}$=$\mathrm{NP}^{\mathrm{NP}}$ w.r.t. $\mathit{size}(O, \mathcal{R})$.

(ii)
Hardness) It follows directly from the hardness part of Proposition 13.
Membership)
Case F is an r-ERDF formula over $\{\mathit{Nam}_{O'} \mid O' \in D^{\mathcal{R}}_O\}$: Assume that we want to verify

if $O \not\models^{\mathsf{st}}_{\mathcal{R}} v(F)$. Guess now a modular semi-Herbrand interpretation I of O w.r.t. $\mathcal{R}$. Then, test if $\mathsf{I} \in \mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}}$ (Step 1) as in the proof of the membership part of Proposition 15(i). If *Satisfies*$(\mathcal{R}, O, \mathsf{I}, O, v(F))$ =FALSE (Step 2) then $O \not\models^{\mathsf{st}}_{\mathcal{R}} v(F)$. The complexity of Step 1 is in P^{NP} w.r.t. *size*$(O, \mathcal{R})$ and the complexity of Step 2 is polynomial w.r.t. *size*$(O, \mathcal{R})$. Therefore, the complexity of checking whether $O \not\models^{\mathsf{st}}_{\mathcal{R}} v(F)$ is in $\mathrm{NP}^{\mathrm{NP}}$ w.r.t. *size*$(O, \mathcal{R})$. Thus, the complexity of deciding whether $O \models^{\mathsf{st}}_{\mathcal{R}} v(F)$ is in co-$\mathrm{NP}^{\mathrm{NP}}$ w.r.t. *size*$(O, \mathcal{R})$.

Case F is not an $\mathtt{r}$-ERDF formula over $\{Nam_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$: We want to verify if $O \models^{\mathsf{st}}_{\mathcal{R}} v(F)$. This is true only if O has no modular stable model w.r.t. $\mathcal{R}$. It follows from Proposition 15(i) that the complexity of this problem is in co-$\mathrm{NP}^{\mathrm{NP}}$ w.r.t. *size*$(O, \mathcal{R})$. □

Proposition 16 Let $O \in \mathcal{R}$ and let F be an $\mathtt{r}$-ERDF formula. Additionally, let v be (i) "yes", if $Var(F) = \emptyset$, or (ii) a mapping $v : Var(F) \rightarrow V_{O,\mathcal{R}}$, if $Var(F) \neq \emptyset$. Then: (i) the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is PSPACE-complete w.r.t. *size*$(O, \mathcal{R})$, and (ii) the problem of establishing whether $v \in Ans^{\mathsf{st}}_{O,\mathcal{R}}(F)$ is PSPACE-complete w.r.t. *size*$(O, \mathcal{R})$.

Proof:
(i)
Hardness) Let $O' = \langle G', P' \rangle$ be an ERDF ontology and let O'' be an ontology such that $G_{O''} = G'$, $P_{O''} = P'$, and $Int_{O''} = \{\}$. Let $\mathcal{R}'' = \{O''\}$. In Proposition 21.1 of [5], we show that the problem of establishing whether O' has an $\#n_{O''}$-stable model is PSPACE-hard w.r.t. the size of O'. In fact, this hardness result is based on a reduction from the *fully quantified boolean formula* (*QBF*) problem, which is a PSPACE-complete problem [69, 57]. It follows from Corollary 1 that O' has an $\#n_{O''}$-stable model iff O'' has a modular stable model w.r.t. $\mathcal{R}''$. Note that to generate O'' and R'' from O', it takes polynomial time and that *size*$(O'', \mathcal{R}'')$ is polynomial to the size of O'. Therefore, the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is PSPACE-hard w.r.t. *size*$(O, \mathcal{R})$.

Membership)
Guess now a modular semi-Herbrand interpretation $\mathcal{M}$ of O w.r.t. $\mathcal{R}$. It is the case that $\mathsf{M} \in \mathcal{M}^{\mathsf{st}}_{O,\mathcal{R}}$ iff Condition 1 and Condition 2 in the proof of the membership part of Proposition 15(i) hold.

