Modularity in the Rule Interchange Format

C. V. Damásio[1] and A. Analyti[2] and G. Antoniou[3]

[1] CENTRIA, Departamento de Informática Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal.
cd@di.fct.unl.pt

[2] Institute of Computer Science, FORTH-ICS, Crete, Greece
analyti@ics.forth.gr

[3] Institute of Computer Science, FORTH-ICS, and
Department of Computer Science, University of Crete, Crete, Greece
antoniou@ics.forth.gr

Abstract. The adoption of standards by the knowledge representation and logic programming communities is essential for their visibility and impact. The Rule Interchange Format is a fundamental effort in this direction that should be supported by users, developers and theoreticians. For this reason, it is essential to the community to discuss the recommendations published by the W3C RIF Working Group. In particular, this paper presents the semantics of Rule Interchange Format (RIF) of multi-documents, analyses it and some deficiencies are elicited. A more general approach is proposed as an alternative semantics for multi-documents. As a side important result, some relevant problems in the semantics of RIF-FLD are also discussed and possible ways out are proposed.

Keywords: Semantic Web, Rule Interchange Format, Modularity, Logic Programming, Language Issues, Many-valued Semantics

1 Introduction

The development of the Semantic Web requires appropriate modular constructs for the combination of rulebases and ontologies. This is an essential problem to software engineering and its difficulty rockets with the distributed and woven character of the Semantic Web. The simplest mechanism is the basic importing of ontologies or rulebases, like the ones available in the Web Ontology Language [21, 20] and Rule Interchange Format [27]. Others address the problem of extracting and reusing subsets of ontologies in order to avoid using irrelevant parts of knowledge and thus reducing complexity of reasoning (e.g. see [14, 16, 18] and references therein). Packaged-Based Description Logic [6] is a general approach of contextualized interpretations of ontologies with semantic mappings among names, addressing several fundamental issues in a theoretical sound way. Compositionality and modularity of reasoning for logic programming languages has been studied for a long time [8] and recently have obtained novel results for Answer Set based semantics [19, 13]. Finally, the syntactic modular constructs for logic programming systems for programming-in-the-large have been extensively analysed and several approaches appear in the literature (e.g. [17, 5]).

This paper results from the attempt to align the syntax and semantics of our Modular Web Framework [4, 5], which allows the safe and controlled use of weak negation in the Semantic Web, with the general RIF Framework for Logic Dialects (RIF-FLD [27]). Unfortunately, this proved to be impossible due to general problems identified both in the semantics of ordinary formulas and of document formulas in RIF-FLD. Here we report the identified problems, discuss and advance proposals for their correction.

The Rule Interchange Format Framework for Logic Dialects [27] proposes two forms of structuring logical documents, via import and module mechanisms. The import mechanism allows the use of profiles in order to include in a RIF logical theory other logical theories with a specific semantics or syntax, e.g. RDF or OWL. The semantics of import with profiles is left open for the RIF dialects to define the intended behaviour. The import mechanism without profile corresponds roughly to the textual import of the contents of RIF documents, modulo local constants, and will be discussed in detail in this paper. The module mechanism allows the remote linking of external theories to the enclosing RIF document where it is used, and will also be analysed.

This paper start by illustrating the MWeb approach with a motivating example. Next, we present the basics of the RIF-FLD semantics, and in particular describing the semantics of some connectives and elicit their problems. Subsequently we describe the syntax of RIF document formulas, and overview in detail the corresponding semantics and its singular features. The deficiencies of the current proposal will be pointed out and an alternative semantics encompassing solution will be provided. The changes in the definition of some connectives require also a generalization of the notion of model, and the corresponding notion of logical entailment, which are briefly treated. Conclusions finish this work.

2 The MWeb Approach

The Modular Web framework [4, 5] addresses in a principled way several aspects of knowledge sharing and integration in the Semantic Web. The MWeb framework provides mechanisms supporting safe uses of non-monotonic negation in scoped closed and open world assumptions in logic rules for Semantic Web applications, under the full control of rulebase providers and consumers. In order to make the MWeb approach widely applicable, we have started the alignment with the RIF syntax as well as an attempt to semantically extend RIF-FLD framework to encompass MWeb's semantics. Several issues have been elicited in this process, which are better explained with a concrete MWeb rulebase example.

Typical MWeb rulebases can be found in figures 1 and 2 where Geographical data is provided and used to describe information about universities[4]. Every

[4] The examples can be tried out with the implementation publicly available at `http://centria.di.fct.unl.pt/~cd/mweb`

MWeb rulebase has an interface, and a logical document part where knowledge is specified by facts and rules. The current MWeb syntax is akin to RIF's presentation syntax, where the # operator expresses class membership in RIF, while ## subclass inclusion. Moreover, we also allow F-logic like frames of the form $o.[p_1\text{-»}v_1, \ldots, p_2\text{-»}v_n, \ldots]$ (equivalent to $o.[p_1\text{-»}v_1]$, and $o.[p_2\text{-»}v_2]$, and ...) represent that object o is related via property p_1 to value v_1, etc. With this information, the meaning of the logical part of both rulebases is immediate.

GEO interface (geo.mw)

```
:- rulebase 'http://geography.int'.

:- prefix xsd = 'http://www.w3.org/2001/XMLSchema#'.
:- prefix geo='http://geography.int#'.
:- import('rdf.mw',interface).

:- defines local closed class(geo:Continent).
:- defines local open class(geo:Country) wrt context class(geo:PoliticalEntity).
:- defines local open ?X.[ rdf:type -» geo:Country ] wrt context class(geo:PoliticalEntity).
:- defines local definite class(geo:City), class(geo:PoliticalEntity).
:- defines local definite property(geo:located_in), property(geo:part_of),
                          property(geo:city_name).
```

