A Principled Framework for Modular Web Rule Bases and Its Semantics

Anastasia Analyti
Institute of Computer Science,
FORTH-ICS, Iraklion,
Crete, Greece
analyti@ics.forth.gr

Grigoris Antoniou
Institute of Computer Science,
FORTH-ICS, Greece
Department of Computer Science,
University of Crete, Greece
antoniou@ics.forth.gr

Carlos Viegas Damásio
CENTRIA, Departamento de Informatica,
Faculdade de Ciencias e Tecnologia,
Universidade Nova de Lisboa,
2829-516 Caparica, Portugal
cd@di.fct.unl.pt

Abstract

We present a principled framework for modular web rule bases, called `MWeb`. According to this framework, 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 valid `MWeb` modular rule bases $\mathcal{S}$, the `MWebAS` and `MWebWFS` semantics of each rule base $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ 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. Our framework supports: (i) local semantics and different points of view, (ii) local closed-world and open-world assumptions, (iii) scoped negation-as-failure, (iv) restricted propagation of local inconsistencies, and (v) monotonicity of reasoning, for "fully shared" predicates.

Keywords: modular web rule bases, local semantics, local closed-world and open-world assumptions, scoped negation-as-failure.

Introduction

The Semantic Web aims at defining formal languages and corresponding tools, enabling automated processing and reasoning over (meta-)data available from the Web. Logic and knowledge representation play a central role, but the distributed and world-wide nature of the Web brings new interesting research problems. In particular, the widely recognized need of having rules in the Semantic Web (see `http://www.ruleml.org`) has started the discussion on *local closed-world assumptions* (Heflin and Munoz-Avila 2002) and *scoped negation-as-failure* (otherwise, called *scoped default negation*) (RIF ; Kifer et al. 2005). Rule systems often provide for negation, founded on the closed-world assumption of complete information. In the Semantic Web, a rule like "if `book1` is not in stock then recommend it" has to be parametrized by the knowledge base (i.e., scope) that is used to search `book1` in the stock listings. Intuitively, the term scoped negation-as-failure indicates negation-as-failure, where the scope of the search failure is well-defined.

Weak (or default) negation is an appropriate rendering of the mechanism of negation-as-failure and is non-monotonic. Strong negation allows the user to express negative knowledge and is monotonic (Baral and Gelfond 1994). Moreover, the combination of weak and strong negation allows the distinction between open and closed predicates, as shown in (Analyti et al. 2008). However, the arbitrary and uncontrolled use of weak negation in the Semantic Web is regarded problematic and unsafe. The difficulty lies on the definition of simple mechanisms that can be easily explained to ordinary users and have nice mathematical properties.

The success of the Semantic Web is impossible without any form of modularity, encapsulation, information hiding, and access control. Currently, there is no notion of scope or context in the Semantic Web: all knowledge is global and all kinds of unexpected interactions can occur. In this paper, we propose a framework enabling collaborative reasoning over a set of web rule bases, while support for hidden knowledge is also provided. Our approach resembles the import/export mechanisms of Prolog, but we are mainly concerned with the safe use of strong and weak negation in the Semantic Web such that proposed mechanisms guarantee monotonicity for "fully shared" predicates.

In particular, we propose a framework, called *modular web logic framework* (`MWeb`), where users or applications import knowledge about predicates (available on rule bases over the web) into their own rule base. When a user or application imports a predicate p, he/she may express that certain nonmonotonic reasoning forms on p are not allowed. On the other hand, the producer of a predicate p in a rule base s can use nonmonotonic constructs on p, knowing that these constructs might be inhibited from an importing rule base. Additionally, he/she can express that a predicate p defined in rule base s is either (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.

We propose two semantics for `MWeb` modular rule bases, called `MWeb` *answer set semantics* (`MWebAS`) and `MWeb` *well-founded semantics* (`MWebWFS`). These semantics extend two major semantics for extended logic programs (ELPs), namely answer set semantics (`AS`) (Gelfond and Lifschitz 1990) and well-founded semantics with explicit negation (`WFSX`) (Pereira and Alferes 1992; Alferes, Damásio, and Pereira 1995) . We show that, similarly to the correspond-

DefinesDecl ::= "`defines`" *ScopeDecl DefinesPred* ["`visible to`" *RuleBaseList*] "."
UsesDecl ::= "`uses`" *UsesPred* ["`from`" *RuleBaseList*] "."

ScopeDecl ::= "`global`" | "`local`" | "`internal`"
RuleBaseList ::= *RuleBaseIRI* ("," *RuleBaseIRI*)*
DefinesPred ::= ("`definite`" | "`open`" | "`posClosed`" | "`negClosed`" | "`normal`") *PredicateInd* ["`wrt context`" *PredicateInd*]
UsesPred ::= ("`definite`" | "`open`" | "`closed`" | "`normal`") *PredicateInd*
PredicateInd ::= *AbsoluteIRI*
RuleBaseIRI ::= *AbsoluteIRI*

Figure 1: The syntax of the `defines` and `uses` declarations of an `MWeb` rule base

ing semantics for ELPs, `MWebAS` is more informative than `MWebWFS`. However, `MWebWFS` has better computational properties than `MWebAS`. Our framework leads to monotonic reasoning for `global` (that is, "fully shared") predicates, in the case that information and sharing of information in a modular rule base is increasing. Additionally, it supports local semantics and different points of view, local closed-world and open-world assumptions, scoped negation-as-failure, and restricted propagation of local inconsistencies.

The rest of the paper is organized as follows: In the next section, we provide a brief overview of our `MWeb` framework. Then, we formally define modular rule bases. The `MWebAS` and `MWebWFS` model-theoretic semantics of modular rule bases are defined in the subsequent section. Then, we provide several properties of `MWebAS` and `MWebWFS`. Subsequently, we discuss related work. Conclusions are provided in the concluding section.

Modularity for Rule Bases on the Web

In this section, we provide a brief overview of our `MWeb` framework. An *IRI* (*Internationalized Resource Identifier*) *reference* (Duerst and Suignard 2005) is a Unicode string that is used to provide globally unique names for web resources. For example, given that the namespace prefix `ex` stands for `http://www.example.org/`, the qualified name `ex:Riesling` (which stands for `http://www.example.org/Riesling`) is an IRI reference.

In our framework, *predicates names* are IRI references. Each (`MWeb`) *rule base* s is associated with a name Nam_s, which is also an IRI reference. A *constant* is an IRI reference or an RDF literal (Klyne and Carroll 2004). A *term* is a constant or a variable. In our presentation, variables are prefixed with a question mark symbol (`?`). Moreover, $\overline{t}$ denotes a sequence of terms, $\overline{x}$ denotes a sequence of variables, and $\overline{c}$ denotes a sequence of constants.

An *atom* is a *simple atom* $p(t_1, ..., t_k)$ or a *qualified atom* $p@Nam_t(t_1, ..., t_k)$, where p is a predicate of arity k, t_i, for $i = 1, ..., k$, are terms and Nam_t is the name of a rule base t. An *objective literal* is either an atom A or the strong negation $\neg A$ of an atom A. A *default literal* is the weak negation $\sim L$ of an objective literal L. An (`MWeb`) *rule* r is a formula of the form: $L \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots \sim L_n$, where L is a simple atom or the strong negation of a simple atom, and L_i (for $i \in \{1, ..., n\}$) is an objective literal. We say that r is *objective*, if no default literal appears in r. An (`MWeb`) *logic program* P is a set of rules. Note that if no qualified atom appears in P then P is an ELP. An (`MWeb`) *modular rule base* $\mathcal{S}$ is a set of rule bases.

Let $\mathcal{S}$ be a modular rule base. In addition to a name, each rule base s is associated with a logic program P_s. However, this information is not enough for determining the way knowledge, distributed over the various rule bases of $\mathcal{S}$, is integrated. Therefore, each rule base $s \in \mathcal{S}$ is also associated with an *interface* Int_s that contains two kinds of declarations: `defines` and `uses` (see Figure 1).

`Defines` declarations determine which predicates p are defined in s, as well as their *defining reasoning mode* and *scope* in s. The user can state the rule bases to which s is willing to export p, through the *visible to* clause.

The *defining reasoning mode* of a predicate takes the values `definite`, `open`, `positively closed`, `negatively closed`, or `normal`. In contrast to *normal* predicates, *definite*, *open*, and *closed* predicates impose restrictions on the use of weak negation in their defining rules. In particular, if a predicate p is declared `definite` in a rule base s then p should be defined (directly or indirectly) by objective rules, only. On the other hand, if a predicate p is declared `open` in s w.r.t. a predicate cxt then p is defined, not only through a set of objective rules, but also through the following rules: $openRules_s(p) = \{\neg p(\overline{x}) \leftarrow cxt(\overline{x}), \sim p(\overline{x}), \quad p(\overline{x}) \leftarrow cxt(\overline{x}), \sim \neg p(\overline{x})\}$. We refer to these rules, as the *contextual OWA rules* of p in s and to predicate cxt, as the *OWA context* of p in s. The contextual OWA rules of a predicate p in s provide a mechanism for making *local OWAs*. In particular, they express that if it exists $\overline{c}$ s.t. $cxt(\overline{c})$ is true in an intended model M of s then $p(\overline{c})$ or $\neg p(\overline{c})$ is true in M. If p is declared open in s without context information then p is called *freely open* in s. Additionally: $openRules_s(p) = \{\neg p(\overline{x}) \leftarrow \sim p(\overline{x}), \;\; p(\overline{x}) \leftarrow \sim \neg p(\overline{x})\}$.

