We present a principled framework for modular web rule bases, called \textit{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 legal \textit{MWeb} modular rule bases \( S \), the \textit{MWebAS} and \textit{MWebWFS} semantics of each rule base \( s \in S \) w.r.t. \( S \) are defined model-theoretically. These semantics extend the answer set semantics (\textit{AS}) and the well-founded semantics with explicit negation (\textit{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.

Categories and Subject Descriptors: I.2.3 [\textbf{Artificial Intelligence}]: Deduction and Theorem Proving—Logic programming; Nonmonotonic reasoning and belief revision; I.2.4 [\textbf{Artificial Intelligence}]: Knowledge Representation Formalisms and Methods—Representation languages

General Terms: Theory, Languages, Design

Additional Key Words and Phrases: modular web rule bases, local semantics, local closed-world and open-world assumptions, scoped negation-as-failure.
1. INTRODUCTION

The Semantic Web research [Berners-Lee et al. 2001] 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 [RuleML; Paschke and Boley 2009; Boley and Kifer 2009; RIF; Paschke et al. 2008] has started the discussion on local closed-world assumptions [Heflin and Munoz-Avila 2002] and scoped negation-as-failure (otherwise, called scoped default negation) [Kifer et al. 2005; RIF]. 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 negation (essentially synonymous with the term “negation-as-failure”) is based on the failure to prove a statement 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 some form of modularity, encapsulation, information hiding, and access control. The issue of modularity in logic programming has been actively investigated during the 90s, for a survey see [Bugliesi et al. 1994]. 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 for modular rule bases, called modular web logic framework (MWeb), in which an MWeb modular rule base $S$ is a set of MWeb rule bases\(^1\). Each rule base $s \in S$ can import knowledge about a predicate $p$ from other rule bases in $S$ that define $p$ and are willing to export this knowledge to $s$. When a rule base imports a predicate $p$, it may express that certain non-monotonic reasoning forms on $p$ are not allowed. On the other hand, a rule base that defines a predicate $p$ can use non-monotonic constructs on $p$, knowing that these constructs might be inhibited by an importing rule base. In particular, each predicate $p$ defined or imported by a rule base is associated with a reasoning mode, definite, open, open.

\(^1\)In the rest of the document, by modular rule base, we refer to an MWeb modular rule base and by rule base, we refer to an MWeb rule base.
closed, or normal. These reasoning modes indicate, respectively, that either weak negation is not accepted at all, only open-world assumptions are accepted, both closed-world and open-world assumptions are accepted, or weak negation is fully accepted. Additionally, a rule base \( s \) can indicate that a defined predicate \( p \) 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 to other rule bases. We call these predicates global, local, or internal to \( s \), respectively.

In summary, in this work:

—We describe a language for defining rule bases, and define legality of modular rule bases, through a formal set of syntactic constraints.

—We propose two model-theoretic semantics for (legal) modular rule bases \( \mathcal{S} \), called MWeb answer set semantics (MWebAS) and MWeb well-founded semantics (MWebWFS). These semantics determine entailment for each rule base \( s \in \mathcal{S} \) and extend, respectively, two major semantics for extended logic programs (ELPs), namely answer set semantics (AS) [Gelfond and Lifschitz 1990; 1991] and well-founded semantics with explicit negation (WFSX) [Pereira and Alferes 1992; Alferes and Pereira 1996; Alferes et al. 1995]. We show that, similarly to the corresponding semantics for ELPs, MWebAS is more informative than MWebWFS. However, MWebWFS has better computational properties than MWebAS.

—We show that 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 \( \mathcal{S} \) is increasing\(^2\). 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.

—We identify a special class of predicates \( p \) closed in a rule base \( s \in \mathcal{S} \), referred to as predicates c-stratified in \( s \) w.r.t. \( \mathcal{S} \), for which \( s \) has full-knowledge. This means that for each tuple \( \bar{c} = c_1, ..., c_n \), where \( c_i \) are constants appearing in \( \mathcal{S} \) and \( n \) is the arity of \( p \), rule base \( s \) either entails \( p(\bar{c}) \) or \( \neg p(\bar{c}) \) w.r.t. \( \mathcal{S} \) under both MWebAS and MWebWFS semantics.

—In addition to model-theoretic semantics of a modular rule base \( \mathcal{S} \), we provide equivalent transformational semantics, where for each \( s \in \mathcal{S} \) four ELPs are generated, one for each reasoning mode definite, open, closed, or normal. For MWebAS and MWebWFS semantics of \( \mathcal{S} \), theses ELPs are evaluated through AS and WFSX, respectively.

Interoperation of rule bases over the web is useful in several applications, such as e-Health, e-Business, e-Government, e.t.c. For example, in the biomedical domain, pharmaceutical, laboratory, medical, and patient rule bases may interoperate in order to decide the best medication for a patient under his current condition. More applications, can be found in [Paschke et al. 2008].

Initial ideas for our framework are presented in [Damásio et al. 2006]. However, the operational semantics of modular rule bases, presented there, are not equivalent

---

\(^2\)We call this modular rule base extension and it includes both the addition of new rules in the existing rule bases and the inclusion of new rule bases in \( \mathcal{S} \).

\(^3\)Note that \( \neg p(\bar{c}) \) denotes the strong (or explicit) negation of \( p(\bar{c}) \).
to the ones presented in this paper, since they do not support restricted propagation of local inconsistencies, and the considered language for defining rule bases is simpler. Additionally, a first-version of the model-theoretic MWebAS and MWebWFS semantics of modular rule bases is presented in our conference paper [Analyti et al. 2008]. However, this version of MWebAS and MWebWFS semantics does not lead to monotonic reasoning for global predicates, in the case of modular rule base extension. This paper revises [Analyti et al. 2008] and extends it by (i) providing more examples and explanations, (ii) providing revised and additional properties of our MWeb framework, (iii) identifying the class of c-stratified predicates in a rule base s, for which s has full-knowledge, (iv) providing transformational semantics equivalent to model-theoretic semantics, and (v) providing proofs of all theorems and propositions. A detailed comparison between the MWebAS and MWebWFS semantics of modular rule bases, as presented in this paper, and the corresponding ones, presented in [Analyti et al. 2008], is provided in the Related Work Section in the Electronic Appendix.