GEO rulebase (geo.rb)

```
:- import('rif.rb', rulebase).
:- import('rdf.rb', rulebase ).

geo:Africa # geo:Continent.           geo:America # geo:Continent.
geo:Antarctica # geo:Continent.           geo:Asia # geo:Continent.
geo:Europe # geo:Continent.           geo:Oceania # geo:Continent.

geo:Country ## geo:PoliticalEntity.

geo:EU.[ geo:part_of -» geo:Europe, rdf:type -» geo:PoliticalEntity].
geo:Spain.[ geo:part_of -» geo:EU, rdf:type -» geo:Country ].
geo:Portugal.[ geo:part_of -» geo:EU, rdf:type -» geo:Country ].
geo:Cataluna.[ geo:part_of -» geo:Spain, rdf:type -» geo:PoliticalEntity ].

geo:Barcelona.[ geo:located_in -» geo:Cataluna, rdf:type -» geo:City,
                geo:city_name -» "Barcelona"^^xsd:string ].
geo:Lisboa.[ geo:located_in -» geo:Portugal, rdf:type -» geo:City,
             geo:city_name -» "Lisboa"^^xsd:string, geo:city_name -» "Lisbon"^^xsd:string ].

?X.[ geo:located_in -» ?Z ] :- ?Y.[ geo:part_of -» ?Z ], ?X.[ geo:located_in -» ?Y ].
```

Fig. 1. Geographic MWeb Rulebase

The interface document provides an Internationalized Resource Identifier (IRI) for the rulebase, an optional base IRI address as well as prefixes for shortening writing of IRIs. More important, the interface can (textually) import other interfaces. In Fig. 1 and Fig. 2, an interface defining the vocabulary of the RDF language is imported (not shown). An optional `vocabulary` declaration can be used to list the vocabulary of the rulebase. Next, follow two blocks of declarations. The first block defines the predicates being defined in the MWeb rulebase, and correspond to a generalization of the export declarations found in logic programming based languages. The second block corresponds to a generalization of import declarations, as shown in the rulebase of Fig. 2.

INST interface (inst.mw)

```
:- rulebase 'http://institution.int'.
:- prefix xsd = 'http://www.w3.org/2001/XMLSchema#'.
:- prefix geo='http://geography.int#'.
:- prefix inst = 'http://institution.int#'.
:- import('rdf.mw',interface).

:- defines local normal class(inst:Institution).
:- defines local normal inst:address(?ID,?NAME,?STREET,?NUMBER,?CITY,?COUNTRY).

:- uses definite class(geo:Country).
:- uses definite property(geo:located_in), property(geo:city_name).
:- uses definite property(rdf:type) from 'http://geography.int#'.
```

INST rulebase (inst.rb)

```
:- import('rdf.rb', rulebase ).

:- defines internal definite inst:address/5.

?ID # inst:Institution :- inst:address( ?ID, ?_, ?_, ?_, ?_ ).

inst:address(inst:UBAR,"Universitat Barcelona"^^xsd:string,
    "Calle 1"^^xsd:string,2,"Barcelona"^^xsd:string).
inst:address(inst:UNL, "Universidade Nova de Lisboa"^^xsd:string,
    "Rua 2"^^xsd:string,3,"Lisboa"^^xsd:string).

inst:address(?ID,?NAME,?STREET,?NUMBER,?CITY,?COUNTRY) :-
    inst:address(?ID,?NAME,?STREET,?NUMBER,?CITYNAME),
    ?CITY.[ geo:city_name ->> ?CITYNAME ],
    ( ?CITY.[ geo:located_in ->> ?COUNTRY ] ) @ 'http://geography.int',
    (?COUNTRY.[ rdf:type ->> geo:Country ] ) @ 'http://geography.int'.
```

Fig. 2. Institutional MWeb Rulebase

The interesting feature of the MWeb framework is that besides scope (i.e. internal, local, or global), different reasoning modes can be associated to predicates (i.e. definite, open, closed, or normal). This allows control of monotonicity of reasoning by the producer and consumer of the knowledge. In what follows, we use the term predicates to mean an ordinary predicate, a class or a property.

Global and local predicates are visible in the Semantic Web, the difference being that local predicates can only be declared in a single rulebase; internal predicates are not visible. Normal predicates are general predicates which can use weak negation in the bodies, and therefore are non-monotonic. The remaining predicates cannot use weak negation in the bodies of rules, but strong negation is allowed. Closed predicates are used to make closed-world assumptions with respect to the rulebase vocabulary, or with respect to the provided context in the declaration. For instance, declaring the class `geo:Continent` closed means that everything which is not concluded to be a continent in the geographic MWeb rulebase of Fig. 1 is assumed to be non-continent. In practice, this corresponds to define the class by the program containing the facts in the program plus the extra logical rule: `neg ?X # geo:Continent :- naf ?X # geo:Continent.`

The rule expresses that every ?X which cannot be proved to be a continent (naf ?X # geo:Continent) then it is known not to be a continent (neg ?X # geo:Continent). In intuitive terms, the above rule states that the list of continents provided is exhaustive. This illustrates the extra power of having two forms of negations, and why we require them in our MWeb language. Open predicates implement open-world assumptions, where unknown information can be either true or false. This is captured by a rule like the one above and its dual with respect to strong negation. Definite predicates are monotonic predicates which can use the monotonic strong negation. Also note in Fig. 1 that the `geo:Country` class is made open and exported in two ways, the last one in the form of frames representing RDF triples. The relationship between RIF predicate # and `rdf:type` frames is captured by the rules in `'rdf.rb'`, which is imported by `'geo.rb'`. The import of document `'rif.rb'` provides the rules for capturing the semantics of # and ##, and is redundant since it is imported as well in `'rdf.rb'`.

Regarding the `uses` declarations, a rulebase must specify the import mode of the predicate which is combined with the defining mode. In general, the predicate is obtained (called) at runtime from all rulebases defining it, except when an explicit rulebase list is provided specifying the providing rulebases (as in the last declaration of Fig. 2). A used predicate can be complemented with extra rules in the logical part. However, one can explicitly indicate the rulebase to call with the `@` operator and thus uses the original predicate, as shown in Fig. 2.

From this short presentation of MWeb, it is concluded that we require mechanisms to represent the interface, the logical document, import documents, call predicates in other modules, as well as two forms of negation. We will see that RIF-FLD presents difficulties/problems in any of these issues.