First, we consider Condition 1. The complexity of Condition 1.1 is in P. To check the complement of Condition 1.2, guess an $r \in P_{O'}$, for $O' \in D^{\mathcal{R}}_{O}$, and a substitution θ of the free variables of r over $V_{O',\mathcal{R}}$. Then, check if *Satisfies*$(\mathcal{R}, O, \mathcal{M}, O', Cond(r\theta))$=TRUE and *Satisfies*$(\mathcal{R}, O, \mathcal{M}, O', Concl(r\theta))$=FALSE. The complexity of this ckeck is in PSPACE (due to the possible quantifiers appearing in $Cond(r)$). Thus, the complexity of the complement of Condition 1.1. is in NPSPACE= PSPACE. Therefore, the complexity of Condition 1.2. is in co-PSPACE=PSPACE. To check the complement of Condition 1.3, guess an $r \in \Pi^{H_{\mathcal{R}}}_{O'}$, for $O' \in D^{\mathcal{R}}_{O}$, and a substitution θ of the free variables of r over $V_{O',\mathcal{R}}$. Then, check if $ELP(\mathcal{M})$ does not satisfy $r\theta$. The complexity of this check is in P. Thus, the complexity of the complement of Condition 1.3 is in NP. Therefore, the complexity of Condition 1.3 is in co-NP $\subseteq$ PSPACE.

Now, we consider Condition 2. Note that the size of $ELP(\mathcal{M})$ is in P w.r.t. *size*$(O, \mathcal{R})$. Additionally, note that the size of each founded proof of an $L \in ELP(\mathcal{M})$ is in P w.r.t. *size*$(O, \mathcal{R})$. The complexity of Condition 2.2 is in NPSPACE. This is because the complexity of the steps (2-4) of Algorithm *MiddleNotSatisfies*$(\mathcal{R}, O, O', \mathcal{I}_i, \mathcal{M}, Cond(r_i\theta_i))$ is in PSPACE (due to the possible quantifiers appearing in $Cond(r_i)$). The complexity of Condition 2.3 is in P. Therefore, the complexity of Condition 2 is in $\mathrm{P}^{\mathrm{NPSPACE}}$=PSPACE.

Thus, Conditions 1 and 2 can be solved by P and PSPACE oracle calls. Thus, the complexity of the problem of establishing whether O has a modular stable model w.r.t. $\mathcal{R}$ is in $\mathrm{NP}^{\mathrm{PSPACE}} \subseteq \mathrm{PSPACE}^{\mathrm{PSPACE}}$=PSPACE w.r.t. $size(O, \mathcal{R})$.

(ii)
Hardness) Let $F = p(s, o) \wedge \neg p(s, o)$. Then, $O \models^{\mathtt{st}}_{\mathcal{R}} F$ iff O has no modular stable model w.r.t. $\mathcal{R}$. It follows from Proposition 16(i) that the complexity of deciding whether O has a modular stable model w.r.t. $\mathcal{R}$ is PSPACE-hard w.r.t. $size(O, \mathcal{R})$. Therefore, the complexity of deciding whether $O \models^{\mathtt{st}}_{\mathcal{R}} F$ is co-PSPACE-hard w.r.t. $size(O, \mathcal{R})$. But co-PSPACE=PSPACE.

Membership)
Case F is an $\mathtt{r}$-ERDF formula over $\{Nam_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$: Assume that we want to verify if $O \not\models^{\mathtt{st}}_{\mathcal{R}} v(F)$. Guess now a modular semi-Herbrand interpretation I of O w.r.t. $\mathcal{R}$. Then, test if $\mathsf{I} \in \mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$ (Step 1) as in the proof of the participation part of Proposition 16(i). If *Satisfies*$(\mathcal{R}, O, \mathsf{I}, O, v(F))$ =FALSE (Step 2) then $O \not\models^{\mathtt{st}}_{\mathcal{R}} v(F)$. The complexity of Step 1 is in $\mathrm{P}^{\mathrm{NPSPACE}} \subseteq$ PSPACE w.r.t. $size(O, \mathcal{R})$ and the complexity of Step 2 is in P w.r.t. $size(O, \mathcal{R})$. Therefore, the complexity of checking whether $O \not\models^{\mathtt{st}}_{\mathcal{R}} v(F)$ is in $\mathrm{NP}^{\mathrm{PSPACE}} \subseteq$PSPACE w.r.t. $size(O, \mathcal{R})$. Thus, the complexity of deciding whether $O \models^{\mathtt{st}}_{\mathcal{R}} v(F)$ is in co-PSPACE=PSPACE w.r.t. $size(O, \mathcal{R})$.
Case F is not an $\mathtt{r}$-ERDF formula over $\{Nam_{O'} \mid O' \in D^{\mathcal{R}}_{O}\}$: We want to verify if $O \models^{\mathtt{st}}_{\mathcal{R}} v(F)$. This is true only if O has no modular stable model w.r.t. $\mathcal{R}$. It follows from Proposition 16(i) that the complexity of this problem is in co-PSPACE=PSPACE w.r.t. $size(O, \mathcal{R})$. □