Below, we justify our choice for choosing the answer set semantics (AS) [Gelfond and Lifschitz 1990; 1991] and the well-founded semantics with explicit negation (WFSX) [Pereira and Alferes 1992; Alferes and Pereira 1996; Alferes et al. 1995] as our basis for our MWeb framework semantics. The capability of representing open and closed information in the Semantic Web requires the availability in the same language of both monotonic and non-monotonic negation [Wagner 1991; Analyti et al. 2008; Wagner et al. 2008]. The integration of monotonic and non-monotonic reasoning into logic-based formalisms has been extensively studied in the 1990s, and it was concluded that logic programming-based formalisms have sound mechanisms for this purpose and clear semantics. Additionally, they are related to major non-monotonic formalisms, like default logic and non-monotonic modal logics (e.g. see [Bochman 2005]).

The study of weak negation resulted in two major semantics [Gelder et al. 1991] and the stable model semantics [Gelfond and Lifschitz 1988]. The study of strong negation introduced new problems and resulted in two major extensions of the previous semantics, well-founded semantics with explicit negation [Pereira and Alferes 1992; Alferes et al. 1995] and answer set semantics [Gelfond and Lifschitz 1990]. For the non-paraconsistent case, both semantics obey to the “coherence principle”, first specified in [Pereira and Alferes 1992], which states that if something is known to be false then it is believed false (or, more formally, if ¬L then ~L). Other contending semantics, like the partial stable semantics for disjunctive programs (p-stable models) [Przymusinski 1991], do not obey to this intuitive principle.

More recent work in the logic programming literature has been addressing fundamental logical questions regarding the semantics of weak and strong negation, like the (partial) equilibrium logics [Pearce 2006; Cabalar et al. 2006]. This is a work that characterizes logically the existing semantics of logic programming, which were originally defined via fixpoint operators. For instance, the work on partial equilibrium logics of [Cabalar et al. 2006] defines logical characterizations of three semantics for extended logic programs without disjunction, as conservative extensions of partial equilibrium logics with strong negation. In particular, one of the semantics
corresponds to p-stable models [Przymusinski 1991], another to WFSX, and the remaining one to strong negation p-stable models [Alferes and Pereira 1992]. It can be shown that p-stable models semantics is weaker than WFSX, and WFSX is weaker than strong negation p-stable model semantics. All strong negation p-stable models are models of WFSX but, for some programs, WFSX has models while there are no strong negation p-stable models. However, the major shortcoming of strong negation p-stable models is the non-existence of the minimal model, increasing computational complexity and, thus, not being a practical extension of the well-founded semantics with coherent strong negation, while WFSX is.

Moreover, both well-founded semantics with explicit negation and answer set semantics do have available state-of-the-art engines which are used, developed, and actively maintained by the community (like the XSB system [Sagonas et al. 1994], the Smoodels [Niemelä and Simons 1997], and the DLV system [Leone et al. 2006]). Furthermore, the complexity of entailment for both semantics is well-known. In particular, for propositional theories, it is polynomial for WFSX and co-NP-complete for AS on the size of the program. Additionally, WFSX is a better approximation to the skeptical semantics of AS than the p-stable model semantics. All this justifies, our choice of AS and WFSX, as the basis for our work.

The rest of the paper is organized as follows: In Section 2, we informally present the use of weak and strong negation in rule bases. Section 3 introduces our language mechanisms for integrating rule bases over the Web, and provides an informal overview of our MWeb framework. In Section 4, we formally define (legal) modular rule bases. The MWebAS and MWebWFS model-theoretic semantics of modular rule bases are defined in Section 5. Transformational semantics, equivalent to MWebAS and MWebWFS, are provided in Section 6. In Section 7, we provide several properties of MWebWFS and MWebAS. Conclusions are provided in Section 8. Related Work is found in Section A of the Electronic Appendix. The proofs of all Propositions, Theorems, and Corollaries are found in Section B of the Electronic Appendix.

2. WEAK & STRONG NEGATION IN WEB RULE BASES

In this section, we motivate the controlled use of weak negation in rules bases, based on the property of monotonicity. In addition, we show how weak and strong negation can be combined to express local closed-world and open-world assumptions.

First, we introduce some basic concepts. An (absolute) IRI (Internationalized Resource Identifier) reference [Duerst and Suignard 2005] is a Unicode string that is used to provide globally unique names for web resources. It may be represented as a qualified name, that is a colon-separated two-part string consisting of a namespace prefix (an abbreviated name for a namespace IRI) and a local name. For example, given that the namespace prefix ex stands for the namespace IRI http://www.example.org/, the qualified name ex:Riesling (which stands for http://www.example.org/Riesling) is an IRI reference.

A plain RDF literal is a string “s”, where s is a sequence of Unicode characters, or a pair of a string “s” and a language tag t, denoted by “s”@t. A typed RDF literal is a pair of a string “s” and a datatype IRI reference d, denoted by “s”ˆd.

5The Resource Description Framework (RDF) is a framework for modeling meta-data about web resources, recommended by W3C [Klyne and Carroll 2004].
For example, “27”\textsuperscript{\texttt{xsd:integer}} is a typed literal.

In our framework, predicate names are IRI references. Each rule base \( s \) is associated with a name \( \text{Nam}_s \), 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. An (MWeb) atom is a simple atom \( p(t_1, \ldots, t_k) \) or a qualified atom \( p(t_1, \ldots, t_k)@\text{Nam}_t \), where \( p \) is a predicate of arity \( k \), \( t_i \) for \( i = 1, \ldots, k \), are terms and \( \text{Nam}_t \) 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) literal is an objective or a default literal. An (MWeb) rule \( r \) is a formula of the form: \( L \leftarrow L_1, \ldots, L_m, \neg L_{m+1}, \ldots, \neg L_n \), where \( L \) is a simple atom or the strong negation of a simple atom, and \( L_i \) (for \( i \in \{1, \ldots, n\} \)) 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.

In addition to a name, each rule base \( s \) is associated with a logic program \( P_s \). A modular rule base \( S \) is a set of rule bases. In both the MWebAS and the MWebWFS semantics of \( S \), each non-ground rule \( r \) of a rule base \( s \in S \) stands for the set of ground rules, obtained by instantiating the variables in \( r \) with the constants appearing in \( s \).

Convention: In the rest of the paper, we will omit the term MWeb, found in previous definitions inside parenthesis.

In our presentation, variables are prefixed with a question mark symbol (?). Moreover, \( t \) denotes a sequence of terms, \( \pi \) denotes a sequence of variables, and \( c \) denotes a sequence of constants.