3 Basics of Rule Interchange Format Semantics

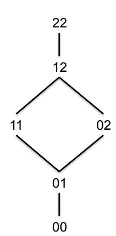
The Rule Interchange Framework for Logic Dialects [27] provides the general syntax and semantics for the rule languages to be used in the Semantic Web. The syntax and semantics is quite flexible and general having 13 distinct kinds of terms and 9 kinds of formulas, which can be further extended. It supports the usual conjunctive, disjunctive, rule, and quantified formulas and provides also two forms of negation, namely symmetric (strong) and default negation. Additionally, it includes document and remote formulas. Document formulas are used for defining the semantics of formulas in multi-documents (e.g. for defining the notion of module) and remote formulas are used for querying other documents inside a formula in a RIF document.

Specific dialects like RIF Core [26], RIF Basic Logic Dialect [23] have been specified but none uses default negation (designated `Naf` by RIF) nor symmetric negation (designated `Neg`). The other main recommendation being developed by W3C RIF group is RIF Production Rule Dialect [28] which uses only default negation over sets of ground facts, and cannot be applied to predicates defined by rules. Additionally, there have been defined some specialized RIF Dialects to cover the syntax of semantics of disjunctive logic programs under the stable model semantics RIF-CASPD ([24]) and well-founded semantics with default and strong (symmetric) negation RIF-CLPWD ([25]). However, the approach followed in the definitions of RIF-CASPD and RIF-CLPWD is against the current trends in the literature for defining the semantics of logic programs [22, 10, 9] because of the way RIF-FLD semantics has been defined. Namely, RIF-CASPD and RIF-CLPWD resort to an explicit quotient syntactic definition in order to obtain the intended models, while in the approaches of [22, 10, 9] this is fully captured model-theoretically by appropriate definitions of the truth-value lattices and interpretation of logical connectives. A "minimization" or preferential entailment is still necessary, as in RIF-FLD semantics.

The details of the syntax and semantics of RIF-FLD are long and involved, and cannot be fully presented in this paper. Therefore, we will focus on the relevant parts of the recommendation where we found some problems. For further information, the reader is referred to the standards defined by the RIF W3C Working Group [27] and to the recent overview [7]. RIF-FLD semantics is based on a complex semantic structure with more than a dozen kinds of different mappings, for capturing the intended meaning of each of the type of terms that the language allows. However, for our purpose is enough to analyse the underlying set of truth-values as well as the semantics of negations and rule formulas.

Definition 1 (Set of truth-values [27]). *Each RIF dialect must define the set of truth values, denoted by* $\mathbf{TV}$.

1. *This set must have a partial order, called the truth order, denoted* $\leq_t$. *In some dialects,* $\leq_t$ *can be a total order.*
2. *In addition,* $\mathbf{TV}$ *must be a complete lattice with respect to* $\leq_t$.
3. $\mathbf{TV}$ *is required to have two distinguished elements,* $\mathbf{f}$ *and* $\mathbf{t}$*, such that* $\mathbf{f} \leq_t elt$ *and* $elt \leq_t \mathbf{t}$ *for every* $elt \in \mathbf{TV}$.
4. $\mathbf{TV}$ *has an operator of negation,* $\sim: \mathbf{TV} \to \mathbf{TV}$*, such that*
   - $\sim$ *is a self-inverse function: applying* $\sim$ *twice gives the identity mapping.*
   - $\sim \mathbf{t} = \mathbf{f}$ *(and thus* $\sim \mathbf{f} = \mathbf{t}$*).*



$$x \wedge y = glb(x, y)$$
$$x \vee y = lub(x, y)$$

→	00	01	11	02	12	22
00	22	22	22	22	22	22
01	00	22	22	22	22	22
11	00	02	22	02	22	22
02	00	11	11	22	22	22
12	00	01	11	02	22	22
22	00	01	11	02	12	22

~	
00	22
01	11
11	11
02	00
12	00
22	00

Fig. 3. Truth values and truth-tables for Partial Equilibrium Logic (WFS) [9]

$\mathbf{TV} = \{-2 < -1 < 0 < +1 < +2\}$

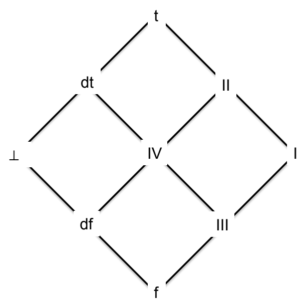
$x \wedge y = \min(x, y)$
$x \vee y = \max(x, y)$

$\rightarrow$	-2	-1	0	$+1$	$+2$
-2	$+2$	$+2$	$+2$	$+2$	$+2$
-1	$+2$	$+2$	$+2$	$+2$	$+2$
0	$+2$	$+2$	$+2$	$+2$	$+2$
$+1$	-1	-1	0	$+2$	$+2$
$+2$	-2	-1	0	$+1$	$+2$

$\sim$	
-2	$+2$
-1	$+2$
0	$+2$
$+1$	-1
$+2$	-2

$-$	
-2	$+2$
-1	$+1$
0	0
$+1$	-1
$+2$	-2

Fig. 4. Truth values and truth-tables for Equilibrium Logic (ASP) [22]



$\rightarrow$	f	df	III	$\bot$	IV	dt	I	II	t
f	t	t	t	t	t	t	t	t	t
df	t	t	t	t	t	t	t	t	t
III	t	t	t	t	t	t	t	t	t
$\bot$	t	t	t	t	t	t	t	t	t
IV	t	t	t	t	t	t	t	t	t
dt	t	t	t	t	t	t	t	t	t
I	f	f	f	f	f	f	t	t	t
II	f	f	f	f	f	f	t	t	t
t	f	f	f	f	f	f	t	t	t

$\sim$	
f	t
df	t
III	I
$\bot$	$\bot$
IV	I
dt	f
I	I
II	I
t	f

$-$	
f	t
df	dt
III	II
$\bot$	$\bot$
IV	IV
dt	df
I	I
II	III
t	f

Fig. 5. Truth values and truth-tables for Nine Logic ($WFSX_p$) [11]

Notice that the condition forcing the existence of elements **f** and **t** is redundant since this follows immediately from **TV** being a complete lattice. Conjunction and disjunction are interpreted as greatest lower bound (glb) and least upper bound (lub) in **TV**. The first significant remark is that the only sensible and widely adopted constraint imposed to negation operators has not been stated: a negation operator must be anti-monotonic, i.e. reverse the truth ordering. Formally, for every $elt_1, elt_2 \in \mathbf{TV}$ if $elt_1 \leq_t elt_2$ then $\sim elt_2 \ \leq_t \sim elt_1$. The other constraints are questionable, as we explain below.