Appendix C: Table of Symbols

List of Symbols	
Symbol	*Description*
$\mathcal{PL}$	the set of plain literals
$\mathcal{TL}$	the set of typed literals
V_G	the set of URI references and literals appearing in the ERDF graph G
V_P	the set of URI references and literals appearing in the r-ERDF program P
$sk(G)$	the skolemization of an ERDF graph G
$Cond(r)$	the condition of an r-ERDF rule r
$Concl(r)$	the conclusion of an r-ERDF rule r
Nam_O	the name of an ontology O
$L_O = \langle G_O, P_O \rangle$	the logic of an ontology O
$Export^{\mathtt{t}}_O$	a (partial) function from $V \cap URI$ to $\mathcal{P}(\mathcal{O}_{\mathtt{nam}} - \{Nam_O\}) \cup \{*\}$
$Exported^{\mathtt{t}}_O$	$\{x \mid Export^{\mathtt{t}}_O(x)$ is defined $\}$
$Import^{\mathtt{t}}_O$	a (partial) function from $V \cap URI$ to $\mathcal{P}(\mathcal{O}_{\mathtt{nam}} - \{Nam_O\}) \cup \{*\}$
$Imported^{\mathtt{t}}_O$	$\{x \mid Import^{\mathtt{t}}_O(x)$ is defined $\}$
$Export^{\mathtt{pr}}_{O,\mathcal{R}}(x)$	the ontologies in $\mathcal{R}$ to which O is willing to export property x
$Export^{\mathtt{cl}}_{O,\mathcal{R}}(x)$	the ontologies in $\mathcal{R}$ to which O is willing to export class x
$Import^{\mathtt{pr}}_{O,\mathcal{R}}(x)$	denotes the ontologies in $\mathcal{R}$ from which O imports property x
$Import^{\mathtt{cl}}_{O,\mathcal{R}}(x)$	denotes the ontologies in $\mathcal{R}$ from which O imports class x
$D^{\mathcal{R}}_O$	the dependencies of O w.r.t. $\mathcal{R}$
$\mathcal{V}^{\#n}_{RDF}$	$\mathcal{V}_{RDF} - \{rdf{:}\_i \mid i > n\}$
$V^{\#n}_O$	$V_{sk(G_O)} \cup V_{P_O} \cup Exported^{\mathtt{pr}}_O \cup Exported^{\mathtt{cl}}_O \cup Imported^{\mathtt{pr}}_O \cup Imported^{\mathtt{cl}}_O \cup V^{\#n}_{RDF}$ $\cup \mathcal{V}_{RDFS} \cup \mathcal{V}_{ERDF}$
$V_{O,\mathcal{R}}$	$\cup\{V^{\#n_O}_{O'} \mid O' \in D^{\mathcal{R}}_O\}$
$Res^{\mathtt{H}}_{O,\mathcal{R}}$	$V_{O,\mathcal{R}}$ with the well-typed XML literals substituted by their XML values
$\mathcal{I}^{\mathtt{H}}_{O,\mathcal{R}}$	the set of modular Herbrand interpretations of O w.r.t. $\mathcal{R}$
$\mathcal{M}^{\mathtt{H}}_{O,\mathcal{R}}$	the set of modular Herbrand models of O w.r.t. $\mathcal{R}$
$\mathcal{M}^{\mathtt{st}}_{O,\mathcal{R}}$	the set of modular stable models of O w.r.t. $\mathcal{R}$
$[P]_V$	instantiation of all rules of an $\mathtt{r}$-ERDF program P according to vocabulary V
$[P_{O'}]$	abbreviation of $[P_{O'}]_{V_{O',\mathcal{R}}}$
$Ans^{\mathtt{st}}_{O,\mathcal{R}}(F)$	the modular stable answers of F w.r.t. O and $\mathcal{R}$
$\Pi_{O,\mathcal{R}}$	$\bigcup\{\Pi^{O'}_{G_{O'}} \cup [\Pi^{O'}_{P_{O'}}]_{V_{O',\mathcal{R}}} \cup [\Pi^{H_{\mathcal{R}}}_{O'}]_{V_{O',\mathcal{R}}} \mid O' \in D^{\mathcal{R}}_O\}$
$[\Pi^{O'}_{P_{O'}}]$	abbreviation for $[\Pi^{O'}_{P_{O'}}]_{V_{O',\mathcal{R}}}$
$[\Pi^{H_{\mathcal{R}}}_{O'}]$	abbreviation for $[\Pi^{H_{\mathcal{R}}}_{O'}]_{V_{O',\mathcal{R}}}$
$\mathtt{G}^{\mathcal{R}}_O$	abbreviation for $\cup_{O' \in D^{\mathcal{R}}_O} \Pi^{O'}_{G_{O'}}$
$\mathtt{H}^{\mathcal{R}}_O$	abbreviation for $\cup_{O' \in D^{\mathcal{R}}_O} [\Pi^{H_{\mathcal{R}}}_{O'}]$
$\mathtt{P}^{\mathcal{R}}_O$	abbreviation for $\cup_{O' \in D^{\mathcal{R}}_O} [\Pi^{O'}_{P_{O'}}]$