**Example 1.** Consider a rule base \( s_1 \) with name: \( \text{Nam}_{s_1} = \text{http://gov.countrY} \). Rule base \( s_1 \) is associated with a logic program \( P_{s_1} \) that expresses immigration laws of an imaginary country \( X \).

\[
\begin{align*}
\text{Enter}(?p) & \leftarrow \text{CountryEU}(?c), \text{citizenOf}(?p,?c). \\
\text{Enter}(?p) & \leftarrow \neg \text{CountryEU}(?c), \text{citizenOf}(?p,?c), \neg \text{RequiresVisa}(?c). \\
\text{Enter}(?p) & \leftarrow \neg \text{CountryEU}(?c), \text{citizenOf}(?p,?c), \text{RequiresVisa}(?c), \text{HasVisa}(?p).
\end{align*}
\]

Notice that all program rules in \( P_{s_1} \) are objective. Predicate Enter captures the following laws:

—A citizen of European Union can enter the country.
—A non European Union citizen can enter the country if a visa is not required.
—A non European Union citizen can enter the country if a visa is required and he/she has it.

These rules are complemented with the following knowledge:

\[\text{To improve readability, namespace prefixes have been eliminated from the example IRIs.}\]
Consider now another rule base $s_2$ with name $\text{Nam}_{s_2} = \text{http://security.int}$ whose associated logic program $P_{s_2}$ is the following:

\[
\begin{align*}
\text{citizenOf}(\text{Anne}, \text{Austria}). & \quad \text{citizenOf}(\text{Chen}, \text{China}). \\
\text{citizenOf}(\text{Boris}, \text{Croatia}). & \quad \text{citizenOf}(\text{Dil}, \text{Djibuti}).
\end{align*}
\]

Then, $S = \{s_1, s_2\}$ is a modular rule base. □

Depending on their reasoning mode, predicates defined in a rule base $s$ are declared as $\text{definite}$, $\text{open}$, $\text{positively closed}$, $\text{negatively closed}$, or $\text{normal}$. In contrast to $\text{normal}$ predicates, $\text{definite}$, $\text{open}$, and $\text{closed}$ predicates impose restrictions on the use of weak negation in their defining rules. Therefore, it is required that definite, open, and closed predicates do not use normal predicates in their defining rules provided by the user. This prevents unintended use of weak negation in the Semantic Web.

In particular, if a predicate $p$ is declared definite in a rule base $s$ then $p$ has to be defined by the user by objective rules, only. Similarly, if a predicate $p$ is declared open in $s$ w.r.t. a predicate $cxt$ then $p$ has to be defined by the user by objective rules, only. However, the definition of $p$ is augmented by our program transformation by the following rules:

\[
\begin{align*}
\text{openRules}_s(p) & = \{\neg p(\overline{\tau}) \leftarrow cxt(\overline{\tau}), \sim p(\overline{\tau}), \quad p(\overline{\tau}) \leftarrow cxt(\overline{\tau}), \sim \neg p(\overline{\tau})\},
\end{align*}
\]

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 there exists $\overline{\tau}$ s.t. $cxt(\overline{\tau})$ is true in an intended model $M$ of $s$ then $p(\overline{\tau})$ or $\neg p(\overline{\tau})$ is true in $M$. If $p$ is declared open in $s$ without context information then $p$ is called freely open in $s$, and we write the OWA rules as: $\text{openRules}_s(p) = \{\neg p(\overline{\tau}) \leftarrow \sim p(\overline{\tau}), \quad p(\overline{\tau}) \leftarrow \sim \neg p(\overline{\tau})\}$.

Similarly, if a predicate $p$ is declared positively or negatively closed in $s$ w.r.t. a context $cxt$ then $p$ has to be defined by the user by objective rules, only. However, the definition of $p$ is augmented by our program transformation by one of the following rules:

\[
\begin{align*}
\text{posClosure}_s(p) & = \{\neg p(\overline{\tau}) \leftarrow cxt(\overline{\tau}), \sim p(\overline{\tau})\},
\end{align*}
\]

called positive contextual CWA rule, or

\[
\begin{align*}
\text{negClosure}_s(p) & = \{p(\overline{\tau}) \leftarrow cxt(\overline{\tau}), \sim \neg p(\overline{\tau})\},
\end{align*}
\]

called negative contextual CWA rule. 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 $\overline{\tau}$ s.t. $cxt(\overline{\tau})$ is true in an intended model $M$ of $s$, if
\(p(\bar{c})\) is believed to be false in \(M\) then \(\neg p(\bar{c})\) is true in \(M\). Similarly, the negative closure rule of a predicate \(p\) in \(s\) expresses that, for any \(\bar{c}\) s.t. \(cxt(\bar{c})\) is true in an intended model \(M\) of \(s\), if \(\neg p(\bar{c})\) is believed to be false in \(M\) then \(p(\bar{c})\) 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. That is, \(\text{posClosure}_s(p) = \{\neg p(\bar{x}) \leftarrow \sim \neg p(\bar{x})\}\) and \(\text{negClosure}_s(p) = \{p(\bar{x}) \leftarrow \sim p(\bar{x})\}\), respectively.

Open and closed predicates appearing in the defining rules of a definite predicate \(p\) are treated, as if they had been declared definite. This means that obtaining the semantics of \(p\), no OWA or CWA rules are added for the open and closed predicates appearing in the defining rules of \(p\). This is because in the definite reasoning mode, weak negation is not accepted at all. Closed predicates appearing in the defining rules of \(p\) are treated, as if they had been declared open. This means that obtaining the semantics of \(p\), instead of CWA rules, OWA rules are added for the closed predicates appearing in the defining rules of \(p\). This is because, in the open reasoning mode, only open-world assumptions are accepted. On the other hand, in closed reasoning mode, both closed-world and open-world assumptions are accepted.

**Example 2.** Returning to Example 1, start by assuming that all predicates are definite. Then, \(\text{Enter}(\text{Anne})\) and \(\text{Enter}(\text{Chen})\) are obtained from \(P_{s_1} \cup P_{s_2}\), under both AS and WFSX. Interestingly, \(\text{Enter}(\text{Boris})\) is not concluded because it is not known that Croatia is a European Union country and also it is not known that it is not a European Union country! One way to circumvent this situation is to state that predicate \(\text{CountryEU}\) is open. By declaring \(\text{CountryEU}\) open in \(s_1\) w.r.t. \(\text{Country}\), the following two rules are added to the definition of \(\text{CountryEU}\):

\[
\text{CountryEU}(?c) \leftarrow \text{Country}(?c), \sim \text{CountryEU}(?c).
\]

\[
\sim \text{CountryEU}(?c) \leftarrow \text{Country}(?c), \sim \text{CountryEU}(?c).
\]

It is now concluded from \(P_{s_1} \cup P_{s_2} \cup \text{openRules}_{s_1}(\text{CountryEU})\) under AS that \(\text{Enter}(\text{Boris})\) holds. The argument is the following: If Croatia is a member of a European Union country then, by the first rule of \(P_{s_1}\), Boris can enter the country. If Croatia is not a European Union country then, by the second rule of \(P_{s_1}\) (since a Visa is not required for Croatia), Boris can also enter the country. Note that WFSX is not capable of doing this case analysis. Therefore, this conclusion is not obtained from \(P_{s_1} \cup P_{s_2}\) under WFSX.