Similarly, if a predicate p is declared `positively closed` or `negatively closed` in s w.r.t. a context cxt then p is defined, not only through a set of objective rules, but also through a *positive contextual CWA rule*: $posClosure_s(p) = \{\neg p(\overline{x}) \leftarrow cxt(\overline{x}), \sim p(\overline{x})\}$ or a *negative contextual CWA rule*: $negClosure_s(p) = \{p(\overline{x}) \leftarrow cxt(\overline{x}), \sim \neg p(\overline{x})\}$. We refer to predicate cxt as the *CWA context* of p in s. The contextual CWA rules of a predicate p in s provide a mechanism for making *local CWAs*. In particular, the positive closure rule of a predicate p in s expresses that, for any $\overline{c}$ s.t. $cxt(\overline{c})$

is true in an intended model M of s, if $p(\overline{c})$ is not true in M then $\neg p(\overline{c})$ is true in M. Similarly, the negative closure rule of a predicate p in s expresses that, for any $\overline{c}$ s.t. $cxt(\overline{c})$ is true in an intended model M of s, if $\neg p(\overline{c})$ is not true in M then $p(\overline{c})$ is true in M. If p is declared positively or negatively closed in s without context information then p is called *freely positively* or *freely negatively closed* in s, respectively. Additionally: $posClosure_s(p) = \{\neg p(\overline{x}) \leftarrow \sim p(\overline{x})\}$ and $negClosure_s(p) = \{p(\overline{x}) \leftarrow \sim \neg p(\overline{x})\}$, respectively.

Let p be a predicate, defined in a rule base $s \in \mathcal{S}$. If the *scope* of p is defined as `global` then predicate p is visible outside s and can be defined by any other rule base $s' \in \mathcal{S}$ in global scope, only. To guarantee monotonicity of reasoning for global predicates, the defining reasoning mode of a global predicate must be always definite or open. If the scope of p is defined as `local` then predicate p is visible outside s and can be also defined by another rule base $s' \in \mathcal{S}$ in internal scope, only. If the scope of a predicate p defined in a rule base s is `global` or `local` and the *visible to* clause of its corresponding `defines` declaration is missing then it is assumed that s is willing to export p to any requesting rule base in $\mathcal{S}$. Differently to global predicates, no constraint is imposed on the defining reasoning mode of local predicates. Finally, if the scope of p is defined as `internal` then predicate p is visible inside s, only. That is, no other rule base $s' \in \mathcal{S}$ can import p from s. Similarly to local predicates, no constraint is imposed on the defining reasoning mode of internal predicates.

`Uses` declarations determine which predicates p are requested by s and their *requesting reasoning mode* in s. The user can state the rule bases from which s requests p, through the *from* clause. The *requesting reasoning mode* of a used predicate p in a rule base s takes the values `definite`, `open`, `closed`, or `normal` and specifies the reasoning mode at which s is willing to import p. If the *from* clause of the `uses` declaration of a predicate p is missing then s imports p from any providing rule base in $\mathcal{S}$.

Assume now that a rule base $s \in \mathcal{S}$ defines a predicate p in a reasoning mode $\mathtt{m}$ and that another rule base $s' \in \mathcal{S}$ imports p from s in an requesting reasoning mode $\mathtt{m}'$, different that $\mathtt{m}$. Then, reasoning modes $\mathtt{m}$ and $\mathtt{m}'$ are combined, and the final reasoning mode that s' imports p from s equals $least(\mathtt{m}, \mathtt{m}')$, where $\mathtt{definite} \leq \mathtt{open} \leq \mathtt{closed} \leq \mathtt{normal}$. However, an error is caused, if the exporting rule base s defines p in normal reasoning mode, and the importing rule base s' declares that is willing to import p from s in definite, open, or closed reasoning mode. This is because, weak negation can freely appear in the definition of p in s. Therefore, the definition of p in s cannot be translated to a form that satisfies the constraints of the definite, open, or closed reasoning mode.

Example 1 Consider two rule bases s, $s' \in \mathcal{S}$ stating, respectively:

`defines local posClosed` p.
`uses open` p `from` Nam_s.

Thus, s defines a local predicate p as freely positively closed and s' states that is willing to accept p from s in open reasoning mode (i.e., the importing reasoning mode of p in s' is open). Then, rule base s' imports p from s, as if p had been declared in s in open reasoning mode. □

Example 2 Consider the `MWeb` modular rule base $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, shown in Figure 2[1]. Rule base s_1, with Nam_{s_1} =`<http://europa.eu>`, defines the list of European Union countries (which does not include `Croatia`), stating that this list is positively closed w.r.t. the CWA context `geo:Country`. Rule base s_2, with Nam_{s_2} =`<http://security.int>`, provides international citizenship information. Additionally, it lists persons suspect for crimes. Rule base s_3, with Nam_{s_3} =`<http://geography.int>`, provides geographical information, stating a positively closed list of countries.

Finally, rule base s_4, with Nam_{s_4} =`<http://gov.countryX>`, defines the immigration policies of an imaginary country `X`, which are supported by the knowledge of the other rule bases in $\mathcal{S}$. Note that even though `eu:CountryEU` is defined in s_1, in positively closed reasoning mode, rule base s_4 imports `eu:CountryEU` from s_1 in open reasoning mode. Furthermore, note that s_4 imports `sec:citizenOf` and `sec:Suspect` from s_2 in definite reasoning mode. Even through `sec:citizenOf` is defined in s_2 in local scope, `sec:citizenOf` is also defined internally in s_4 and additional facts about this predicate are stated. This is allowed since the internal information of `sec:citizenOf` in s_4 is not made public. Finally, note the presence of a default qualified literal in the rules of P_{s_4}.

Note that s_2 is providing confidential information to any requester. Safety can be improved if s_2 specifies the authorized consumers of `sec:citizenOf`, as in:

`defines global open sec:citizenOf visible to`
`<http://gov.countryX>`. □

Formalization of Modular Rule Bases

In this section, we formalize `MWeb` modular rule bases and define their validity. We denote the set of IRI references by $\mathcal{IRI}$ and the set of RDF literals by $\mathcal{LIT}$. Additionally, we denote the set of variable symbols by Var. The sets Var, $\mathcal{IRI}$, and $\mathcal{LIT}$ are pairwise disjoint.

An (`MWeb`) *vocabulary* V is a triple $\langle RBase, Pred, Const\rangle$, where $RBase \subseteq \mathcal{IRI}$ is a set of rule base names, $Pred \subseteq \mathcal{IRI}$ is a set of predicate names, and $Const \subseteq \mathcal{IRI} \cup \mathcal{LIT}$ is a set of constant symbols.

Each predicate name $p \in Pred$ is associated with an arity $arity(p) \in I\!N$. A *term* t over V is an element of $Const \cup Var$. Predicate names, rule base names, and terms are used for forming atoms and literals, as follows:

Let $V = \langle RBase, Pred, Const\rangle$ be a vocabulary. An (`MWeb`) *atom* over V is a *simple atom* $p(t_1, ..., t_n)$ or a *qualified atom* $p@rbase(t_1, ..., t_n)$, where $p \in Pred$, $rbase \in RBase$, $n = arity(p)$, and t_i is a term over V, for $i = 1, ..., n$.

Let $V = \langle RBase, Pred, Const\rangle$ be a vocabulary. An *objective literal* over V is an atom A or the strong negation $\neg A$

[1] To improve readability, namespace prefixes have been eliminated from the IRIs, representing constants.

Rule base s_1

```
⟨http://europa.eu⟩
defines local posClosed eu:CountryEU
        wrt context geo:Country.
uses definite geo:Country from
        ⟨http://geography.int⟩.
eu:CountryEU(Austria).
eu:CountryEU(Greece).  ···
```

Rule base s_2

```
⟨http://security.int⟩
defines local open sec:citizenOf.
defines glocal open sec:Suspect.
sec:citizenOf(Anne,Austria).
sec:citizenOf(Boris,Croatia).
sec:Suspect(Peter).
```

Rule base s_3

```
⟨http://geography.int⟩
defines local posClosed geo:Country.
geo:Country(Egypt).  geo:Country(Canada).  ···
```

Rule base s_4

```
⟨http://gov.countryX⟩
defines local normal gov:Enter visible to ⟨http://security.int⟩.
defines local negClosed gov:RequiresVisa wrt context geo:Country.
defines internal open sec:citizenOf.
uses definite geo:Country from ⟨http://geography.int⟩.
uses open eu:CountryEU from ⟨http://europa.eu⟩.
uses definite sec:citizenOf from ⟨http://security.int⟩.
uses definite sec:Suspect from ⟨http://security.int⟩.
  gov:Enter(?p) ←  eu:CountryEU(?c), sec:citizenOf(?p,?c),
                   ∼sec:Suspect@⟨http://security.int⟩(?p).
  gov:Enter(?p) ← ¬eu:CountryEU(?c), sec:citizenOf(?p,?c),
                   ¬gov:RequiresVisa(?c),
                   ∼sec:Suspect@⟨http://security.int⟩(?p).
¬gov:RequiresVisa(Croatia).     sec:citizenOf(Peter,Greece).
```

Figure 2: An MWeb modular rule base

of an atom A over V. A *default literal* over V is the weak negation $\sim L$ of an objective literal L over V. An (MWeb) *literal* over V is an objective or a default literal over V.

We denote the set of objective literals over V and the set of literals over V by $Lit^{\circ}(V)$ and $Lit(V)$, respectively. Let $L \in Lit(V)$. We define $pred(L) = p$, where p is the predicate name appearing in L. If L is built from a qualified atom $p@rbase(\overline{t})$, we define the *qualifying rule base* of L as $qual(L) = rbase$. Otherwise, $qual(L)$ is undefined.

Let L be a qualified literal, we denote by $simple(L)$, the literal L without $qual(L)$, e.g. $simple($`sec:Suspect@`$\langle http://security.int\rangle$`(?p)`$)$= `sec:Suspect(?p)`.

Let $L \in Lit^{\circ}(V)$, we define $\neg(\neg L) = L$ and $\sim(\sim L) = L$. Additionally, let $S \subseteq Lit^{\circ}(V)$. We define $\neg S = \{\neg L \mid L \in S\}$ and $\sim S = \{\sim L \mid L \in S\}$.