Table 1. Symbols and Description

References

1. M. Alves, C. V. Damásio, W. Nejdl, and D. Olmedilla. A Distributed Tabling Algorithm for Rule Based Policy Systems. In *7th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY-2006)*, pages 123–132, 2006.
2. A. Analyti, G. Antoniou, and C. V. Damásio. A Principled Framework for Modular Web Rule Bases and Its Semantics. In *11th International Conference on Knowledge Representation and Reasoning (KR-2008)*, pages 390–400, 2008.
3. A. Analyti, G. Antoniou, and C. V. Damasio. A Formal Theory for Modular ERDF Ontologies. In *3rd International Conference Web Reasoning and Rule Systems (RR 2009)*, pages 212–226, 2009.
4. A. Analyti, G. Antoniou, and C. V. Damasio. MWeb: a Principled Framework for Modular Web Rule Bases and its Semantics. *ACM Transactions on Computational Logic (ACM TOCL)*, 12(2), 2011.
5. A. Analyti, G. Antoniou, and C. V. Damásio. Extended RDF: Computability and Complexity Issues. In *submitted for publication*, 2012. Available at `http://www.ics.forth.gr/~analyti/Papers_2/ERDF_Complexity.pdf`.
6. A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner. Extended RDF as a Semantic Foundation of Rule Markup Languages. *Journal of Artificial Intelligence Research (JAIR)*, 32:37–94, 2008.
7. A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner. On the Computability and Complexity Issues of Extended RDF. In *10th Pacific Rim International Conference on Artificial Intelligence (PRICAI-2008)*, pages 5–16, 2008.
8. C. B. Aranda, M. Arenas, and Ó. Corcho. Semantics and Optimization of the SPARQL 1.1 Federation Extension. In *8th Extended Semantic Web Conference (ESWC-2011), part II*, pages 1–15, 2011.
9. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
10. J. Bao, D. Caragea, and V. Honavar. On the Semantics of Linking and Importing in Modular Ontologies. In *5th International Semantic Web Conference (ISWC'06)*, pages 72–86, 2006.
11. J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar. Package-Based Description Logics. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 349–371. 2009.
12. T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler. N3Logic: A Logical Framework For the World Wide Web. *Theory and Practice of Logic Programming (TPLP)*, 8(3):249–269, 2008.
13. H. Boley and M. Kifer. RIF Framework for Logic Dialects (Second Edition). W3C Recommendation 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-fld/`.
14. P. A. Bonatti, J. L. D. Coi, D. Olmedilla, and L. Sauro. Rule-Based Policy Representations and Reasoning. In *Semantic Techniques for the Web, The REWERSE Perspective 2009*, pages 201–232. 2009.
15. P. A. Bonatti, J. L. D. Coi, D. Olmedilla, and L. Sauro. A Rule-Based Trust Negotiation System. *IEEE Transactions on Knowledge and Data Engineering*, 22(11):1507–1520, 2010.
16. A. Borgida and L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal Data Semantics*, 1:153–184, 2003.
17. G. Brewka and T. Eiter. Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In *22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, pages 385–390, 2007.
18. G. Brewka, T. Eiter, and M. Fink. Nonmonotonic Multi-Context Systems: A Flexible Approach for Integrating Heterogeneous Knowledge Sources. In *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, pages 233–258, 2011.
19. J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs. *Journal of Web Semantics*, 3(4), 2005.
20. J. L. D. Coi and D. Olmedilla. A Review of Trust Management, Security and Privacy Policy Languages. In *International Conference on Security and Cryptography (SECRYPT-2008)*, pages 483–490, 2008.