Finally, assume that in addition to the previous declaration, \(\text{Enter}\) is declared positively closed in \(s_1\) w.r.t. \(\text{Person}\). Then, the following rule is added to the definition of \(\text{Enter}\): \(\sim \text{Enter}(?p) \leftarrow \text{Person}(?p), \sim \text{Enter}(?p)\).

This rule expresses that if by the immigration laws it cannot be concluded that a person can enter the country then that person cannot enter the country. Then, it can be concluded from \(P_{s_1} \cup P_{s_2} \cup \text{openRules}_{s_1}(\text{CountryEU}) \cup \text{posClosure}_{s_1}(\text{Enter})\), under both AS and WFSX, that \(\sim \text{Enter}(\text{Dil})\) holds, while all previous inferences remain unaffected.

However, if we add to rule base \(s_1\) the fact \(\sim \text{RequiresVisa}(\text{Djibuti})\), then it can be concluded from \(P_{s_1} \cup P_{s_2} \cup \text{openRules}_{s_1}(\text{CountryEU}) \cup \text{posClosure}_{s_1}(\text{Enter})\).
under both AS and WFSX, that \textit{Enter(0il)} holds. □

It can be observed that if all predicates are definite or open then the addition of new rules, defining old and/or new definite and open predicates, does not affect old conclusions, in both AS and WFSX semantics. Thus, in this case, monotonicity of reasoning is achieved. However, this is not true if some predicates are closed or normal. Monotonicity for definite and open predicates under modular rule base extension is considered in Section 7.1.

3. MODULARITY FOR RULE BASES ON THE WEB

In this section, we introduce the modularity mechanisms of our MWeb framework. Moreover, we discuss the combination of (i) the reasoning mode of a predicate \( p \), defined in a rule base \( s \), and (ii) the reasoning mode in which another rule base \( s' \) requests \( p \). The former reasoning mode is referred to as the \textit{defining reasoning mode} of \( p \) in \( s \) and takes the values \textit{definite}, \textit{open}, \textit{posClosed}, \textit{negClosed}, and \textit{normal}. The latter reasoning mode is referred to as the \textit{requesting reasoning mode} of \( p \) in \( s' \) and takes the values \textit{definite}, \textit{open}, \textit{closed}, and \textit{normal}. We also define the scopes (\textit{global}, \textit{local}, or \textit{internal}) of a predicate \( p \), that is defined in a rule base \( s \). These scopes determine the visibility of \( p \) to other rule bases and its possibilities for re-definition by other rules bases. Further, we discuss the conflicts between the different scopes of the same predicate that has been defined in multiple rule bases.

Let \( S \) be a modular rule base. As seen in Section 2, each rule base \( s \in S \) is associated with a name \( \text{Nam}_s \) and a logic program \( P_s \). However, this information is not enough for determining the way knowledge, distributed over the various rule bases of \( S \), is integrated. Therefore, each rule base \( s \in S \) is also associated with an \textit{interface} \( \text{Int}_s \) that contains two kinds of declarations, \textit{defines} and \textit{uses}, that have the following syntax:

\[
\begin{align*}
\text{DefinesDecl} &::= \text{"defines" ScopeDecl DefinesPred ["visible to" RuleBaseList] "."}, \\
\text{UsesDecl} &::= \text{"uses" UsesPred ["from" RuleBaseList] "."}, \\
\text{ScopeDecl} &::= \text{"global" | "local" | "internal"}, \\
\text{RuleBaseList} &::= \text{RuleBaseIRI ("," RuleBaseIRI)*}, \\
\text{DefinesPred} &::= \text{"definite" | "open" | "posClosed" | "negClosed" | "normal"}, \\
\text{PredicateInd} &::= \text{"wrt context" PredicateInd}, \\
\text{UsesPred} &::= \text{"definite" | "open" | "closed" | "normal"}, \\
\text{PredicateInd} &::= \text{AbsoluteIRI}, \\
\text{RuleBaseIRI} &::= \text{AbsoluteIRI}.
\end{align*}
\]

\textit{defines:} These declarations determine which predicates \( p \) are defined in \( s \) and their defining reasoning mode in \( s \) (\textit{definite, open, posClosed, negClosed, or normal}) through the \textit{DefinesPred} clause. The scope of the predicates \( p \) in \( s \) (\textit{global, local, or internal}) is determined through the \textit{ScopeDecl} clause. The user can state the rule bases to which \( s \) is willing to export \( p \), through the \textit{visible to} clause. If this clause is omitted then \( s \) is willing to export \( p \) to any requesting rule base. Finally, if the \textit{wrt context} clause of an open or closed predicate \( p \) is omitted then \( p \) is assumed to be freely open or freely closed, respectively.

\textit{uses:} These declarations determine which predicates \( p \) are requested by \( s \) and
their requesting reasoning mode in $s$ (definite, open, closed, or normal) through
the $\text{UsesPred}$ clause. The user can state the rule bases from which $s$ requests $p$,
through the $\text{from}$ clause. If this clause is omitted then $s$ requests $p$ from any pro-
viding rule base.

As mentioned above, the scope of a predicate $p$, defined in a rule base $s \in \mathcal{S}$, can
take the following values:

**global**: In this case, predicate $p$ is visible outside $s$ and can be defined by any
other rule base $s' \in \mathcal{S}$ in global or internal scope, only. To guarantee monotonicity
of reasoning for global predicates, the defining reasoning mode of a global predicate
must always be definite or open.

**local**: In this case, predicate $p$ is visible outside $s$ and can be defined by any
other rule base $s' \in \mathcal{S}$ in internal scope, only. Differently to global predicates, no
constraint is imposed on the defining reasoning mode of local predicates.