Based on literals, we define rules and logic programs, as follows:

Definition 1 (Logic program) Let $V = \{RBase, Pred, Const\}$ be a vocabulary. An (MWeb) *rule* r over V is an expression $L_0 \leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n$, where: (i) $L_0 \in Lit^{\circ}(V) \cup \{\mathtt{f}\}$ and $qual(L_0)$ is undefined, (ii) $L_i \in Lit^{\circ}(V) \cup \{\mathtt{t}, \mathtt{u}\}$, for $i = 1, ..., m$, and (iii) $L_i \in Lit^{\circ}(V)$, for $i = m+1, ..., n$. We define $Head_r = L_0$, $Body_r^+ = \{L_1, ..., L_m\}$, $Body_r^- = \{L_{m+1}, ..., L_n\}$, and $Body_r = Body_r^+ \cup \sim Body_r^-$. An (MWeb) *logic program* over V is a set of rules over V. □

The symbols `t`, `u`, and `f` are called *special literals* and represent the truth values `true`, `undefined`, and `false`, respectively. A rule r over a vocabulary V with $Head_r = \mathtt{f}$ is called (MWeb) *constraint* over V.

As we have seen in the previous section, each rule base s is associated with a name Nam_s, a logic program P_s, and an interface Int_s that includes `defines` and `uses` declarations.

Definition 2 (Rule base) Let $V = \langle RBase, Pred, Const\rangle$ be a vocabulary. An (MWeb) *rule base* s over V is a triple $s = \langle Nam_s, P_s, Int_s\rangle$, where: (i) $Nam_s \in RBase$ is the *name* of s, (ii) P_s is a logic program over V, called the *logic program* of s, and (iii) $Int_s = \langle Def_s, Uses_s\rangle$ is the *interface* of s, where:

- Def_s is a set of tuples $\langle p, sc, mod, cxt, Exp\rangle$, where $p \in Pred$, $sc \in \{\mathtt{gl}, \mathtt{lc}, \mathtt{int}\}$, $mod \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}^+, \mathtt{c}^-, \mathtt{n}\}$, $cxt \in Pred \cup \{\mathtt{n/a}\}$, and $Exp \subseteq RBase - \{Nam_s\}$ or $Exp = \{*\}$.
  We define $Pred_s^{\mathtt{D}} = \{p \mid \exists \langle p, sc, mod, cxt, Exp\rangle \in Def_s\}$, $scope_s(p) = sc$, $mode_s^{\mathtt{D}}(p) = mod$, $context_s(p) = cxt$, and $Export_s(p) = Exp$.
- $Uses_s$ is a set of tuples $\langle p, mod, Imp\rangle$, where $p \in Pred$, $mod \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, and $Imp \subseteq RBase - \{Nam_s\}$ or $Imp = \{*\}$.

We define $Pred^{\mathtt{U}}_s = \{p \mid \exists \langle p, mod, Imp\rangle \in Uses_s\}$, $mode^{\mathtt{U}}_s(p) = mod$, and $Import_s(p) = Imp$. □

Let s be a rule base. We define: $Pred_s = Pred^{\mathtt{D}}_s \cup Pred^{\mathtt{U}}_s$. Intuitively, each tuple $\langle p, sc, mod, cxt, Exp\rangle \in Def_s$ corresponds to a `defines` declaration of s, where p is a predicate defined in s, sc is the scope of p in s (i.e., `global`, `local`, or `internal`), mod is the defining reasoning mode of p in s (i.e., `definite`, `open`, `positively closed`, `negatively closed`, or `normal`), cxt is the context of p in s (if defined), and Exp is the list of rules bases to which s is willing to export p. If the *wrt context* clause of the `defines` declaration is missing then $cxt = \mathtt{n/a}$. Additionally, if $sc = \mathtt{int}$ and the *visible to* clause of the `defines` declaration is missing then $Exp = \{\}$. However, if $sc \in \{\mathtt{gl}, \mathtt{lc}\}$ and the *visible to* clause of the `defines` declaration is missing then $Exp = *$. This means that s is willing to export p to any requesting rule base. We say that p is *freely open* (resp. *freely closed*) in s if $mode = \mathtt{o}$ (resp. $mode \in \{\mathtt{c}^+, \mathtt{c}^-\}$) and $cxt = \mathtt{n/a}$.

Similarly, each tuple $\langle p, mod, Imp\rangle \in Uses_s$ corresponds to a `uses` declaration of s, where p is a predicate requested by s, mod is the requesting reasoning mode of p in s (i.e., `definite`, `open`, `closed`, or `normal`), and Imp is the list of rules bases from which p is requested. If the *from* clause of the `uses` declaration is missing then $Imp = *$. In this case, s imports p from any providing rule base.

Example 3 Consider rule base s_1 of Example 2. Then, Def_{s_1} = {⟨ `eu:CountryEU`, `lc`, $\mathtt{c}^+$, `geo:Country`, $*$⟩} and $Uses_{s_1}$ = {⟨ `geo:Country`, `d`, {<http://geography.int>}⟩}. □

We define[2]: $|\mathtt{d}| = \mathtt{d}$, $|\mathtt{o}| = \mathtt{o}$, $|\mathtt{c}^+| = |\mathtt{c}^-| = \mathtt{c}$, and $|\mathtt{n}| = \mathtt{n}$. Then, we impose the following total order: $\mathtt{d} \le \mathtt{o} \le \mathtt{c} \le \mathtt{n}$, called *reasoning mode extension*. Additionally, we impose the following total order on predicate scopes: $\mathtt{int} \le \mathtt{lc} \le \mathtt{gl}$, called *predicate scope extension*.

Definition 3 (Valid rule base) A rule base $s = \langle Nam_s, P_s, Int_s\rangle$ over a vocabulary $V = \{RBase, Pred, Const\}$ is *valid* iff:

1. If $\langle p, sc, mod, cxt, Exp\rangle, \langle p, sc', mod', cxt', Exp'\rangle \in Def_s$ then $sc = sc'$, $mod = mod'$, $cxt = cxt'$, and $Exp = Exp'$.
2. If $\langle p, mod, Imp\rangle, \langle p, mod', Imp'\rangle \in Uses_s$ then $mod = mod'$ and $Imp = Imp'$.
3. For all $r \in P_s$:
   (a) $pred(Head_r) \in Pred^{\mathtt{D}}_s$.
   (b) for all $L \in Body_r$, $pred(L) \in Pred^{\mathtt{D}}_s \cup Pred^{\mathtt{U}}_s$.
4. For all $\langle p, sc, mod, cxt, Exp\rangle \in Def_s$:
   (a) if $mod \in \{\mathtt{o}, \mathtt{c}^+, \mathtt{c}^-\}$ and $cxt \in Pred$ then $cxt \in Pred_s$ and $arity(cxt) = arity(p)$,
   (b) if $cxt \in Pred^{\mathtt{D}}_s$ then $mode^{\mathtt{D}}_s(cxt) \in \{\mathtt{d}\}$,
   (c) if $cxt \in Pred^{\mathtt{U}}_s$ then $mode^{\mathtt{U}}_s(cxt) \in \{\mathtt{d}\}$,
   (d) if $mod \in \{\mathtt{d}, \mathtt{n}\}$ then $cxt = \mathtt{n/a}$.
5. If $p \in Pred^{\mathtt{D}}_s$ and $scope_s(p) = \mathtt{gl}$ then $mode^{\mathtt{D}}_s(p) \in \{\mathtt{d}, \mathtt{o}\}$.
6. If $p \in Pred^{\mathtt{D}}_s$ and $scope_s(p) = \mathtt{int}$ then $Export_s(p) = \{\}$.
7. If $p \in Pred^{\mathtt{D}}_s \cap Pred^{\mathtt{U}}_s$ then $mode^{\mathtt{U}}_s(p) \le |mode^{\mathtt{D}}_s(p)|$.
8. For all $r \in P_s$, and for all $L \in Body_r$:
   if $qual(L) \in RBase$ then $qual(L) \in Import_s(pred(L))$ or $Import_s(pred(L)) = \{*\}$.
9. For all $r \in P_s$, and for all $L \in Body_r$:
   if $mode^{\mathtt{D}}_s(pred(Head_r)) \neq \mathtt{n}$ then:
   (a) $Body^-_r = \{\}$,
   (b) for all $L \in Body^+_r$, if $pred(L) \in Pred^{\mathtt{D}}_s$ then $mode^{\mathtt{D}}_s(pred(L)) \neq \mathtt{n}$, and
   (c) for all $L \in Body^+_r$, if $pred(L) \in Pred^{\mathtt{U}}_s$ then $mode^{\mathtt{U}}_s(pred(L)) \neq \mathtt{n}$. □

Let s be a valid rule base. Constraint 1 of Definition 3 expresses that for each defined predicate, there should be only one `defines` declaration in s. Constraint 2 expresses that for each requested predicate, there should be only one `uses` declaration in s. Constraint 3 expresses that for each predicate appearing in the head of a rule $r \in P_s$, there should be a corresponding `defines` declaration. Additionally, for each predicate appearing in the body of r, there should be a corresponding `defines` or `uses` declaration.

Constraint 4 expresses that each open or closed predicate p, defined in s, can be associated with a predicate cxt. This predicate cxt should be defined in s or requested by s and have the same arity as p. If cxt is defined in (resp. requested by) s then its defining (resp. requesting) reasoning mode should be definite.

Constraint 5 expresses that each global predicate of s should be defined in definite or open reasoning mode. This is because reasoning on global predicates should be monotonic. Constraint 6 expresses that each internal predicate of s is not visible by other rule bases. Constraint 7 expresses that if a predicate p is both defined in s and requested by s then its defining reasoning mode in s should extend its requesting reasoning mode in s. Intuitively, this means that the use of weak negation in the imported definition of p should satisfy the constraints of the defining reasoning mode of p in s.

Constraint 8 expresses that if a qualified literal L appears in the body of a rule $r \in P_s$ then rule base s should request $pred(L)$ from rule base $qual(L)$. Constraint 9 expresses that for each rule $r \in P_s$, if the defining reasoning mode of the predicate appearing in $Head_r$ is restricted (i.e., not `normal`) then: (i) no default literal should appear in $Body_r$, and (ii) the defining (resp. requesting) reasoning mode of each defined (resp. requested) predicate appearing in $Body_r$ should also be restricted.

Example 4 Rule bases s_1, s_2, s_3, and s_4 of Example 2 are valid.