The interpretation of the negations is then defined in RIF-FLD as follows, where $TVal_{\mathbf{I}}$ is the truth-valuation function from the set of formulas other than document formulas and remote formulas to the set of truth-values **TV**. Document formulas and remote terms are analysed in the next section.

$$\mathbf{TV} = \{F_0 < F_1 < F_2 < \ldots < 0 < \ldots < T_2 < T_1 < T_0\}$$

$$x \rightarrow y = \begin{cases} T_0 \text{ if } x \leq y \\ y \text{ otherwise} \end{cases}$$

$\sim$	F_0	F_1	$\ldots$	F_n	$\ldots$	0	$\ldots$	T_n	$\ldots$	T_1	T_0
	T_1	T_2	$\ldots$	T_{n+1}	$\ldots$	0	$\ldots$	F_{n+1}	$\ldots$	F_2	F_1

Fig. 6. Truth values and truth-tables of the Infinite Valued Semantics (WFS) [10]

Definition 2 (Truth-valuation for negations [27]). *Let Φ be a RIF well-formed formula other than a document formula or a remote formula, and $\mathbf{I}$ be a semantic structure.*

- $TVal_{\mathbf{I}}(\texttt{Neg Neg } \Phi) = TVal_{\mathbf{I}}(\Phi)$.
- $TVal_{\mathbf{I}}(\texttt{Naf } \phi) = \sim TVal_{\mathbf{I}}(\Phi)$.

While the interpretation of `Neg` is rather loose, just requiring that the double negation law is obeyed, on the contrary, the interpretation of `Naf` is too strong. The essential problem with the previous definition is that the negations of Equilibrium Logic [22] underlying Answer Set semantics [15] and extensions are ruled out by RIF-FLD semantics. The same happens to Well-founded Semantics with Explicit Negation and its extensions [2, 1, 12]. The reason is the same for all these semantics since the double negation law for default negation does not hold, as can be seen in Figures 3-6 (the truth-table for symmetric negation is $-$).

Moreover, the remaining condition imposing that the negation of true is false and vice-versa is not even obeyed by the infinite-valued logic [10] presented in Fig. 6, which can be seen as the proper model-theoretical definition of well-founded semantics. Thus, both conditions imposed by Def. 1 discard all the (recent) model-theoretical approaches found in the literature for the major semantics of logic programming. In fact, a common general condition that all these semantics obey is that negation is anti-monotonic, which is not enforced. The double negation law for strong negation holds in the shown cases, corresponding to the major semantics in the literature.

The semantics of rule-implication also brings problems to some of the above-mentioned semantics since one of the imposed constraints is again not obeyed.

Definition 3 (Truth-valuation for rule implication [27]). *Let Φ be a RIF well-formed formula other than a document formula or a remote formula, and $\mathbf{I}$ be a semantic structure.*

1. $TVal_{\mathbf{I}}(head \texttt{ :- } body) = \mathbf{t}$ *if* $TVal_{\mathbf{I}}(head) \geq_t TVal_{\mathbf{I}}(body)$;
2. $TVal_{\mathbf{I}}(head \texttt{ :- } body) <_t \mathbf{t}$ *otherwise.*

All the mentioned model-theoretical semantics for logic programming with negation(s) obey to the first condition imposed. However, the second condition is not satisfied by the Equilibrium Logic and by the Well-founded Semantics with Explicit Negation (the ones having strong negation – see figures 4 and 5). For this reason, we suggest discarding completely the second condition from the semantics of rule implication. This is related to the notion of logical entailment, which we will discuss subsequently. However, we require first the notion of multi-structure to be able to introduce the notion of logical entailment.

4 Rule Interchange Format Document Formulas

The RIF document formulas provide the general encapsulation of the RIF group of Formulas. Note that group formulas are not document formulas, and a document formula is associated to at most one group formula.

Definition 4 (Document and Group Formulas [27]). *The presentation syntax of document and group formulas of RIF-FLD is captured by the following EBNF grammar.*

```
Document ::= IRIMETA? 'Document'
      '(' Dialect? Base? Prefix* Import* Module* Group? ')'
Dialect  ::= 'Dialect' '(' Name ')'
Base     ::= 'Base' '(' ANGLEBRACKIRI ')'
Prefix   ::= 'Prefix' '(' NCName ANGLEBRACKIRI ')'
Import   ::= IRIMETA? 'Import' '(' LOCATOR PROFILE? ')'
Module   ::= IRIMETA? 'Module' '(' (Const | Expr) LOCATOR ')'
Group    ::= IRIMETA? 'Group' '(' (FORMULA | Group)* ')'
```

For our purposes we are interested solely in the `Import` and `Module` directives. The import directive has a mandatory `LOCATOR` which specifies where the corresponding document can be found. Locators include for instance $URLs$ but others can be defined by the dialect. The module directive has a an argument (a logical $Term$) before `LOCATOR` which will be used to refer to the module in remote terms of the form Φ @ $Term$, which can be found at `LOCATOR`.

A first issue is immediately recognized: RIF-FLD does not have the notion of interface. There are two easy solutions for this problem, always depending on using a document (formula) to express the interface information. A first solution relies on the use of meta-annotations to provide the missing information, while the other alternative requires assertion of specific formulas in a new vocabulary to express this information, as is done for instance with RDF schema or OWL2 `Declaration` axiom. We tend to prefer the latter approach, since this would allow to use RIF's `Import` to express our own `import` (in interfaces and logical parts of MWeb), and RIF's `Module` to capture our `uses` declaration.