**internal**: In this case, 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.

**Example 3.** The declaration defines local open $p$ of a rule base $s \in \mathcal{S}$ defines
a local predicate $p$ that is freely open in $s$. Predicate $p$ can be imported from $s$ by
any other rule base $s' \in \mathcal{S}$ that requests $p$ from $s$. □

Assume now that a rule base $s \in \mathcal{S}$ defines a predicate $p$ in a reasoning mode
$m$ and that another rule base $s' \in \mathcal{S}$ imports $p$ from $s$ in a requesting reasoning
mode $m'$ different than $m$. Then, reasoning modes $m$ (declared by exporter) and
$m'$ (declared by importer) are combined as shown in Table I. Note that the final
reasoning mode in which $s'$ imports $p$ from $s$ equals $\text{least}(m, m')$, where $\text{definite} <
\text{open} < \text{closed} < \text{normal}$. 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 it
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.

The rationale for the combination of requesting and defining reasoning modes,
present in Table I, is to respect the intents of the importer and the exporter of
knowledge. First it should be stressed that the definite and open reasoning modes
are monotonic, while the closed reasoning mode is not. When importing a predicate
in definite or open reasoning mode, there is an implicit intent of the importer to
preserve monotonicity of the entailed conclusions. Therefore, the predicate declared
closed by the exporter must be used, as if it was declared in a reasoning mode
preserving monotonicity (i.e. either definite or open). Under the $\text{MWebAS}$ semantics,
the adoption of open mode, for this particular case, allows to potentially extract
more knowledge than in definite mode because alternative models will be considered
according to the open world assumption. For the $\text{MWebWFS}$, the same objective literal
conclusions (i.e. positive or strongly negated atoms) will be obtained.

**Example 4.** Consider two rule bases $s, s' \in \mathcal{S}$ stating, respectively:

defines local $\text{posClosed}$ $p$.  uses open $p$ from $\text{Nam}_{s}$.
Thus, $s$ defines a local predicate $p$ as freely positively closed and $s'$ states that it is willing to accept $p$ from $s$ in open reasoning mode (i.e., the requesting reasoning mode of $p$ in $s'$ is open). Then, according to Table I, rule base $s'$ imports $p$ from $s$, as if $p$ had been declared in $s$ in open reasoning mode. $\Box$

**Example 5.** Consider two rule bases $s, s' \in S$ stating, respectively:

- $s$ defines local negClosed $p$.
- $s'$ uses normal $p$.

Thus, $s$ defines a local predicate $p$ as freely negatively closed and $s'$ states that it is willing to accept $p$ (from any providing source in $S$) in normal reasoning mode. Then, according to Table I, rule base $s'$ imports $p$ from $s$ in closed reasoning mode. $\Box$

As we have already mentioned, it is possible that two rule bases $s, s' \in S$ define the same predicate $p$. However, not all combinations of predicate scopes of the same predicate are allowed. The allowed combinations of predicate scopes are presented in Table II.

<table>
<thead>
<tr>
<th>Table I. Combinations of defining and requesting reasoning modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>importer</td>
</tr>
<tr>
<td>closed</td>
</tr>
<tr>
<td>open</td>
</tr>
<tr>
<td>definite</td>
</tr>
<tr>
<td>definite</td>
</tr>
</tbody>
</table>

As we have already mentioned, it is possible that two rule bases $s, s' \in S$ define the same predicate $p$. However, not all combinations of predicate scopes of the same predicate are allowed. The allowed combinations of predicate scopes are presented in Table II.

<table>
<thead>
<tr>
<th>Table II. Allowed combinations of scopes of the same predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>scope in $s'$</td>
</tr>
<tr>
<td>local</td>
</tr>
<tr>
<td>internal</td>
</tr>
<tr>
<td>global</td>
</tr>
</tbody>
</table>

Obviously, it is an error if a rule base $s$ declares a predicate $p$ as local, and another rule base $s'$ declares the same predicate as global or local. This goes against the notion that there is a sole provider for a local predicate. However, it is allowed to internally define a local predicate of a different rule base, since it is not made public. As an example, it is valid if a rule base $s$ locally defines a predicate $p$ in definite reasoning mode, and another rule base $s'$ internally defines $p$ in closed reasoning mode.

In order to simplify our presentation, we have assumed that predicate indicators and predicate names coincide and are associated with a single arity. However, our theory can be easily extended to the case where a predicate name is possibly associated with multiple arities, while a predicate indicator $p = \text{pred\_name}/n$ is associated with a predicate name $\text{pred\_name} \in \text{IRI}$ and a unique arity $n \in \mathbb{N}$ (as in Prolog systems). In this case, the definition of predicate indicator $\text{PredicateInd}$
in the syntax of the uses declarations should be replaced by: (i) PredicateInd ::= AbsoluteIRI “/” Arity, and (ii) Arity ::= Natural.

The following example combines several concepts of our framework:

**Example 6.** Consider the MWeb modular rule base \( \mathcal{S} = \{ s_1, s_2, s_3, s_4 \} \), shown in Figure 1\(^6\). Rule base \( s_1 \), with \( \text{Nam}_{s_1} = \text{http://europa.eu} \), 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 \( \text{Nam}_{s_2} = \text{http://security. int} \), provides international citizenship information and lists persons suspect of crimes. Rule base \( s_3 \), with \( \text{Nam}_{s_3} = \text{http://geography. int} \), provides geographical information, stating a positively closed list of countries.

Finally, rule base \( s_4 \), with \( \text{Nam}_{s_4} = \text{http://gov. countryY} \), defines the immigration policies of an imaginary country \( Y \), which are supported by the knowledge of the other rule bases in \( \mathcal{S} \). Note that even though \( \text{eu: CountryEU} \) is defined in \( s_1 \), in positively closed reasoning mode, rule base \( s_4 \) imports \( \text{eu: CountryEU} \) from \( s_1 \) in open reasoning mode. Furthermore, note that \( s_4 \) imports \( \text{sec: citizenOf} \) and \( \text{sec: Suspect} \) from \( s_2 \) in definite reasoning mode. Even though \( \text{sec: citizenOf} \) is defined in \( s_2 \) in local scope, \( \text{sec: citizenOf} \) is also defined internally in \( s_4 \) and additional facts about this predicate are stated. This is allowed since the internal information of \( \text{sec: citizenOf} \) 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 \( \text{sec: citizenOf} \), as in:

defines global open \( \text{sec: citizenOf} \) visible to \( \text{http://gov. countryY} \). □

4. FORMALIZATION OF MODULAR RULE BASES

In this section, we formalize modular rule bases and define their legality. We denote the set of IRI references by \( \text{Tri} \) and the set of RDF literals by \( \text{LIT} \). Additionally, we denote the set of variable symbols by \( \text{Var} \). The sets \( \text{Var} \), \( \text{Tri} \), and \( \text{LIT} \) are pairwise disjoint.