Definition 4 (Modular rule base) An (`MWeb`) *modular rule base* $\mathcal{S}$ over a vocabulary V is a set of valid rule bases over V. □

[2] This auxiliary definition is needed, because the defining reasoning modes of a predicate p are $\{\mathtt{d}, \mathtt{o}, \mathtt{c}^+, \mathtt{c}^-, \mathtt{n}\}$, where as the requesting reasoning modes of a predicate p, and the reasoning modes of an interpretation of a rule base s (to be defined later) are $\{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$.

Let $\mathcal{S}$ be a modular rule base, $s \in \mathcal{S}$, and $p \in \mathit{Pred}^{\mathtt{D}}_s$. We define:

$$\mathit{Export}^{\mathcal{S}}_s(p) = \begin{cases} \{\mathit{Nam}_{s'} \mid s' \in \mathcal{S} - \{s\}\} \text{ if } \mathit{Export}_s(p) = \{*\} \\ \mathit{Export}_s(p) \cap \{\mathit{Nam}_{s'} \mid s' \in \mathcal{S}\} \quad \text{otherwise} \end{cases}$$

Intuitively, $\mathit{Export}^{\mathcal{S}}_s(p)$ denotes the rule bases in $\mathcal{S}$ that s is willing to export p. We refer to $\mathit{Export}^{\mathcal{S}}_s(p)$ as the ***exporting rule base list*** of p in s w.r.t. $\mathcal{S}$.

Let $\mathcal{S}$ be a modular rule base, $s \in \mathcal{S}$, and $p \in \mathit{Pred}^{\mathtt{U}}_s$. We define:

$$\mathit{Import}^{\mathcal{S}}_s(p) = \begin{cases} \mathit{ExportingTo}_{\mathcal{S}}(p, s) \quad \text{if } \mathit{Import}_s(p) = \{*\} \\ \mathit{Import}_s(p) \cap \mathit{ExportingTo}_{\mathcal{S}}(p, s) \text{ otherwise} \end{cases}$$

where $\mathit{ExportingTo}_{\mathcal{S}}(p, s) = \{\mathit{Nam}_{s'} \mid s' \in \mathcal{S},\ \mathit{Nam}_s \in \mathit{Export}^{\mathcal{S}}_{s'}(p)\}$.

Intuitively, $\mathit{ExportingTo}_{\mathcal{S}}(p, s)$ denotes the rule bases in $\mathcal{S}$ that are wiling to export p to s. Note that for the modular rule base $\mathcal{S}$ of Example 2, $\mathit{ExportingTo}_{\mathcal{S}}$(`sec:citizenOf`, s_4) $= \{s_2\}$. Additionally, $\mathit{Import}^{\mathcal{S}}_s(p)$ denotes the rule bases in $\mathcal{S}$ from which s imports p. We refer to $\mathit{Import}^{\mathcal{S}}_s(p)$ as the *importing rule base list* of p in s w.r.t. $\mathcal{S}$. Note that: for all $p \in \mathit{Pred}_s$, $\mathit{Nam}_s \notin \mathit{Export}^{\mathcal{S}}_s(p) \cup \mathit{Import}^{\mathcal{S}}_s(p)$.

Example 5 Consider the modular rule base $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ of Example 2. It holds, $\mathit{Export}^{\mathcal{S}}_{s_2}$(`sec:citizenOf`) $= \{s_4\}$, while Export_{s_2}(`sec:citizenOf`) $= \{*\}$. Additionally, $\mathit{Export}^{\mathcal{S}}_{s_4}$(`gov:Enter`) $= \{\}$. Note that even though s_4 is willing to export `gov:Enter` to s_2, rule base s_2 does not use `gov:Enter`. Additionally, it holds $\mathit{Import}^{\mathcal{S}}_{s_4}$(`sec:citizenOf`) $= \{s_2\}$. □

Definition 5 (Valid modular rule base) A modular rule base $\mathcal{S}$ is *valid* iff:

1. If $s \in \mathcal{S}$ then s is a valid rule base.
2. If $s, s' \in \mathcal{S}$ and $s \neq s'$ then $\mathit{Nam}_s \neq \mathit{Nam}_{s'}$.
3. For all $s, s' \in \mathcal{S}$ s.t. $s \neq s'$, and for all $p \in \mathit{Pred}^{\mathtt{D}}_s$: if $\mathit{scope}_s(p) = \mathtt{int}$ then $p \notin \mathit{Pred}^{\mathtt{U}}_{s'}$.
4. For all $s, s' \in \mathcal{S}$ s.t. $s \neq s'$, and for all $p \in \mathit{Pred}^{\mathtt{D}}_s \cap \mathit{Pred}^{\mathtt{D}}_{s'}$ s.t. $\mathit{Nam}_s \in \mathit{Import}^{\mathcal{S}}_{s'}(p)$: if $\mathit{mode}^{\mathtt{D}}_s(p) = \mathtt{n}$ then $\mathit{mode}^{\mathtt{U}}_{s'}(p) = \mathtt{n}$.
5. For all $s, s' \in \mathcal{S}$ s.t. $s \neq s'$, and for all $p \in \mathit{Pred}^{\mathtt{D}}_s \cap \mathit{Pred}^{\mathtt{D}}_{s'}$: if $\mathit{scope}_s(p) = \mathtt{lc}$ then $\mathit{scope}_{s'}(p) = \mathtt{int}$.
6. For all $s, s' \in \mathcal{S}$ s.t. $s \neq s'$, and for all $p \in \mathit{Pred}^{\mathtt{D}}_s \cap \mathit{Pred}^{\mathtt{D}}_{s'}$: if $\mathit{scope}_s(p) = \mathtt{gl}$ then $\mathit{scope}_{s'}(p) = \mathtt{int}$ or $\mathit{scope}_{s'}(p) = \mathtt{gl}$.
7. If $s \in \mathcal{S}$ and $p \in \mathit{Pred}^{\mathtt{U}}_s$ then $\mathit{Import}_s(p) = \{*\}$ or $\mathit{Import}_s(p) \subseteq \mathit{ExportingTo}_{\mathcal{S}}(p, s)$. □

Let $\mathcal{S}$ be a valid modular rule base. Constraint 1 of Definition 5 expresses that each rule base in $\mathcal{S}$ should be a valid rule base. Constraint 2 expresses that distinct rule bases in $\mathcal{S}$ should have distinct names. Constraint 3 expresses that if a rule base $s \in \mathcal{S}$ defines internally a predicate p then another rule base $s' \in \mathcal{S}$ cannot request p from s. Constraint 4 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `normal` reasoning mode and requested by another rule base $s' \in \mathcal{S}$ from s then its requesting reasoning mode in s' should also be `normal`. This is because, the use of weak negation in the definition of p in s is unrestricted.

Constraint 5 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `local` scope then it can be defined by another rule base $s' \in \mathcal{S}$ only in `internal` scope. This is because internal predicates are invisible to other rule bases. Constraint 6 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `global` scope then it can be defined by another rule base $s' \in \mathcal{S}$ only in `global` or `internal` scope. Constraint 7 expresses that if a rule base $s \in \mathcal{S}$ requests a predicate p from a *specific* rule base s' then s' should be a rule base of $\mathcal{S}$ that defines p and is willing to export p to s. That is, $\mathit{Import}_s(p) = \{*\}$ or $\mathit{Import}_s(p) = \mathit{Import}^{\mathcal{S}}_s(p)$.

Example 6 Modular rule base $\mathcal{S}$ of Example 2 is valid. □

Convention: *In this work, we consider valid modular rule bases, only.*

Model-theoretic Semantics for Modular Rule Bases

In this section, we propose the `MWeb` *answer set semantics* (`MWebAS`) and the `MWeb` *well-founded semantics* (`MWebWFS`) for modular rule bases. We will show that these semantics extend the answer set semantics (`AS`) and the well-founded semantics with explicit negation (`WFSX`) on ELPs, respectively. First, we define the normal and extended interpretations of a modular rule base.

Let $\mathcal{S} = \{s_1, ..., s_n\}$ be a modular rule base. The *Herbrand universe* of $\mathcal{S}$ is defined as: $\mathtt{HU}_{\mathcal{S}} = \mathtt{HU}_{s_1} \cup ... \cup \mathtt{HU}_{s_n}$, where $\mathtt{HU}_{s_i}$ (for $i = 1, ..., n$) is the set of constants appearing in P_{s_i}.

Let $\mathcal{S}$ be a modular rule base over a vocabulary $V = \{\mathit{RBase}, \mathit{Pred}, \mathit{Const}\}$. Let $p \in \mathit{Pred}$, $n = \mathit{arity}(p)$, and $\mathit{rbase} \in \mathit{RBase}$. We denote by $[p]_{\mathcal{S}}$ the set of literals $p(t_1, ..., t_n)$ and $\neg p(t_1, ..., t_n)$, where $t_i \in \mathtt{HU}_{\mathcal{S}}$, for $i = 1, ..., n$. Additionally, we denote by $[p@\mathit{rbase}]_{\mathcal{S}}$ the set of literals $p@\mathit{rbase}(t_1, ..., t_n)$ and $\neg p@\mathit{rbase}(t_1, ..., t_n)$, where $t_i \in \mathtt{HU}_{\mathcal{S}}$, for $i = 1, ..., n$.

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. The *Herbrand base* of s w.r.t. $\mathcal{S}$ is defined as: $\mathtt{HB}^{\mathcal{S}}_s = \{[p]_{\mathcal{S}} \mid p \in \mathit{Pred}_s\} \cup \{[p@\mathit{rbase}]_{\mathcal{S}} \mid p \in \mathit{Pred}^{\mathtt{U}}_s, \mathit{rbase} \in \mathit{Import}^{\mathcal{S}}_s(p)\}$.