The semantics of document formulas is captured by semantic multi-structures, i.e. a special set of RIF-FLD semantic structures. See [27, 7] for full details. Semantic structures are an extension of first-order logic semantics in order to be able to assign meaning to every kind of (HiLOG) terms and formulas in the language, including for instance datatypes and remote terms. These semantic structures are built from a set of truth-values $\mathbf{TV}$ (see the previous section), datatypes, the domain $\mathbf{D}$ (or universe) and several total mappings to interpret the different RIF-FLD formulas. In particular, we need for our discussion the total mapping $\mathbf{I}_C$ mapping constants in the countable infinite set of constant symbols `Const` to elements of the domain $\mathbf{D}$, and $\mathbf{I}_{truth}$ mapping elements of the domain $\mathbf{D}$ to truth values in $\mathbf{TV}$ for evaluating arbitrary formulas (see [27] for more details). A significant feature of the RIF-FLD semantics is that any formula is mapped first into an element of the domain, and afterwards the mapping $\mathbf{I}_{truth}$ is used to determine its truth-value.

Definition 5 (Semantic multi-structures [27]). *A semantic multi-structure,* $\hat{\mathbf{I}} = \{J, K; I^{i_1}, I^{i_2}, \ldots; M^{j_1}, M^{j_2}, \ldots\}$*, is a set of semantic structures such that*

- *J and K are the usual RIF-FLD semantic structures; and*

– I^{i_k} *and* M^{j_k}*, where* $k = 0, 1, 2, \ldots$*, are semantic structures adorned with locators of RIF-FLD document formulas (one can think of adorned structures as locator-structure pairs). The locators used in* $\hat{\mathbf{I}}$ *must be of the kinds allowed in the* `Import` *and* `Module` *directives.*

The semantic structures J*,* K*, and all the structures* I^{i_k} *in the import group are required to be identical in all respects except that the mappings* J_C*,* K_C*, and* $I_C^{i_k}$ *(for all* i_k*), which interpret constants in the semantic structures, may differ on those constants in* `Const` *that belong to the* `rif:local` *symbol space. The semantic structures* M^{j_k} *in the last group have many more degrees of freedom: they are required to agree with the other structures in* $\hat{\mathbf{I}}$ *only to the extent that the mappings* $M_C^{j_k}$ *must coincide with* J_C*,* K_C*, and* $I_C^{i_k}$ *on all constants in* `Const` *except the ones in the* `rif:local` *symbol space.*

As detailed in [27], the first semantic structure, J, is used to interpret non-document formulas (i.e. the group formula and its contents). The structure K is used for document formulas. The structures in the middle group, I^{i_k}, are optional; they are used to interpret imported documents (via the `Import` directive). All structures in that group must be adorned with the locators of distinct documents. The structures in the last group, M^{j_k}, are also optional; they are used to interpret documents that are linked as remote modules to other documents (via the `Module` directive). The structures in that group must also be adorned with locators of distinct documents. However, the same locator can adorn a structure in the import group and a structure in the module group.

Mark that `rif:local` constants are interpreted locally in each document, either in imported or module documents. The import structures must coincide except for these constants, while modules must interpret constants in the same way except for the `rif:local` constants. This simplifies a lot matters, but some approaches like Package-Description Logics refuse such a condition [6] having explicit mechanisms for mapping of vocabulary between different packages, in particular constants. We will not address this issue in this paper but can be identified as a potential problem of the RIF multi-document semantics.

Definition 6 (Imported document [27]). *Let* Δ *be a document formula and* `Import(loc)` *be one of its import directives, where* `loc` *is a locator of another document formula,* Δ'*. In this case, we say that* Δ' *is directly imported into* Δ*. A document formula* Δ' *is said to be imported into* Δ *if it is either directly imported into* Δ *or it is imported (directly or not) into another document, which itself is directly imported into* Δ*.*

Definition 7 (Remote module [27]). *Let* Δ *be a document formula and let* `Module(n loc)` *be one of its remote module directives*[5]*, where* `loc` *is a locator for another document formula,* Δ'*. In this case, we say that* Δ' *is a directly linked remote module of* Δ*. A document formula* Δ' *is said to be a linked remote module for* Δ *if it is either directly linked to* Δ *or it is linked (directly or not) to another document, which is directly linked to* Δ*.*

[5] Note that n is the term used to refer to the module.

Notice that the import relation and the linked module relation are made independent by the previous definitions, which is in our opinion a major source of problems. However, some relation between them is imposed by the next definition, which specifies how remote formulas are handled by linking to the interpretation of remote modules:

Definition 8 (Term-interpreting mapping for remote term refs [27]). *Let Δ be a document formula and $\hat{\mathbf{I}} = \{J, K; I^{i_1}, I^{i_2}, \ldots; M^{j_1}, M^{j_2}, \ldots\}$ be a semantic multi-structure that contains semantic structures for all the documents that are imported into Δ or linked to it as remote modules (directly or indirectly). Let $\Phi@r$ be a remote term that appears in Δ or one of its imported or linked documents, say Δ', and let $L \in \hat{\mathbf{I}}$ be a semantic structure. If there is a unique remote module directive* `Module(`$n\ \ j_k$`)` *in Δ' such that $L(r) = L(n)$ then $L(\Phi@r) = M^{j_k}(\Phi)$. If no such remote module directive exists or if such a directive is not unique, then $L(\Phi@r)$ is indeterminate, i.e., it can be any element in the domain of L. Truth valuation is extended as $TVal_L(\Phi@r) = I_{truth}(L(\Phi@r))$*[6].

In our reading, the previous definition has two problems. First, if there is a module directive inside an imported document, say i_l, then the corresponding semantic structure will not be added to $\hat{\mathbf{I}}$ since the import relation does not include them (modules are not followed by the import). But, then the module directive `Module(`$n\ \ j_k$`)` in Δ_{i_l} will be used to provide meaning to the remote term $\Phi@r$ via the (possibly) non-existing M^{j_k} semantic structure. An even more strange behaviour occurs for import directives inside modules, which are ignored by the previous definition. An example of these situations can be found in Fig. 7, which is based on the rulebases described in Section 2.