**Definition 4.1. (Vocabulary)** An (MWeb) vocabulary \( V \) is a triple \( (\text{RBase}, \text{Pred}, \text{Const}) \), where \( \text{RBase} \subseteq \text{Tri} \) is a set of rule base names, \( \text{Pred} \subseteq \text{Tri} \) is a set of predicate names, and \( \text{Const} \subseteq \text{Tri} \cup \text{LIT} \) is a set of constant symbols. □

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

**Definition 4.2. (Atom)** Let \( V = (\text{RBase}, \text{Pred}, \text{Const}) \) be a vocabulary. An atom over \( V \) is a simple atom \( p(t_1, \ldots, t_n) \) or a qualified atom \( p(t_1, \ldots, t_n)@\text{rbase} \), where \( p \in \text{Pred} \), \( \text{rbase} \in \text{RBase} \), \( n = \text{arity}(p) \), and \( t_i \) is a term over \( V \), for \( i = 1, \ldots, n \). □

**Definition 4.3. (Literal)** Let \( V = (\text{RBase}, \text{Pred}, \text{Const}) \) be a vocabulary. An objective literal over \( V \) is an atom \( A \) or the strong negation \( \neg A \) of an atom \( A \) over

---

\(^6\)To improve readability, namespace prefixes have been eliminated from the IRIs, representing constants.
Rule base $s_1$

\[\text{http://europa.eu}\]

defines local posClosed eu:CountryEU wrt context geo:Country.
uses definite geo:Country from \((\text{http://geography.int})\).

eu:CountryEU(Austria).
edu:CountryEU(Greece). ···

Rule base $s_2$

\[\text{http://security.int}\]

defines local open sec:citizenOf.
defines glocal open sec:Suspect.

sec:citizenOf(Anne,Austria).
sec:citizenOf(Boris,Croatia).
sec:Suspect(Peter).

Rule base $s_3$

\[\text{http://geography.int}\]

defines local posClosed geo:Country.
geo:Country(Egypt).
geo:Country(Canada). ···

Rule base $s_4$

\[\text{http://gov.countryY}\]

defines local normal gov:Enter visible to \((\text{http://security.int})\).
defines local negClosed gov:RequiresVisa wrt context geo:Country.
defines internal open sec:citizenOf.
uses definite geo:Country from \((\text{http://geography.int})\).
uses open eu:CountryEU from \((\text{http://europa.eu})\).
uses definite sec:citizenOf from \((\text{http://security.int})\).
uses definite sec:Suspect from \((\text{http://security.int})\).

\[\text{gov:Enter}(?p) \leftarrow \text{eu:CountryEU}(?c), \text{sec:citizenOf}(?p,?c), \neg \text{sec:Suspect}(?p)@\text{(http://security.int)}.
\]
\[\text{gov:Enter}(?p) \leftarrow \neg \text{eu:CountryEU}(?c), \text{sec:citizenOf}(?p,?c), \neg \text{gov:RequiresVisa}(?c), \neg \text{sec:Suspect}(?p)@\text{(http://security.int)}.
\]

\[\neg \text{gov:RequiresVisa}(\text{Croatia}). \text{sec:citizenOf}(\text{Peter},\text{Greece}).\]

Fig. 1. An MWeb modular rule base

$V$. A default literal over $V$ is the weak negation $\neg L$ of an objective literal $L$ over $V$. A literal over $V$ is an objective or a default literal over $V$. \Box

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

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

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

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

**Definition 4.4. (Logic program)** Let $V = (\text{RBase}, \text{Pred}, \text{Const})$ be a vocabulary. A rule $r$ over $V$ is an expression $L_0 \leftarrow L_1, \ldots, L_m, \neg L_{m+1}, \ldots, \neg L_n$, where: (i) $L_0 \in \text{Lit}^o(V)$ is a simple literal (i.e., $\text{qual}(L_0)$ is undefined), (ii) $L_i \in \text{Lit}^o(V) \cup \{t, u\}$, for $i = 1, \ldots, m$, and (iii) $L_i \in \text{Lit}^o(V)$, for $i = m+1, \ldots, n$. \noindent
We define $\text{Head}_s = L_0$, $\text{Body}_s^+ = \{L_1, ..., L_m\}$, $\text{Body}_s^- = \{L_{m+1}, ..., L_n\}$, and $\text{Body}_s = \text{Body}_s^+ \cup \sim \text{Body}_s^-$.

A logic program over $V$ is a set of rules over $V$. □

The symbols $t$ and $u$ are called special literals and represent the truth values true and undefined, respectively. As a shorthand, we will represent the rules of the form: “$L_0 \leftarrow t$” by the fact “$L_0$.”

As we have seen in the previous section, each rule base $s$ is associated with a name $\text{Name}_s$, a logic program $P_s$, and an interface $\text{Int}_s$ that includes defines and uses declarations.

**Definition 4.5. (Rule base)** Let $V = \langle \text{RBase}, \text{Pred}, \text{Const} \rangle$ be a vocabulary. A rule base $s$ over $V$ is a triple $s = (\text{Name}_s, P_s, \text{Int}_s)$, where: (i) $\text{Name}_s \in \text{RBase}$ is the name of $s$, (ii) $P_s$ is a logic program over $V$, called the logic program of $s$, and (iii) $\text{Int}_s = (\text{Def}_s, \text{Use}_s)$ is the interface of $s$, where:

- $\text{Def}_s$ is a set of tuples $\langle p, sc, mod, cxt, Exp \rangle$, where $p \in \text{Pred}$, $sc \in \{gl, lc, int\}$, $mod \in \{d, o, c^+, c^-\}$, $cxt \in \text{Pred} \cup \{n/a\}$, and $Exp \subseteq \text{RBase} - \{\text{Name}_s\}$ or $Exp = \{\ast\}$.
  
  We define $\text{Pred}_s^0 = \{p \mid \langle p, sc, mod, cxt, Exp \rangle \in \text{Def}_s\}$. If $\langle p, sc, mod, cxt, Exp \rangle \in \text{Def}_s$ then $\text{scope}_s(p) = sc$, $\text{mode}_s^0(p) = mod$, $\text{context}_s(p) = cxt$, and $\text{Export}_s(p) = Exp$.

- $\text{Use}_s$ is a set of tuples $\langle p, mod, Imp \rangle$, where $p \in \text{Pred}$, $mod \in \{d, o, c, n\}$, and $Imp \subseteq \text{RBase} - \{\text{Name}_s\}$ or $Imp = \{\ast\}$.
  