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. A *simple normal interpretation* of s w.r.t. $\mathcal{S}$ is a set $I \subseteq \mathtt{HB}^{\mathcal{S}}_s$ s.t. $I \cap \neg I = \emptyset$ (*consistency*) or $I = \mathtt{HB}^{\mathcal{S}}_s$. If $I = \mathtt{HB}^{\mathcal{S}}_s$ then I is called *inconsistent*. Otherwise, I is called *consistent*. As usual, I can be seen, equivalently, as a function from $\mathtt{HB}^{\mathcal{S}}_s \rightarrow \{0, 1\}$, where: (i) $I(L) = 1$, if $L \in I$, and (ii) $I(L) = 0$, if $L \notin I$. Let $L \in \mathtt{HB}^{\mathcal{S}}_s$. We define: (i) $I(\sim L) = 1 - I(L)$, if I is consistent, and (ii) $I(\sim L) = 1$, otherwise. I also assigns values to special literals $\mathtt{t}$ and $\mathtt{f}$. In particular, we define $I(\mathtt{t}) = 1$. Additionally, we define: (i) $I(\mathtt{f}) = 0$, if I is consistent, and (ii) $I(\mathtt{f}) = 1$, otherwise.

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. A *simple extended interpretation* of s w.r.t. $\mathcal{S}$ is a set $I = T \cup {\sim}F_{\mathtt{d}}$, where $T, F_{\mathtt{d}} \subseteq \mathtt{HB}_s^{\mathcal{S}}$ s.t. either: (i) $\neg T \subseteq F_{\mathtt{d}}$ (*coherency*) and $T \cap F_d = \emptyset$ (*consistency*), or (ii) $T = F_{\mathtt{d}} = \mathtt{HB}_s^{\mathcal{S}}$. If $I = \mathtt{HB}_s^{\mathcal{S}} \cup {\sim}\mathtt{HB}_s^{\mathcal{S}}$ then I is called *inconsistent*. Otherwise, I is called *consistent*. As usual, I can be seen, equivalently, as a function from $\mathtt{HB}_s^{\mathcal{S}} \cup {\sim}\mathtt{HB}_s^{\mathcal{S}} \rightarrow \{0, 1/2, 1\}$, where: (i) $I(L) = 1$, if $L \in I$, (ii) $I(L) = 0$, if $L \notin I$ and ${\sim}L \in I$, and (iii) $I(L) = 1/2$, if $L \notin I$ and ${\sim}L \notin I$. I also assigns values to special literals $\mathtt{t}$, $\mathtt{u}$, and $\mathtt{f}$. In particular, we define $I(\mathtt{t}) = 1$ and $I(\mathtt{u}) = 1/2$. Additionally, we define: (i) $I(\mathtt{f}) = 0$, if I is consistent, and (ii) $I(\mathtt{f}) = 1$, otherwise.

Let I be a simple normal or extended interpretation of s w.r.t. $\mathcal{S}$ and let $L \in \mathtt{HB}_s^{\mathcal{S}}$. It easy to see that: for all $L \in \mathtt{HB}_s^{\mathcal{S}}$, $I(\neg L) = 1$ implies $I({\sim}L) = 1$ (*coherency*), but not vice-versa. Additionally, let $S \subseteq \mathtt{HB}_s^{\mathcal{S}} \cup {\sim}\mathtt{HB}_s^{\mathcal{S}} \cup \{\mathtt{t}, \mathtt{u}, \mathtt{f}\}$. We define: $I(S) = min\{I(L) | \, L \in S\}$.

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. A *normal* (resp. *extended*) *interpretation* of s w.r.t. $\mathcal{S}$ is a tuple $I_s = \langle I_s^{\mathtt{d}}, I_s^{\mathtt{o}}, I_s^{\mathtt{c}}, I_s^{\mathtt{n}} \rangle$, where $I_s^{\mathtt{x}}$ is a simple normal (resp. extended) interpretation of s w.r.t. $\mathcal{S}$ (for $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$). Intuitively, $I_s^{\mathtt{d}}, I_s^{\mathtt{o}}, I_s^{\mathtt{c}}$, and $I_s^{\mathtt{n}}$ correspond to the `definite`, `open`, `closed`, and `normal` reasoning modes of I_s, respectively. A *normal* (resp. *extended*) *interpretation* of $\mathcal{S}$ is a set $\mathsf{I} = \{I_s \mid s \in \mathcal{S}\}$, where I_s is a normal (resp. extended) interpretation of s w.r.t. $\mathcal{S}$.

Let $\mathcal{S}$ be a modular rule base. Let $\mathsf{I} = \{I_s \mid s \in \mathcal{S}\}$ and $\mathsf{J} = \{J_s \mid s \in \mathcal{S}\}$ be normal (resp. extended) interpretations of $\mathcal{S}$. We say that: J *extends* I w.r.t. *truth* ($\mathsf{I} \leq_{\mathtt{t}} \mathsf{J}$) iff for all $s \in \mathcal{S}$, for all $L \in \mathtt{HB}_s^{\mathcal{S}}$, and for all $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, $I_s^{\mathtt{x}}(L) \leq J_s^{\mathtt{x}}(L)$.

Before, we define the *normal* and *extended answer sets* of a modular rule base, a few auxiliary definitions are provided. Let $\mathcal{S}$ be a modular rule base and let P be a logic program. We will denote by $[P]_{\mathcal{S}}$ the set of rules in P instantiated over $\mathtt{HU}_{\mathcal{S}}$. Below, we adapt the definition of an ELP model in `WFSX` (Pereira and Alferes 1992) to our framework.

Definition 6 (Logic program satisfaction) Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let I be a simple normal (resp. extended) interpretation of s w.r.t. $\mathcal{S}$. We say that I *satisfies* a logic program P ($I \models P$) iff for all $r \in [P]_{\mathcal{S}}$, (i) $I(Head_r) \geq I(Body_r)$, or (ii) $I(\neg Head_r) = 1$ and $I(Body_r) = 1/2$. □

Let $\mathcal{S}$ be a modular rule base. For each $s \in \mathcal{S}$, we define[3] four logic programs that correspond to the four reasoning modes of s, that is `definite`, `open`, `closed`, and `normal`.

$$P_s^{\mathtt{d}} = \{r \in P_s \mid mode_s^{\mathtt{D}}(pred(Head_r)) \neq \mathtt{n}\}.$$

$$P_s^{\mathtt{o}} = \{r \in P_s \mid \, |mode_s^{\mathtt{D}}(pred(Head_r))| \in \{\mathtt{o}, \mathtt{c}\}\} \cup \{openRules_s(p) \mid \, |mode_s^{\mathtt{D}}(p)| \in \{\mathtt{o}, \mathtt{c}\}\}.$$

$$P_s^{\mathtt{c}} = \{r \in P_s \mid \, |mode_s^{\mathtt{D}}(pred(Head_r))| = \mathtt{c}\} \cup \{posClosure_s(p) \mid mode_s^{\mathtt{D}}(p) = \mathtt{c}^+\} \cup \{negClosure_s(p) \mid mode_s^{\mathtt{D}}(p) = \mathtt{c}^-\}.$$

[3]Rules $openRules_s(p)$, $posClosure_s(p)$, and $negClosure_s(p)$ are defined in the section that follows Introduction.

$$P_s^{\mathtt{n}} = \{r \in P_s \mid mode_s^{\mathtt{D}}(pred(Head_r)) = \mathtt{n}\}.$$

It holds: $(P_s^{\mathtt{d}} \cup P_s^{\mathtt{o}} \cup P_s^{\mathtt{c}}) \cap P_s^{\mathtt{n}} = \emptyset$.

Example 7 Consider rule base s_1 of Example 2. It holds:

$P_{s_1}^{\mathtt{d}} = P_{s_1}$,
$P_{s_1}^{\mathtt{o}} = P_{s_1} \cup$ {¬`eu:CountryEU(?x)`←`geo:Country(?x)`, ∼ `eu:CountryEU(?x)`.
`eu:CountryEU(?x)`←`geo:Country(?x)`, ∼¬`eu:CountryEU(?x)`.},
$P_{s_1}^{\mathtt{c}} = P_{s_1} \cup$ {¬`eu:CountryEU(?x)`←`geo:Country(?x)`, ∼ `eu:CountryEU(?x)`.}.
$P_{s_1}^{\mathtt{n}} = \{\}$. □

Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let I be a simple normal (resp. extended) interpretation of s w.r.t. $\mathcal{S}$. Let P be a logic program. The logic program $P/_{\mathcal{S}}I$ is obtained from $[P]_{\mathcal{S}}$ as follows:

1. Remove from $[P]_{\mathcal{S}}$, all rules that contain in their body an objective literal L s.t. $I(\neg L) = 1$ or a default literal ${\sim}L$ s.t. $I(L) = 1$.
2. Remove from the body of remaining rules, any default literal ${\sim}L$ s.t. $I(L) = 0$.
3. Replace all remaining default literals ${\sim}L$ with $\mathtt{u}$. □

The above $P/_{\mathcal{S}}I$ modulo transformation is actually an adaptation of the P/I modulo transformation of `WFSX` (Pereira and Alferes 1992) to our framework. Note that this transformation extends the P/I modulo transformation of `AS` (Gelfond and Lifschitz 1990).

Example 8 Consider the modular rule base of Example 2.

Let $P = \{$ `gov:Enter(?p)` ← `eu:CountryEU(?c)`, `sec:citizenOf(?p,?c)`, ∼ `sec:Suspect@(?p)`. }.

Consider now the simple *normal* interpretation of s_4 w.r.t. $\mathcal{S}$, $I = \{$`sec:Suspect@(Peter)`$\}$. Then,

$P/_{\mathcal{S}}I = \{$ `gov:Enter`(p) ← `eu:CountryEU`(c), `sec:citizenOf`(p, c). | $p \in \mathtt{HU}_{\mathcal{S}} - \{$`Peter`$\}$ and $c \in \mathtt{HU}_{\mathcal{S}}$ }.

Additionally, consider the simple *extended* interpretation of s_4 w.r.t. $\mathcal{S}$, $I = \{$`sec:Suspect@(Peter)`$\}$. Then,

$P/_{\mathcal{S}}I = \{$ `gov:Enter`(p) ← `eu:CountryEU`(c), `sec:citizenOf`(p, c), $\mathtt{u}$. | $p \in \mathtt{HU}_{\mathcal{S}} - \{$`Peter`$\}$ and $c \in \mathtt{HU}_{\mathcal{S}}$ }. □

We are now ready to define the normal and extended answer sets of a modular rule base.