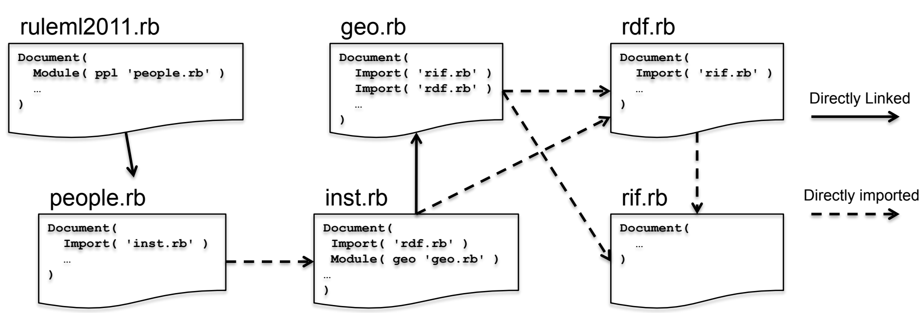


Fig. 7. Multi-document example

When trying to assign meaning to the document formula in 'ruleml2011.rb', the multi-document structure will only include semantic structures for documents in 'people.rb', while one would expect to have semantic structure for all

[6] I_{truth} is a total mapping $\mathbf{D} \rightarrow \mathbf{TV}$ used to define truth valuation for formulas

documens in the figure, and in particular the module in 'geo.rb'. Additionally, there is a problem lurking which is that imported documents in different modules are interpreted by the same semantic structure, which does not make sense at all. In the figure, one would expect a different semantic structure for interpreting the imported document 'rdf.rb' in rulebases 'inst.rb' and 'geo.rb'.

To complete the overview of the RIF-FLD semantics for document formulas we just need to introduce the truth-valuation of all formulas, document or not:

Definition 9 (Truth valuation in multi-document structures [27]). *Let Δ be a document formula and let $\Delta_1 \ldots \Delta_n$ be all the RIF-FLD document formulas that are imported (directly or indirectly, according to the previous definition) into Δ. Let $\Gamma, \Gamma_1, \ldots, \Gamma_n$ denote the respective group formulas associated with these documents. Let $\hat{\mathbf{I}} = \{J, K; I^{i_1}, I^{i_2}, \ldots; M^{j_1}, M^{j_2}, \ldots\}$ be a semantic multi-structure whose import group contains semantic structures adorned with the locators $i_1 \ldots i_n$ in the documents $\Delta_1 \ldots \Delta_n$. Then we define:*

$$TVal_{\hat{\mathbf{I}}}(\Delta) = glb_t(TVal_K(\Gamma), TVal_{I^{i_1}}(\Gamma_1), \ldots, TVal_{I^{i_n}}(\Gamma_n)).$$

For the non-document formulas Φ then $TVal_{\hat{\mathbf{I}}}(\Phi) = TVal_J(\Phi)$.

5 Alternative Semantics for Multi-documents

Our approach to the semantics of multi-documents is based on separating the interpretations of import from module directives. We take care first of import via importing multi-structures, which will be used to interpret documents as well as modules. A semantic importing multi-structure corresponds to RIF-FLD multi-structure without the optional module semantic structures.

Definition 10 (Semantic importing multi-structures). *A semantic importing multi-structure, $\tilde{\mathbf{I}} = \{J_{\mathbf{I}}, K_{\mathbf{I}}; \mathbf{I}^{i_1}, \mathbf{I}^{i_2}, \ldots\}$, is a set of semantic structures of the form where*

- *$J_{\mathbf{I}}$ and $K_{\mathbf{I}}$ are the usual RIF-FLD semantic structures; and*
- *$\mathbf{I}^{i_k}$ where $k = 0, 1, 2, \ldots$, are semantic structures adorned with locators of RIF-FLD document import formulas.*

The semantic structures $J_{\mathbf{I}}$, $K_{\mathbf{I}}$, and all the structures $\mathbf{I}^{i_k}$ in the import group are required to be identical in all respects except that the mappings $J_{\mathbf{I}_C}$, $K_{\mathbf{I}_C}$, and $\mathbf{I}_C^{i_k}$ (for all i_k), which interpret constants in the semantic structures, may differ on those constants in `Const` *that belong to the* `rif:local` *symbol space.*

Each module as well as the main document will have a corresponding semantic importing multi-structure in our new notion of modular semantic multi-structure.

Definition 11 (Modular semantic multi-structure). *A modular semantic multi-structure is a set $\overline{\mathbf{M}} = \{\tilde{\mathbf{M}}^{?j_0}, \tilde{\mathbf{M}}^{j_1}, \tilde{\mathbf{M}}^{j_2}, \ldots\}$ of semantic importing multi-structures such that*

- $\tilde{\mathbf{M}}^{?j_0}$ *is a semantic importing multi-structure, for providing the interpretation of the main document, which might be adorned with a locator* j_0*; and*
- $\tilde{\mathbf{M}}^{j_k}$ *where* $k = 0, 1, 2, \ldots$ *are semantic importing multi-structures adorned with locators of RIF-FLD document module formulas,*

Moreover, all the mappings of constants in the semantic structures composing $\tilde{\mathbf{M}}^{?j_0}, \tilde{\mathbf{M}}^{j_1}, \tilde{\mathbf{M}}^{j_2} \ldots$ *must coincide on all constants in* `Const` *except the ones in the* `rif:local` *symbol space.*

The rationale of modular semantic multi-structures is that each module may interpret differently their imported documents but must all coincide in the non-local constants. Remote term references will be interpreted in the corresponding semantic importing multi-structure.

Definition 12 (Term-interpreting mapping for remote term references). *Let* Δ *be a document formula and* $\overline{\mathbf{M}} = \{\tilde{\mathbf{M}}^{?j_0}, \tilde{\mathbf{M}}^{j_1}, \tilde{\mathbf{M}}^{j_2}, \ldots\}$ *be the modular semantic multi-structure with optional locator* j_0 *such that*

- $\tilde{\mathbf{M}}^{?j_0}$ *is a semantic importing multi-structure, containing an ordinary semantic structure for all the documents that are imported into* Δ *(directly, or indirectly);*
- $\tilde{\mathbf{M}}^{j_k}$ *is a semantic importing multi-structure, containing an ordinary semantic structure for all the documents that are imported into* Δ^{j_k} *(directly, or indirectly), where* Δ^{j_k} *is the document formula of a module located in* j_k*;*
- *if a directive* `Module(`$n\ j_m$`)` *occurs in any document formula of* Δ *or* Δ^{j_k} *or in any document imported or linked by them, then the corresponding semantic importing multi-structure* $\tilde{\mathbf{M}}^{j_m}$ *must occur in* $\overline{\mathbf{M}}$*. The locator* j_0 *of main document is mandatory if it occurs in some* `Module` *directive.*