  We define $\text{Pred}_s^1 = \{p \mid \langle p, mod, Imp \rangle \in \text{Use}_s\}$. If $\langle p, mod, Imp \rangle \in \text{Use}_s$ then $\text{mode}_s^1(p) = mod$, and $\text{Import}_s(p) = Imp$. □

Let $s$ be a rule base. We define: $\text{Pred}_s = \text{Pred}_s^0 \cup \text{Pred}_s^1$. Intuitively, each tuple $\langle p, sc, mod, cxt, Exp \rangle \in \text{Def}_s$ 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 $cxt = n/a$. Additionally, if $sc = \text{int}$ and the visible to clause of the defines declaration is missing then $Exp = \{\}$.

However, if $sc \in \{gl, lc\}$ and the visible to clause of the defines declaration is missing then $Exp = \{\ast\}$. 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 $mod = o$ (resp. $mod \in \{c^+, c^-\}$) and $cxt = n/a$.

Similarly, each tuple $\langle p, mod, Imp \rangle \in \text{Use}_s$ 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 $Imp = \{\ast\}$. In this case, $s$ imports $p$ from any providing rule base.

We define7: $|d| = d$, $|o| = o$, $|c^+| = |c^-| = c$, and $|n| = n$. Then, we impose the following total order: $d < o < c < n$, called reasoning mode extension. Additionally,

7This auxiliary definition is needed, because the defining reasoning modes of a predicate $p$ are $\{d, o, c^+, c^-\}$, whereas 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 $\{d, o, c, n\}$.
we impose the following total order on predicate scopes: \( \text{int} < c < \text{gl} \), called predicate scope extension.

**Example 7.** Consider rule base \( s_1 \) of Example 6. Then, \( \text{Def}_{s_1} = \{ \langle \text{eu: CountryEU}, \text{lc}, c^+, \text{geo: Country}, \{\ast\} \rangle \} \) and \( \text{Use}_{s_1} = \{ \langle \text{geo: Country}, d, \{<\text{http://geography.} \rangle \rangle \} \). □

In order for a rule base to be legal, it has to satisfy a number of legality constraints.

**Definition 4.6. (Legal rule base)** A rule base \( s = \langle \text{Nam}_s, P_s, \text{Int}_s \rangle \) is legal iff: (Legality Constraints)

1. If \( \langle p, \text{sc}, \text{mod}, \text{ctx}, \text{Exp} \rangle, \langle p', \text{sc}', \text{mod}', \text{ctx}', \text{Exp}' \rangle \in \text{Def}_s \) then \( \text{sc} = \text{sc}' \), \( \text{mod} = \text{mod}' \), \( \text{ctx} = \text{ctx}' \), and \( \text{Exp} = \text{Exp}' \).

2. If \( \langle p, \text{mod}, \text{Imp} \rangle, \langle p, \text{mod}', \text{Imp}' \rangle \in \text{Use}_s \) then \( \text{mod} = \text{mod}' \) and \( \text{Imp} = \text{Imp}' \).

3. For all \( r \in P_s \):
   - (a) \( \text{pred}((\text{Head}_r)) \in \text{Pred}_s^p \).
   - (b) \( \text{Body}_r \cap \{\text{u}\} = \emptyset \).
   - (c) for all \( L \in \text{Body}_r - \{\text{t}\} \), \( \text{pred}(L) \in \text{Pred}_s^d \cup \text{Pred}_s^u \).

4. For all \( \langle p, \text{sc}, \text{mod}, \text{ctx}, \text{Exp} \rangle \in \text{Def}_s \):
   - (a) if \( \text{mod} \in \{o, c^+, c^-\} \) and \( \text{ctx} \in \text{Pred} \) then \( \text{ctx} \in \text{Pred}_s \) and \( \text{arity}(\text{ctx}) = \text{arity}(p) \).
   - (b) if \( \text{ctx} \in \text{Pred}_s^d \) then \( \text{mode}_s^d(\text{ctx}) \in \{d\} \).
   - (c) if \( \text{ctx} \in \text{Pred}_s^2 \) then \( \text{mode}_s^2(\text{ctx}) \in \{d\} \).
   - (d) if \( \text{mod} \in \{d, n\} \) then \( \text{ctx} = \text{n/a} \).

5. If \( p \in \text{Pred}_s^2 \) and \( \text{scope}_s(p) = \text{gl} \) then \( \text{mode}_s^2(p) \in \{d, o\} \).

6. If \( p \in \text{Pred}_s^d \) and \( \text{scope}_s(p) = \text{int} \) then \( \text{Export}_s(p) = \{\} \).

7. If \( p \in \text{Pred}_s^2 \cap \text{Pred}_u^d \) then \( \text{mode}_s^2(p) \leq |\text{mode}_s^2(p)| \).

8. For all \( r \in P_s \), and for all \( L \in \text{Body}_r \):
   - if \( \text{qual}(L) \in \text{RBase} \) then \( \text{qual}(L) \in \text{Import}_s(\text{pred}(L)) \) or \( \text{Import}_s(\text{pred}(L)) = \{\ast\} \).

9. For all \( r \in P_s \), and for all \( L \in \text{Body}_r \):
   - if \( \text{mode}_s^2(\text{pred}(\text{Head}_r)) \neq \text{n} \) then:
     - (a) \( \text{Body}_r = \{\} \).
     - (b) for all \( L \in \text{Body}_r^2 \), if \( \text{pred}(L) \in \text{Pred}_s^2 \) then \( \text{mode}_s^2(\text{pred}(L)) \neq \text{n} \), and
     - (c) for all \( L \in \text{Body}_r^u \), if \( \text{pred}(L) \in \text{Pred}_s^u \) then \( \text{mode}_s^u(\text{pred}(L)) \neq \text{n} \). □

Let \( s \) be a legal rule base. Constraint 1 of Definition 4.6 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, the special literal \( \text{u} \) should not appear in the body of any rule \( r \in P_s \). Further, for each predicate appearing in the body of the rule \( r \), 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 \( \text{cxt} \). This predicate \( \text{cxt} \) should be defined in \( s \) or requested by \( s \) and have the same arity as \( p \). If \( \text{cxt} \) 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 \( \text{pred}(L) \) from rule base \( \text{qual}(L) \). Constraint 9 expresses that for each rule \( r \in P_s \) if the defining reasoning mode of the predicate appearing in \( \text{Head}_r \) is restricted (i.e., not \text{normal}) then: (i) no default literal should appear in \( \text{Body}_r \), and (ii) the defining (resp. requesting) reasoning mode of each defined (resp. requested) predicate appearing in \( \text{Body}_r \) should also be restricted.