Definition 7 (Normal & extended answer sets of a MRB) Let $\mathcal{S}$ be a modular rule base. A *normal* (resp. *extended*) *answer set* of $\mathcal{S}$ is the minimum (w.r.t. $\leq_{\mathtt{t}}$) normal (resp. extended) interpretation of $\mathcal{S}$, $\mathsf{M} = \{M_s \mid s \in \mathcal{S}\}$, such that (for $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$):

1. For all $p \in Pred_s^{\mathtt{D}}$ s.t. $\mathtt{x} > |mode_s^{\mathtt{D}}(p)|$, for all $L \in [p]_{\mathcal{S}}$: $M_s^{\mathtt{x}}(L) \geq M_s^{\mathtt{m}}(L)$, where $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$,
2. For all $p \in Pred_s^{\mathtt{U}}$, and for all $s' \in \mathcal{S}$ s.t. $Nam_{s'} \in Import_s^{\mathcal{S}}(p)$:
   (a) for all $L \in [p]_{\mathcal{S}}$: $M_s^{\mathtt{x}}(L) \geq M_{s'}^{\mathtt{m}}(L)$, where $\mathtt{m} = least(\mathtt{x}, mode_s^{\mathtt{U}}(p))$,

(b) for all $L \in [p@Nam_{s'}]_{\mathcal{S}}$:
$M_s^{\mathtt{x}}(L) \geq M_{s'}^{\mathtt{m}}(simple(L))$,
where $\mathtt{m} = least(\mathtt{x}, mode_s^{\mathtt{U}}(p))$,

3. $M_s^{\mathtt{x}} \models P_s^{\mathtt{x}} /_{\mathcal{S}} M_s^{\mathtt{x}}$.

We denote the set of normal answer sets of $\mathcal{S}$ by $\mathcal{M}^{\mathtt{AS}}(\mathcal{S})$ and the set of extended answer sets of $\mathcal{S}$ by $\mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$. □

Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $\mathsf{M} = \{M_s \mid s \in \mathcal{S}\}$ be a normal (resp. extended) answer set of $\mathcal{S}$. Intuitively, Definition 7 expresses that if L is a literal defined in a rule base s at reasoning mode $\mathtt{m}$ then the truth value of L, according to M_s at reasoning mode $\mathtt{x} \geq |\mathtt{m}|$, is greater or equal to the truth value of L, according to M_s at reasoning mode $|\mathtt{m}|$. If L be a literal imported in a rule base s from a rule base s' at requesting reasoning mode $\mathtt{y}$ then the truth value of L, according to M_s at reasoning mode $\mathtt{x}$, is greater or equal to the truth value of L, according to $M_{s'}$ at reasoning mode $\mathtt{m} = least(\mathtt{x}, \mathtt{y})$. Additionally, it holds: $M_s^{\mathtt{x}} \models P_s^{\mathtt{x}} /_{\mathcal{S}} M_s^{\mathtt{x}}$, for $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$.

Example 9 Consider the modular rule base $\mathcal{S}$ of Example 2. Let $L = \neg$`eu:CountyEU(Croatia)`.

For all $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S})$, it holds: $M_{s_1}^{\mathtt{d}}(L) = 0$, $M_{s_1}^{\mathtt{o}}(L) \in \{0, 1\}$, and $M_{s_1}^{\mathtt{c}}(L) = M_{s_1}^{\mathtt{n}}(L) = 1$. Additionally, it holds: $M_{s_4}^{\mathtt{d}}(L) = 0$, $M_{s_4}^{\mathtt{o}}(L) \in \{0, 1\}$, and $M_{s_4}^{\mathtt{c}}(L) = M_{s_4}^{\mathtt{n}}(L) \in \{0, 1\}$. Furthermore, it holds: $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Anne)`) $= 1$, $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Boris)`) $= 1$, and $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Peter)`) $= 0$.

For all $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$, it holds: $M_{s_1}^{\mathtt{d}}(L) = 0$, $M_{s_1}^{\mathtt{o}}(L) \in \{0, 1/2, 1\}$, and $M_{s_1}^{\mathtt{c}}(L) = M_{s_1}^{\mathtt{n}}(L) = 1$. Additionally, it holds: $M_{s_4}^{\mathtt{d}}(L) = 0$, $M_{s_4}^{\mathtt{o}}(L) \in \{0, 1/2, 1\}$, and $M_{s_4}^{\mathtt{c}}(L) = M_{s_4}^{\mathtt{n}}(L) \in \{0, 1/2, 1\}$. Furthermore, it holds: $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Anne)`) $= 1$, $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Boris)`) $\in \{1/2, 1\}$, and $M_{s_4}^{\mathtt{n}}$(`gov:Enter(Peter)`) $= 0$. □

Let $\mathcal{S}$ be a modular rule base and let M be a normal (resp. extended) answer set of $\mathcal{S}$. Let $s \in \mathcal{S}$ and let L be an objective literal, whose predicate p is defined in s. The following proposition relates the truth values of L, according to the different reasoning modes of M_s.

Proposition 1 Let $\mathcal{S}$ be a modular rule base and let $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S}) \cup \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$. Let $s \in \mathcal{S}$ and let $p \in Pred_s^{\mathtt{D}}$. Additionally, let $\mathtt{x}, \mathtt{y} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\mathtt{x} \leq |mode_s^{\mathtt{D}}(p)| \leq \mathtt{y}$. It holds that:

1. For all $L \in [p]_{\mathcal{S}}$, $M_s^{\mathtt{x}}(L) \leq M_s^{\mathtt{m}}(L)$, where $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$.
2. If $M_s^{\mathtt{y}}$ is consistent then: for all $L \in [p]_{\mathcal{S}}$, $M_s^{\mathtt{y}}(L) = M_s^{\mathtt{m}}(L)$, where $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$. □

Let $\mathcal{S}$ be a modular rule base and let M be a normal (resp. extended) answer set of $\mathcal{S}$. Let L be a literal imported in a rule base s from a rule base s'. Proposition 2 relates the truth value of L w.r.t. M_s with the truth value of L w.r.t. $M_{s'}$.

Proposition 2 Let $\mathcal{S}$ be a modular rule base and let $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S}) \cup \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$. Let $s \in \mathcal{S}$ and let $p \in Pred_s^{\mathtt{U}}$. Let $s' \in \mathcal{S}$ s.t. $Nam_{s'} \in Import_s^{\mathcal{S}}(p)$. Additionally, let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$ and $\mathtt{m}' = |mode_{s'}^{\mathtt{D}}(p)|$. It holds that:

1. For all $L \in [p]_{\mathcal{S}}$, $M_s^{\mathtt{x}}(L) \geq M_{s'}^{least(\mathtt{x},\mathtt{m},\mathtt{m}')}(L)$, where $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$.
2. If $p \notin Pred_s^{\mathtt{D}}$ and $|Import_s^{\mathcal{S}}(p)| = 1$ then: for all $L \in [p]_{\mathcal{S}}$, $M_s^{\mathtt{x}}(L) = M_{s'}^{least(\mathtt{x},\mathtt{m},\mathtt{m}')}(L)$, where $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. □

The following proposition shows that inconsistency, *local* to a rule base s and reasoning mode $\mathtt{x}$, propagates to (i) reasoning modes of s greater that $\mathtt{x}$, and (ii) to reasoning modes greater or equal to $\mathtt{x}$ of other rule bases s' that import information from s at requesting reasoning mode $\mathtt{m} \geq \mathtt{x}$. In all other cases, reasoning remains unaffected from the local inconsistency.

Proposition 3 Let $\mathcal{S}$ be a modular rule base and let $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S}) \cup \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$. Let $s \in \mathcal{S}$ and $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $M_s^{\mathtt{x}}$ is inconsistent. It holds that:

1. For all $\mathsf{N} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S}) \cup \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$, $N_s^{\mathtt{x}}$ is inconsistent.
2. For all $\mathtt{y} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\mathtt{y} \geq \mathtt{x}$, $M_s^{\mathtt{y}}$ is inconsistent.
3. If $s' \in \mathcal{S}$, $p \in Pred_{s'}^{\mathtt{U}}$, $Nam_s \in Import_{s'}^{\mathcal{S}}(p)$, and $mode_{s'}^{\mathtt{U}}(p) \geq \mathtt{x}$ then $M_{s'}^{\mathtt{x}}$ is inconsistent. □

Below, we define `MWebAS` and `MWebWFS` entailment over a rule base s w.r.t. a modular rule base $\mathcal{S}$.

Definition 8 (`MWebAS` & `MWebWFS` entailment) Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let:

1. $p \in Pred_s^{\mathtt{D}}$, $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$, and $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$, or
2. $p \in Pred_s^{\mathtt{U}} - Pred_s^{\mathtt{D}}$, $\mathtt{m} = mode_s^{\mathtt{U}}(p)$, $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$, or
3. $p \in Pred_s^{\mathtt{U}}$, $\mathtt{m} = mode_s^{\mathtt{U}}(p)$, $Nam_{s'} \in Import_s^{\mathcal{S}}(p)$, and $L \in [p@Nam_{s'}]_{\mathcal{S}} \cup \sim[p@Nam_{s'}]_{\mathcal{S}}$.