21. C. V. Damasio, A. Analyti, and G. Antoniou. Embeddings of Simple Modular Extended RDF. In *4th International Conference Web Reasoning and Rule Systems (RR 2010), (short paper)*, 2010.
22. C. V. Damasio, A. Analyti, and G. Antoniou. Implementing Simple Modular ERDF Ontologies. In *19th European Conference on Artificial Intelligence (ECAI-2010), (short paper)*, 2010.
23. C. V. Damásio, A. Analyti, and G. Antoniou. Modularity in the Rule Interchange Format. In *5th International Symposium on Rule-Based Reasoning, Programming, and Applications (RULEML-2011)*, pages 313–328, 2011.
24. J. de Bruijn and C. Welty. RIF RDF and OWL Compatibility. W3C Recommendation 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-rdf-owl/`.
25. T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. *Artificial Intelligence*, 172(12-13), 2008.
26. F. Ensan. Formalizing Ontology Modularization through the Notion of Interfaces. In *16th International Conference on Knowledge Engineering: Practice and Patterns (EKAW-2008)*, pages 74–82, 2008.
27. W. Faber, G. Pfeifer, and N. Leone. Semantics and complexity of recursive aggregates in answer set programming. *Artificial Intelligence*, 175(1), 2011.
28. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett. No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web. In *1st European Semantic Web Symposium (ESWS 2004)*, pages 342–356, 2004.
29. A. V. Gelder, K. A. Ross, and J. S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
30. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. A. Bowen, editors, *5th International Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
31. M. Gelfond and V. Lifschitz. Logic programs with Classical Negation. In *7th International Conference on Logic Programming*, pages 579–597, 1990.
32. C. Ghidini and L. Serafini. Reconciling Concepts and Relations in Heterogeneous Ontologies. In *3rd European Semantic Web Conference (ESWC 2006)*, pages 50–64, 2006.
33. C. Ghidini and L. Serafini. Mapping Properties of Heterogeneous Ontologies. In *13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2008)*, pages 181–193, 2008.
34. C. Ghidini, L. Serafini, and S. Tessaris. On Relating Heterogeneous Elements from Different Ontologies. In *2007 International Workshop on Description Logics (DL2007)*, 2007.
35. C. Ghidini, L. Serafini, and S. Tessaris. Complexity of reasoning with expressive ontology mappings. In *Fifth International Conference on Formal Ontology in Information Systems (FOIS-2008)*, pages 151–163, 2008.
36. C. Golbreich and E. K. Wallace. OWL 2 Web Ontology Language New Features and Rationale (Second Edition). W3C Recommendation 11 December 2012. Latest version available at `http://www.w3.org/TR/owl2-new-features/`.
37. B. C. Grau and O. Kutz. Modular Ontology Languages Revisited. In *IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa'07)*, 2007.
38. B. C. Grau, B. Parsia, and E. Sirin. Working with Multiple Ontologies on the Semantic Web. In *Third International Semantic Web Conference (ISWC-2004)*, pages 620–634, 2004.
39. B. C. Grau, B. Parsia, and E. Sirin. Ontology integration using epsilon-connections. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, pages 293–320. Springer, 2009.
40. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *12th International World Wide Web Conference (WWW2003)*, pages 48–57, 2003.
41. S. Harris and A. Seaborne. SPARQL 1.1 Query Language. W3C Working Draft 24 July 2012 available at `http://www.w3.org/TR/2012/WD-sparql11-query-20120724/`.
42. P. Hayes. RDF Semantics. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`.