Let $\Phi@r$ *be a remote term that appears in* Δ *(resp.* Δ^{j_k}*) or in one of its imported documents, say* Δ'*, and let* $L \in \tilde{\mathbf{M}}^{?j_0}$ *(resp.* $L \in \tilde{\mathbf{M}}^{j_k}$*) be an ordinary semantic structure. If there is a unique remote module directive* `Module(`$n\ j_m$`)` *in* Δ' *such that* $L(r) = L(n)$ *then* $L(\Phi@r) = J_{\tilde{\mathbf{M}}^{j_m}}(\Phi)$*. If no such remote module directive exists or if such a directive is not unique, then* $L(\Phi@r)$ *is indeterminate, i.e., it can be any element in the domain of* L*. Truth valuation is extended as* $TVal_L(\Phi@r) = I_{truth}(L(\Phi@r))$*.*

The conditions of Definition 12 guarantee that imported modules are treated separately in the several modules, and that each semantic importing multi-structure contains all the imported modules directly or indirectly. Furthermore, all modules occurring in a document formula or any imported or remote linked document have a corresponding semantic importing multi-structure.

Returning to the example of Fig. 7, we will have in the modular semantic multi-structure three semantic importing multi-structures. The one for the main document located in `'ruleml2011.org'` has no importing semantic structures. The importing multi-structure for document in `'geo.rb'` contains importing semantic structures for `'rdf.rb'` and `'rif.rb'`. The remaining importing semantic structure results from module in `'people.rb'`, and contains three importing structures for `'rdf.rb'`, `'rif.rb'`, and `'inst.rb'`. Note that it is not needed a semantic importing multi-structure for `'inst.rb'` since this document is never mentioned in a `Module` declaration. Finally, the semantics for multi-documents can be appropriately defined by extending the truth-valuation function for semantic importing and modular semantic multi-structures.

Definition 13 (Truth valuation in multi-document structures). *Let Δ be a document formula and let $\Delta_1 \dots \Delta_n$ be all the RIF-FLD document formulas that are imported (directly or indirectly, according to the previous definition) into Δ. Let $\Gamma, \Gamma_1, \dots, \Gamma_n$ denote the respective group formulas associated with these documents. Let $\tilde{\mathbf{I}} = \{J_{\mathbf{I}}, K_{\mathbf{I}}; \mathbf{I}^{i_1}, \mathbf{I}^{i_2}, \dots\}$ be a semantic importing multi-structure whose import group contains semantic structures adorned with the locators $i_1 \dots i_n$ in the documents $\Delta_1 \dots \Delta_n$. Then we define:*

$$TVal_{\tilde{\mathbf{I}}}(\Delta) = glb_t(TVal_{K_{\mathbf{I}}}(\Gamma), TVal_{\mathbf{I}^{i_1}}(\Gamma_1), \dots, TVal_{\mathbf{I}^{i_n}}(\Gamma_n)).$$

For the non-document formulas Φ then $TVal_{\tilde{\mathbf{I}}}(\Phi) = TVal_{J_{\mathbf{I}}}(\Phi)$.

Consider now a modular semantic multi-structure $\overline{\mathbf{M}} = \{\tilde{\mathbf{M}}^{?j_0}, \tilde{\mathbf{M}}^{j_1}, \tilde{\mathbf{M}}^{j_2}, \dots\}$ for any formula (document or not) φ then $TVal_{\overline{\mathbf{M}}}(\varphi) = TVal_{\tilde{\mathbf{M}}^{?j_0}}(\varphi)$.

Basically, the original truth-valuation for multi-document structures is adopted by semantic importing multi-structures, and the semantics of a document is provided by the semantic importing multi-structure of its document formula captured by $\tilde{\mathbf{M}}^{?j_0}$. Notice that the other semantic importing multi-structures in $\overline{\mathbf{M}}$ only affect the semantics of the original document via remote terms, which is apparently the intention under the RIF-FLD semantics for remote modules.

6 Models and Logical entailment

The notion of model by RIF-FLD follows a standard approach of classical logic and some fuzzy logics:

Definition 14 (Models [27]). *Let $\mathbf{I}$ be a semantic structure or multi-structure. We say that $\mathbf{I}$ is a model of a formula, Φ , written as $\mathbf{I} \models \Phi$ iff $TVal_{\mathbf{I}}(\Phi) = \mathbf{t}$. Here Φ can be a document or a non-document formula.*

The only comment that this definition deserves is that it restricts the usual notion of model in many-valued logics by limiting the distinguished truth-values to $\mathbf{t}$. However, for instance, the logic underlying paraconsistent well-founded semantics with explicit negation has three distinguished truth-values. We suggest generalizing the above definition by allowing a set of truth-values $\mathbf{DV}$ and substituting the condition $TVal_{\mathbf{I}}(\Phi) = \mathbf{t}$ by $TVal_{\mathbf{I}}(\Phi) \in \mathbf{DV}$ and additionally enforcing that $\mathbf{t} \in \mathbf{DV}$. Notice that this definition is an extension of Def. 14 and does not have any impact in already existing dialects. Finally, let us discuss logical entailment:

Definition 15 (Logical entailment [27]). *Let Φ and Ψ be (document or non-document) RIF-FLD formulas. We say that $\Phi \models \Psi$ iff for every intended semantic multi-structure $\hat{\mathbf{I}}$ it is the case that $TVal_{\hat{\mathbf{I}}}(\Phi) \leq_t TVal_{\hat{\mathbf{I}}}(\Psi)$.*

This is a natural definition of logical entailment for many-valued logics, and the relationship to the original definition of rule implication is obvious since $\Phi \models \Psi$ iff $\models \Phi \to \Psi$, i.e. it is assumed the deduction meta-theorem. However, the notion of logical entailment adopted for instance in Equilibrium Logic [22] resorts again to the notion of designated truth-values. Accordingly, an alternative more encompassing notion of logical entailment for the RIF-FLD framework would be that $\Phi \models \Psi$ iff $\models \Phi \to \Psi$ iff for every intended semantic multi-structure $\hat{\mathbf{I}}$ it is the case that $TVal_{\hat{\mathbf{I}}}(\Phi \to \Psi) \in \mathbf{DV}$, which captures the notion of logical entailment for all the logics presented in Section 3, and generalises Def. 15.