We say that:

- *s entails* L w.r.t. $\mathcal{S}$ under `MWebAS` ($s \models_{\mathcal{S}}^{\mathtt{mAS}} L$) iff for all $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}(\mathcal{S})$, $M_s^{\mathtt{m}}(L) = 1$.
- *s entails* L w.r.t. $\mathcal{S}$ under `MWebWFS` ($s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$) iff for all $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}(\mathcal{S})$, $M_s^{\mathtt{m}}(L) = 1$. □

Example 10 Consider the modular rule base $\mathcal{S}$ of Example 2. For $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$, it holds: $s_1 \models_{\mathcal{S}}^{\mathtt{SEM}} \neg$`eu:CountryEU(Croatia)`, $s_4 \not\models_{\mathcal{S}}^{\mathtt{SEM}} \neg$`eu:CountryEU(Croatia)`, $s_4 \models_{\mathcal{S}}^{\mathtt{SEM}}$`gov:Enter(Anne)`, and $s_4 \models_{\mathcal{S}}^{\mathtt{SEM}} \sim$`gov:Enter(Peter)`. Additionally, it holds $s_4 \models_{\mathcal{S}}^{\mathtt{mAS}}$`gov:Enter(Boris)`, while $s_4 \not\models_{\mathcal{S}}^{\mathtt{mWFS}}$`gov:Enter(Boris)`. This is because, `MWebAS`, in contrast to `MWebWFS`, supports case-based analysis on the truth values of `eu:CountryEU(Croatia)` and $\neg$`eu:CountryEU(Croatia)` in rule base s_4. □

Properties of `MWebAS` & `MWebWFS`

In this section, we present several properties of the proposed semantics. First, we show that, similarly to `AS` and `WFSX` on ELPs, `MWebAS` is more informative than `MWebWFS`. However, `MWebWFS` has better computational properties.

Proposition 4 Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup \sim\mathtt{HB}_s^{\mathcal{S}}$. It holds that: if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ then $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. □

The following proposition provides the data complexities of $\texttt{MWebAS}$ and $\texttt{MWebWFS}$ semantics. These complexities are the same as the complexities of the answer set ($\texttt{AS}$) and well-founded semantics with explicit negation ($\texttt{WFSX}$) on ELPs, respectively. This result follows from the fact that we can define the $\texttt{MWebAS}$ and $\texttt{MWebWFS}$ semantics of a rule base s w.r.t. a modular rule base $\mathcal{S}$, through appropriately defined ELPs $\Pi^{\texttt{d}}_{s,\mathcal{S}}$, $\Pi^{\texttt{o}}_{s,\mathcal{S}}$, $\Pi^{\texttt{c}}_{s,\mathcal{S}}$, and $\Pi^{\texttt{n}}_{s,\mathcal{S}}$ (not given here due to space limitations).

Proposition 5 Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let $L \in \texttt{HB}^{\mathcal{S}}_s \cup \sim\texttt{HB}^{\mathcal{S}}_s$. It holds that: (i) the problem of establishing if $s \models^{\texttt{mAS}}_{\mathcal{S}} L$ is data complete for co-NP, and (ii) the problem of establishing if $s \models^{\texttt{mWFS}}_{\mathcal{S}} L$ has polynomial data complexity. □

The following proposition shows that $\texttt{MWebAS}$ and $\texttt{MWebWFS}$ semantics extend $\texttt{AS}$ and $\texttt{WFSX}$ semantics on ELPs, respectively. Let P be an ELP. We denote by $\mathcal{C}_{\texttt{SEM}}(P)$ the set of literals, obtained from P under $\texttt{SEM} \in \{\texttt{AS}, \texttt{WFSX}\}$.

Proposition 6 Let s be a rule base s.t. no qualified literal appears in P_s and for all $p \in \mathit{Pred}^{\texttt{D}}_s$, $\mathit{mode}^{\texttt{D}}_s(p) \in \{\texttt{d}, \texttt{n}\}$. Let $\mathcal{S} = \{s\}$, let $p \in \mathit{Pred}^{\texttt{D}}_s$, and let $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$. It holds that: (i) $s \models^{\texttt{mAS}}_{\mathcal{S}} L$ iff $\mathcal{C}_{\texttt{AS}}(P_s)$, and (ii) $s \models^{\texttt{mWFS}}_{\mathcal{S}} L$ iff $\mathcal{C}_{\texttt{WFSX}}(P_s)$. □

Below, we prove that reasoning on the `definite` and `open` predicates (and thus, also `global` predicates) of a modular rule base is monotonic w.r.t. modular rule base extension. Intuitively, a modular rule base $\mathcal{S}$ is *extended* by extending the rule bases in $\mathcal{S}$ and by adding to $\mathcal{S}$ more rule bases. Now, a rule base s is extended by extending: (i) the logic program of s, (ii) the defined predicates p of s, along with their scope, defining reasoning mode (if $\mathit{mode}^{\texttt{D}}_s(p) \in \{\texttt{d}, \texttt{o}\}$), and exporting rule base list, and (iii) the used predicates of s, along with their requesting reasoning mode and importing rule base list. In other words, information and sharing of information in $\mathcal{S}$ is increased.

Definition 9 (Extending modular rule bases) Let $\mathcal{S}$, $\mathcal{S}'$ be modular rule bases. We say that $\mathcal{S}'$ *extends* $\mathcal{S}$ ($\mathcal{S} \leq \mathcal{S}'$) iff for all $s \in \mathcal{S}$, $\exists s' \in \mathcal{S}$:

i. $\mathit{Nam}_s = \mathit{Nam}_{s'}$, $P_s \subseteq P_{s'}$, $\mathit{Pred}^{\texttt{D}}_s \subseteq \mathit{Pred}^{\texttt{D}}_{s'}$, $\mathit{Pred}^{\texttt{U}}_s \subseteq \mathit{Pred}^{\texttt{U}}_{s'}$,

ii. For all $p \in \mathit{Pred}^{\texttt{D}}_s$:
  (a) $\mathit{scope}_s(p) \leq \mathit{scope}_{s'}(p)$ and $\mathit{Export}^{\mathcal{S}}_s(p) \subseteq \mathit{Export}^{\mathcal{S}'}_{s'}(p)$,
  (b) if $\mathit{mode}^{\texttt{D}}_s(p) \in \{\texttt{d}, \texttt{o}\}$ then $\mathit{mode}^{\texttt{D}}_s(p) \leq |\mathit{mode}^{\texttt{D}}_{s'}(p)|$ and $\mathit{mode}^{\texttt{D}}_{s'}(p) \neq \texttt{n}$.
  (c) if $|\mathit{mode}^{\texttt{D}}_s(p)| \in \{\texttt{c}, \texttt{n}\}$ then $\mathit{mode}^{\texttt{D}}_s(p) = \mathit{mode}^{\texttt{D}}_{s'}(p)$.
  (d) if $|\mathit{mode}^{\texttt{D}}_s(p)|$, $|\mathit{mode}^{\texttt{D}}_{s'}(p)| \in \{\texttt{o}, \texttt{c}\}$ then $\mathit{context}_s(p) = \mathit{context}_{s'}(p)$, and

iii. For all $p \in \mathit{Pred}^{\texttt{U}}_s$: $\mathit{mode}^{\texttt{U}}_s(p) \leq \mathit{mode}^{\texttt{U}}_{s'}(p)$ and $\mathit{Import}^{\mathcal{S}}_s(p) \subseteq \mathit{Import}^{\mathcal{S}'}_{s'}(p)$. □

Theorem 1 (Monotonicity) Let $\mathcal{S}$, $\mathcal{S}'$ be modular rule bases s.t. $\mathcal{S} \leq \mathcal{S}'$. Let $s \in \mathcal{S}$, $s' \in \mathcal{S}'$ s.t. $\mathit{Nam}_s = \mathit{Nam}_{s'}$. Let (i) $p \in \mathit{Pred}^{\texttt{D}}_s$ s.t. $\mathit{mode}^{\texttt{D}}_s(p) \in \{\texttt{d}, \texttt{o}\}$, or (ii) $p \in \mathit{Pred}^{\texttt{U}}_s - \mathit{Pred}^{\texttt{D}}_s$ s.t. $\mathit{mode}^{\texttt{U}}_s(p) \in \{\texttt{d}, \texttt{o}\}$. Additionally, let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models^{\texttt{SEM}}_{\mathcal{S}} L$ then $s' \models^{\texttt{SEM}}_{\mathcal{S}'} L$, for $\texttt{SEM} \in \{\texttt{mAS}, \texttt{mWFS}\}$. □

Related Work

Initial ideas for our framework and additional motivation are provided in (Damásio et al. 2006). The combination of open-world and closed-world reasoning, in the same framework, is also proposed in (Analyti et al. 2008), where the stable model semantics of Extended RDF (ERDF) ontologies is developed, based on partial logic (Herre, Jaspars, and Wagner 1999). 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 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. However, modularity issues are not considered there, and local CWAs and OWAs are not declared w.r.t. a context.

A form of local CWA w.r.t. a context is proposed in (Cortés-Calabuig et al. 2005), where the local CWA is applied on the predicates of a *single* data source s, containing positive facts, only. In this work, a context is a first-order formula over the predicates of s. The semantics of the proposed local CWA syntax is defined in first-order logic. Rules, strong negation, and modularity issues are not considered, in this work.

An alternative proposal for local CWAs is present in the `dlvhex` system (Eiter et al. 2005). This answer-set programming system has features, like high-order atoms and external atoms, which are very flexible. For instance, assuming that a generic external atom $KB[C](X)$ is available for querying a concept C in a knowledge base KB then a CWA can be stated as follows: $C'(X) \leftarrow \mathit{concept}(C), \mathit{concept}(C'), \mathit{cwa}(C, C'), o(X), \sim KB[C](X)$, where $\mathit{concept}(C)$ is a predicate which holds for all concepts C, the predicate $\mathit{cwa}(C, C')$ states that C' is the complement of C under the closed world assumption, and $o(X)$ is a predicate that holds for all individuals occurring in KB.

Flora-2 (Yang, Kifer, and Zhao 2003) is a rule-based object-oriented knowledge base system for reasoning with semantic information on the Web. It is based on F-logic (Kifer, Lausen, and Wu 1995) 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, reasoning mode and predicate scope issues are ignored. Additionally, 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 $\mathcal{S}$ is defined, based on the F-logic semantics (Kifer, Lausen, and Wu 1995) of an equivalent rule base with no modules. In particular, each qualified atom $\mathit{subject}[\mathit{predicate} \rightarrow \mathit{object}]@\mathit{Nam}_s$ (where Nam_s is a module name) is translated to $\mathit{subject}[\mathit{predicate}\#\mathit{Nam}_s \rightarrow \mathit{object}]$, where $\mathit{predicate}\#\mathit{Nam}_s$ is a new predicate name.