43. H. Herre, J. Jaspars, and G. Wagner. Partial Logics with Two Kinds of Negation as a Foundation of Knowledge-Based Reasoning. In D. M. Gabbay and H. Wansing, editors, *What Is Negation?* Kluwer Academic Publishers, 1999.
44. P. Hitzler, M. Krotzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation 11 December 2012. Latest version available at `http://www.w3.org/TR/owl2-primer/`.
45. T. Janhunen, E. Oikarinen, H. Tompits, and S. Woltran. Modularity Aspects of Disjunctive Stable Models. *Journal of Artificial Intelligence Research (JAIR)*, 35:813–857, 2009.
46. M. Kifer. Nonmonotonic Reasoning in FLORA-2. In *8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2005)*, pages 1–12, 2005.
47. M. Kifer and H. Boley. RIF Overview (Second Edition). W3C Working Group Note 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-overview/`.
48. M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843, 1995.
49. G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`.
50. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of Abstract Description Systems. *Artificial Intelligence*, 156(1):1–73, 2004.
51. N. Li and J. Mitchell. RT: A Role-based Trust-management Framework. In *DARPA Information Survivability Conference and Exposition (DISCEX III)*, 2003.
52. N. Li and J. C. Mitchell. DATALOG with Constraints: A Foundation for Trust Management Languages. In *5th International Symposium on Practical Aspects of Declarative Languages (PADL-2003)*, pages 58–73, 2003.
53. S. Liang, P. Fodor, H. Wan, and M. Kifer. OpenRuleBench: an Analysis of the Performance of Rule Engines. In *18th International Conference on World Wide Web (WWW-2009)*, pages 601–610, 2009.
54. J. W. Lloyd and R. W. Topor. Making Prolog more Expressive. *Journal of Logic Programming*, 1(3):225–240, 1984.
55. I. Niemelä and P. Simons. Smodels - An Implementation of the Stable Model and Well-Founded Semantics for Normal LP. In *4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'97)*, pages 421–430, 1997. Available at `http://www.tcs.hut.fi/Software/smodels/`.
56. E. Oikarinen and T. Janhunen. Achieving compositionality of the stable model semantics for smodels programs. *Theory and Practice of Logic Programming (TPLP)*, 8(5-6):717–761, 2008.
57. C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
58. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Transanctions on Database Systems*, 34(3):16:1–16:45, 2009.
59. A. Polleres. From SPARQL to rules (and back). In *16th International Conference on World Wide Web (WWW-2007)*, pages 787–796. ACM, 2007.
60. A. Polleres, C. Feier, and A. Harth. Rules with Contextually Scoped Negation. In *3rd European Semantic Web Conference (ESWC-2006)*, pages 332–347, 2006.
61. A. Polleres and J. Wallner. On the Relation between SPARQL1.1 and Answer Set Programming. *Journal of Applied Non-Classical Logics (JANCL), Special issue on Equilibrium Logic and Answer Set Programming*, 2013. To appear.
62. E. Prudhommeaux and C. Buil-Aranda. SPARQL 1.1 Federated Query. W3C Working Draft 17 November 2011 available at `http://www.w3.org/TR/2011/WD-sparql11-federated-query-20111117/`.
63. K. F. Sagonas, T. Swift, and D. S. Warren. XSB as an Efficient Deductive Database Engine. In R. T. Snodgrass and M. Winslett, editors, *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 442–453. ACM Press, 1994. Available at `http://xsb.sourceforge.net/`.
64. S. Schaffert, F. Bry, P. Besnard, H. Decker, S. Decker, C. F. Enguix, and A. Herzig. Paraconsistent Reasoning for the Semantic Web. In *Workshop on Uncertainty Reasoning for the Semantic Web, co-located with ISWC-2005*, pages 104–105, 2005.

65. S. Schenk and S. Staab. Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. In *17th International Conference on World Wide Web (WWW-2008)*, pages 585–594, 2008.
66. L. Serafini, A. Borgida, and A. Tamilin. Aspects of Distributed and Modular Ontology Reasoning. In *19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 570–575, 2005.
67. M. Sintek and S. Decker. TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In *1st International Semantic Web Conference (ISWC-2002)*, pages 364–378. Springer-Verlag, 2002.
68. W. N. Stefan Decker, Michael Sintek. The Model-Theoretic Semantics of TRIPLE. Technical Report, 2002.
69. L. J. Stockmeyer and A. R. Meyer. Word Problems Requiring Exponential Time: Preliminary Report. In *Fifth Annual ACM Symposium on Theory of Computing (STOC'73)*, pages 1–9, 1973.
70. H. J. ter Horst. Completeness, Decidability and Complexity of Entailment for RDF Schema and a Semantic Extension Involving the OWL Vocabulary. *Journal of Web Semantics*, 3(2-3):79–115, 2005.
71. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation 11 December 2012. Latest version available at `http://www.w3.org/TR/owl2-overview/`.
72. G. Yang, M. Kifer, and C. Zhao. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *2nd International Conference on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE'03)*, pages 671–688, 2003.
73. A. Zimmermann. Integrated Distributed Description Logics. In *2007 International Workshop on Description Logics (DL-2007)*, 2007.