7 Conclusions and Further Work

This paper brings out some issues in the RIF-FLD semantics, for negation connectives, rule implication, models and logic entailment, as well as for multi-documents, all necessary to capture the Modular Web framework [4, 5]. It has been shown that the adoption of double negation law for the semantics of negation hinders the RIF intention of providing a sufficient general semantics capable of capturing in a natural way an interesting subset of existing semantics for rule-based systems, namely for extended logic programming under Answer Set Semantics [15] or Well-founded Semantics with Explicit Negation [3, 2]. It is argued that the only natural condition to impose is anti-monotonicity of the negation symbol. Problems are also found in the semantics of rule implication, and suggested the removal of one of the conditions. The notion of model is generalized and a proposal for new definition of logical entailment is also advanced. All these proposals are backwards compatible with any existing RIF-FLD dialects. Finally, the semantics of multi-documents in RIF-FLD has been detailed and problems brought out. These issues have been formally corrected according to what seems to be the spirit of [27]. In this way, we expect to succeed in the alignment of the syntax and semantics of our Modular Web framework.

References

1. J. Alcântara, C. V. Damásio, and L. M. Pereira. An encompassing framework for paraconsistent logic programs. *J. Applied Logic*, 3(1):67–95, 2005.
2. J. J. Alferes, C. V. Damásio, and L. M. Pereira. A Logic Programming System for Non-monotonic Reasoning. *Journal of Automated Reasoning*, 14(1):93–147, 1995.
3. J. J. Alferes and L. M. Pereira. On Logic Program Semantics with Two Kinds of Negation. In *Proc. of JICSLP-1992*, pages 574–588, 1992.
4. A. Analyti, G. Antoniou, and C. V. Damásio. A Principled Framework for Modular Web Rule Bases and Its Semantics. In *KR-2008*, pages 390–400. AAAI press, 2008.
5. A. Analyti, G. Antoniou, and C. V. Damásio. MWeb: A principled framework for modular web rule bases and its semantics. *ACM TOCL*, 12(2):#17, 41 pages, 2011.
6. J. Bao, G. Voutsadakis, G. Slutzki, and V. Honavar. Package-based description logics. In *Modular Ontologies*, LNCS 5445, pages 349–371. Springer, 2009.
7. H. Boley and M. Kifer. A guide to the basic logic dialect for rule interchange on the web. *IEEE Trans. Knowl. Data Eng.*, 22(11):1593–1608, 2010.

8. M. Bugliesi, E. Lamma, and P. Mello. Modularity in Logic Programming. *Journal of Logic Programming*, 19/20:443–502, 1994.
9. P. Cabalar, S. P. Odintsov, D. Pearce, and A. Valverde. Partial equilibrium logic. *Ann. Math. Artif. Intell.*, 50(3-4):305–331, 2007.
10. P. Cabalar, D. Pearce, P. Rondogiannis, and W. W. Wadge. A purely model-theoretic semantics for disjunctive logic programs with negation. In *Proc. of LP-NMR'07*, LNCS 4483, pages 44–57. Springer, 2007.
11. C. V. Damásio and L. M. Pereira. A model theory for paraconsistent logic programming. In C. Pinto-Ferreira and N. J. Mamede, editors, *Proc. of the 7th Portuguese Conf. on Artificial Intelligence (EPIA'95)*, LNAI 990, pages 377–386. Springer, Oct. 1995.
12. C. V. Damásio and L. M. Pereira. A Survey of Paraconsistent Semantics for Logic Programs. In D. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, pages 241–320. Kluwer Academic Publishers, 1998.
13. M. Dao-Tran, T. Eiter, M. Fink, and T. Krennwallner. Modular nonmonotonic logic programming revisited. In *Proc. of ICLP'09*, LNCS 5649, pages 145–159, 2009.
14. P. Doran, V. Tamma, and L. Iannone. Ontology module extraction for ontology reuse: an ontology engineering perspective. In *Proc. of ACM-CIKM '07*, pages 61–70. ACM, 2007.
15. M. Gelfond and V. Lifschitz. Logic programs with Classical Negation. In *7th International Conference on Logic Programming (ICLP'90)*, pages 579–597, 1990.
16. B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *In Proc. IJCAI-2007*, pages 298–304. AAAI, 2007.
17. M. Kifer. Flora-2: An object-oriented knowledge base language, 2007. Available at `http:/flora.sourceforge.net/`.
18. B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logics. In *Proc. of ECAI-2008*, pages 55–59. IOS Press, 2008.
19. E. Oikarinen and T. Janhunen. Modular Equivalence for Normal Logic Programs. In *Proc. of ECAI-2006*, pages 412–416, 2006.
20. OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009, Edited by W3C OWL Working Group.
21. P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, 10 February 2004.
22. D. Pearce. Equilibrium logic. *Annals of Math. and Artificial Intelligence*, 47(1-2):3–41, 2006.
23. RIF Basic Logic Dialect, 2010. W3C Recommendation 22 June 2010, Edited by Harold Boley and Michael Kifer.
24. RIF Core Answer Set Programming Dialect, 2009. RuleML specification 17 December 2009, Edited by Stijn Heymans and Michael Kifer.
25. RIF Core Logic Programming Dialect Based on the Well-founded Semantics, 2009. RuleML specification 13 August 2010, Edited by Michael Kifer.
26. RIF Core Logic Dialect, 2010. W3C Recommendation 22 June 2010, Edited by Harold Boley, Gary Hallmark, Michael Kifer, Adrian Paschke, Axel Polleres and Dave Reynolds.
27. RIF Framework for Logic Dialects, 2010. W3C Recommendation 22 June 2010, Edited by Harold Boley and Michael Kifer.
28. RIF Production Rule Dialect. W3C Recommendation 22 June 2010. Edited by Christian de Sainte Marie and Gary Hallmark and Adrian Paschke.