TRIPLE (Sintek and Decker 2002) 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 (Kifer, Lausen, and Wu 1995) and supports a fragment of RDFS and first-order logic. All variables must be explicitly quantified, either existentially or universally. Arbitrary formulas can be used in the body, while the head of the rules is restricted to atoms or conjunctions of molecules. 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. The semantics of a modular rule base is defined, based on the *well-founded semantics* (WFS) (Gelder, Ross, and Schlipf 1991) of an equivalent logic program. In this work, reasoning mode, predicate scope, and visibility issues are ignored. Additionally, strong negation is not supported.

In (Pontelli, Son, and Baral 2006), a simple modularity framework for rule bases is proposed and its AS semantics is defined. However, in this work, the dependency graph G between the rule bases of a modular rule base $\mathcal{S}$ (formed based on the rule bases' `import` statements) should be acyclic. The answer sets of a module $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined based of the answer sets of the modules that are lower than s in the dependency graph G. In this work, reasoning mode and predicate scope issues are ignored. Additionally, strong negation is not supported. The predicate visibility mechanism is simple, as it is assumed that exported predicates are provided to any requesting rule base.

Another simple modularity framework for rule bases is proposed in (Polleres 2006), where weakly negated rule literals should be qualified and depend (directly or indirectly) on qualified literals, only. In this work, reasoning mode, predicate scope, and visibility issues are ignored. Additionally, strong negation is not supported. The *contextually bounded* AS and *contextually bounded* WFS semantics of a modular rule base $\mathcal{S}$ are defined, through the AS and WFS semantics of an equivalent logic program $\mathcal{S}_{CB}$. $\mathcal{S}_{CB}$ consists of the rules of each rule base $s \in \mathcal{S}$ (called *contexts*, there) union the rules $p@Nam_s(t_1, ..., t_n) \leftarrow Body$, where $p(t_1, ..., t_n) \leftarrow Body$ is a rule defined in a rule base $s \in \mathcal{S}$. Another proposal made in (Polleres 2006) is to qualify all simple atoms appearing in a rule base s by the name of s. The resulting rules union the original rules of each rule base $s \in \mathcal{S}$ form a normal logic program $\mathcal{S}_{CC}$. The *contextually closed* AS and *contextually closed* WFS semantics of a modular rule base $\mathcal{S}$ are defined, through the AS and WFS semantics of $\mathcal{S}_{CC}$.

Modularity for rule bases is also considered in (Brewka, Roelofsen, and Serafini 2007), where the *contextual* AS and the *contextual* WFS semantics of a modular rule base are defined, model-theoretically. However, in this work, reasoning mode, predicate scope, and visibility issues are ignored. Simple literals appearing in a rule base s (called *context*, there) are assumed to be qualified by the name of s. Intuitively, we can say that, if (i) all predicates are defined in `normal` reasoning mode, (ii) all literals appearing in the body of the rules of the rule bases are qualified, and (iii) predicate scope and visibility issues are ignored, MWebAS and *contextual* AS semantics are equivalent. However, this is not true for MWebWFS and *contextual* WFS. Indeed, in contrast to MWebWFS, contextual WFS is not coherent.

Note that all modularity frameworks in (Pontelli, Son, and Baral 2006; Polleres 2006; Brewka, Roelofsen, and Serafini 2007) consider that all weakly negated literals appearing in a rule are qualified and depend (directly and indirectly) on qualified literals, only. Therefore, all frameworks achieve monotonicity of reasoning in the case that a modular rule base is *expanded* with additional rule bases. Our framework achieves also this kind of monotonicity, in the case that all weakly negated rule literals, appearing in $\mathcal{S}$, are defined (i) in local/internal scope or are qualified, and (ii) depend (directly or indirectly) on literals that are defined in local/internal scope or are qualified. However, our framework achieves also a more general kind of monotonicity for `global` predicates that is described in Theorem 1.

Finally, we would like to mention a general framework for multi-context reasoning, proposed in (Brewka and Eiter 2007), that allows to combine arbitrary monotonic and nonmonotonic logics. Information flow between the different contexts is achieved though a set of nonmonotonic bridge rules. In this work, several notions for acceptable belief states for the multicontext system are investigated.

Conclusions

In this paper, we presented a principled framework for modular web rule bases, called MWeb. According to this framework, each predicate p defined in a rule base s is characterized by its defining reasoning mode (`definite`, `open`, `positively closed`, `negatively closed`, or `normal`), scope (`global`, `local`, or `internal`), and exporting rule base list. Each predicate p used in a rule base s is characterized by its requesting reasoning mode and importing rule base list. For valid MWeb modular rule bases $\mathcal{S}$, the MWebAS and MWebWFS semantics of each $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined, model-theoretically. These semantics extend the AS (Gelfond and Lifschitz 1990) and WFSX (Pereira and Alferes 1992; Alferes, Damásio, and Pereira 1995) semantics on ELPs, respectively, keeping all of their semantical and computational characteristics.

Our framework supports: (i) local semantics and different points of view, (ii) local closed-world and open-world assumptions, through the contextual CWA and OWA rules, (iii) scoped negation-as-failure and scoped literal evaluation, through the use of qualified literals, the `local` and `internal` predicate scopes, and the restricted evaluation of literals in an MWeb modular rule base, (iv) restricted propagation of local inconsistencies, making possible reasoning even in the presence of an inconsistency, local to a web rule base and reasoning mode, and (v) monotonicity of reasoning, in the case that the MWeb modular rule base is extended, for definite, open (and thus, also `global`) predicates.

In future work, we plan to define a notion of m-equivalent MWeb rule bases such that, for any modular rule base $\mathcal{S}$ and $s \in \mathcal{S}$, if s is replaced in $\mathcal{S}$ by an m-equivalent rule base s' then the WMeb semantics of the other rule bases in $\mathcal{S}$ will remain unaffected. This is related to the work in (Oikarinen and Janhunen 2006).

Closing, we would like to mention that the modular framework proposed in this paper has been solely motivated by the needs of the Semantic Web community, which have been discussed in several forums for a long time. To the best of our knowledge, this is the first time that all of the above mentioned issues of modularity for rule bases in the web are combined in a single framework with a precise semantics. All of these issues have been identified as phase 2 general directions for extensions of the Rule Interchange Framework (RIF). Unfortunately, no use cases for this phase 2 have yet been published and, thus, no real examples are available for the community. The current proposal is a step in this direction, which requires further exploitation for assessing its practical significance.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments. This research has been partially funded by European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779.

References

Alferes, J. J.; Damásio, C. V.; and Pereira, L. M. 1995. A logic programming system for non-monotonic reasoning. *Journal of Automated Reasoning* 14(1):93–147.

Analyti, A.; Antoniou, G.; Damásio, C. V.; and Wagner, G. 2008. Extended RDF as a Semantic Foundation of Rule Markup Languages. *Journal of Artificial Intelligence Research (JAIR)* 32:37–94.

Baral, C., and Gelfond, M. 1994. Logic programming and knowledge representation. *Journal of Logic Programming* 19/20:73–148.

Brewka, G., and Eiter, T. 2007. Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In *22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, 385–390.

Brewka, G.; Roelofsen, F.; and Serafini, L. 2007. Contextual default reasoning. In *20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, 268–273.

Cortés-Calabuig, A.; Denecker, M.; Arieli, O.; Nuffelen, B. V.; and Bruynooghe, M. 2005. On the local closed-world assumption of data-sources. In *17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'05)*, 333–334.

Damásio, C. V.; Analyti, A.; Antoniou, G.; and Wagner, G. 2006. Supporting open and closed world reasoning on the web. In *4th International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'06)*, 149–163.

Duerst, and Suignard. 2005. Internationalized Resource Identifiers. RFC 3987.

Eiter, T.; Ianni, G.; Schindlauer, R.; and Tompits, H. 2005. A uniform integration of higher-order reasoning and external evaluations in answer-set programming. In *19th Intern. Joint Conference on Artificial Intelligence (IJCAI'05)*, 90–96.

Gelder, A. V.; Ross, K. A.; and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM* 38(3):620–650.

Gelfond, M., and Lifschitz, V. 1990. Logic programs with classical negation. In *7th International Conference on Logic Programming (ICLP'90)*, 579–597.

Heflin, J., and Munoz-Avila, H. 2002. Lcw-based agent planning for the semantic web. In *AAAI Workshop on Ontologies and the Semantic Web*, 63–70.

Herre, H.; Jaspars, J.; and Wagner, G. 1999. Partial Logics with Two Kinds of Negation as a Foundation of Knowledge-Based Reasoning. In Gabbay, D. M., and Wansing, H., eds., *What Is Negation?* Kluwer Academic Publishers.

Kifer, M.; de Bruijn, J.; Boley, H.; and Fensel, D. 2005. A realistic architecture for the semantic web. In *1st International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'05)*, 17–29.

Kifer, M.; Lausen, G.; and Wu, J. 1995. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* 42(4):741–843.

Klyne, G., and Carroll, J. J. 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation. Available at `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210`.

Oikarinen, E., and Janhunen, T. 2006. Modular Equivalence for Normal Logic Programs. In *7th European Conference on Artificial Intelligence (ECAI-2006)*, 412–416.

Pereira, L. M., and Alferes, J. J. 1992. Well founded semantics for logic programs with explicit negation. In *10th European Conference on Artificial Intelligence (ECAI-1992)*, 102–106.

Polleres, A. 2006. Logic programs with contextually scoped negation. In *20th Workshop on Logic Programming (WLP'06)*, 129–136.

Pontelli, E.; Son, T. C.; and Baral, C. 2006. A framework for composition and inter-operation of rules in the semantic web. In *2nd International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06)*, 39–50.

RIF: The Rule Interchange Working Group Charter. Available at `http://www.w3.org/2005/rules/wg/charter`.

Sintek, M., and Decker, S. 2002. TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In *1st International Semantic Web Conference (ISWC'02)*, 364–378.

Yang, G.; Kifer, M.; and Zhao, C. 2003. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *2nd Int. Conf. on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE'03)*, 671–688.