**Example 8.** All rule bases \( s_1, s_2, s_3, \) and \( s_4 \) of Example 6 are legal. \( \square \)

**Definition 4.7. (Modular rule base)** A modular rule base \( S \) over a vocabulary \( V \) is a set of legal rule bases over \( V \). \( \square \)

Let \( S \) be a modular rule base, let \( s \in S \), and let \( p \in \text{Pred}^S \). We define:

\[
\text{Export}^S_s(p) = \begin{cases} 
\{ \{ \text{Nam}_s \mid s' \in S - \{s\} \} & \text{if } \text{Export}^S_s(p) = \{\ast\} \\
\text{Export}^S_s(p) \cap \{ \{ \text{Nam}_s \mid s' \in S \} & \text{otherwise}
\end{cases}
\]

Intuitively, \( \text{Export}^S_s(p) \) denotes the rule bases in \( S \) to which \( s \) is willing to export \( p \). We refer to \( \text{Export}^S_s(p) \) as the exporting rule base list of \( p \) in \( s \) w.r.t. \( S \).

**Example 9.** Consider the modular rule base \( S \) of Example 6. Then, \( \text{Export}^S_{s_1}(\text{sec: citizenOf}) = \{ s_1, s_3, s_4 \} \), while \( \text{Export}^S_{s_2}(\text{sec: citizenOf}) = \{ \ast \} \). Additionally, \( \text{Export}^S_{s_2}(\text{gov: Enter}) = \{ s_2 \} \). \( \square \)

Let \( S \) be a modular rule base, let \( s \in S \), and let \( p \in \text{Pred}^S \). We define:

\[
\text{Import}^S_s(p) = \begin{cases} 
\text{ExportingTo}_S(p, s) & \text{if } \text{Import}^S_s(p) = \{\ast\} \\
\text{Import}^S_s(p) \cap \text{ExportingTo}_S(p, s) & \text{otherwise},
\end{cases}
\]

where \( \text{ExportingTo}_S(p, s) = \{ \text{Nam}_s' \mid s' \in S, \text{Nam}_s \in \text{Export}^S_s(p) \} \).

Intuitively, \( \text{ExportingTo}_S(p, s) \) denotes the rule bases in \( S \) that are willing to export \( p \) to \( s \). Additionally, \( \text{Import}^S_s(p) \) denotes the rule bases in \( S \) from which \( s \) imports \( p \). We refer to \( \text{Import}^S_s(p) \) as the importing rule base list of \( p \) in \( s \) w.r.t. \( S \). Note that: For all \( p \in \text{Pred}_s \), \( \text{Nam}_s \notin \text{Export}^S_s(p) \) and \( \text{Nam}_s \notin \text{Import}^S_s(p) \).

**Example 10.** For the modular rule base \( S \) of Example 6, \( \text{ExportingTo}_S(\text{sec: citizenOf}, s_1) = \{ s_2 \} \). Additionally, \( \text{Import}^S_{s_2}(\text{sec: citizenOf}) = \{ s_1 \} \). \( \square \)

In order for a modular rule base to be legal, it has to satisfy a number of legality constraints.

**Definition 4.8. (Legal modular rule base)** A modular rule base \( S \) is legal iff:

**(Legality Constraints)**
In this section, we propose the MWeb answer set semantics (MWebAS) and the MWeb well-founded semantics (MWebWFS) of modular rule bases. We will show that these semantics extend the answer set semantics (AS) [Gelfond and Lifschitz 1990] and the well-founded semantics with explicit negation (WFSX) [Pereira and Alferes 1992; Alferes et al. 1995] on ELPs, respectively.

**Convention:** In this Section, by \( \mathcal{S} \), we denote a modular rule base, by \( s \), we denote a rule base \( s \in \mathcal{S} \), and by \( m \), we denote a reasoning mode \( m \in \{d, o, c, n\} \).

5. MODEL-THEORETIC SEMANTICS FOR MODULAR RULE BASES

Let \( \mathcal{S} \) be a legal modular rule base. Constraint 1 of Definition 4.8 expresses that each rule base in \( \mathcal{S} \) should be a legal rule base. Constraint 2 expresses that distinct rule bases in \( \mathcal{S} \) should have distinct names. Constraint 3 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, \( \text{Import}_s(p) = \{s\} \) or \( \text{Import}_s(p) \subseteq \text{ExportingTo}_s(p, s) \).

**Example 11.** Modular rule base \( \mathcal{S} \) of Example 6 is legal. □
5.1 Normal & Extended Interpretations of a Rule Base

In this subsection, we define the simple normal and extended interpretations of a rule base w.r.t. a modular rule base. Based on these, we define the normal and extended interpretations of a rule base in a reasoning mode w.r.t. a modular rule base.

First, we define the Herbrand universe of a modular rule base, as the union of its constituent rule bases.

**Definition 5.1. (Herbrand universe of a MRB)** Let $S = \{s_1, ..., s_n\}$. The Herbrand universe of $S$ is defined as: $HU_S = HU_{s_1} \cup ... \cup HU_{s_n}$, where $HU_{s_i}$ (for $i = 1, ..., n$) is the set of constants appearing in $P_{s_i}$. □

Let $V = \{RBase, Pred, Const\}$ be a vocabulary, $p \in Pred$, $k = \text{arity}(p)$, and rbase $\in RBase$. Additionally, let $S$ be a modular rule base. We denote by $[p]_S$ the set of literals $p(c_1, ..., c_k)$ and $\neg p(c_1, ..., c_k)$, where $c_i \in HU_S$, for $i = 1, ..., k$. Additionally, we denote by $[\text{rbase}]_S$ the set of literals $p(c_1, ..., c_k)@\text{rbase}$ and $\neg p(c_1, ..., c_k)@\text{rbase}$, where $c_i \in HU_S$, for $i = 1, ..., k$.

Based on the Herbrand Universe of $S$, we define the Herbrand base of $s$ w.r.t. $S$.

**Definition 5.2. (Herbrand base of a rule base w.r.t. a MRB)** The Herbrand base of $s$ w.r.t. $S$ is defined as:

$$HB^S = \{[p]_S \mid p \in Pred_s\} \cup \{[\text{rbase}]_S \mid p \in Pred^p_s, \text{rbase} \in \text{Import}^S(p)\}. □$$

**Definition 5.3. (Simple normal interpretation of a rule base w.r.t. a MRB)** A simple normal interpretation of $s$ w.r.t. $S$ is a set $I \subseteq HB^S$ s.t. $I \cap \sim I = \emptyset$ (consistency) or $I = HB^S$. □

Let $I$ be a simple normal interpretation of $s$ w.r.t. $S$. If $I = HB^S$ then $I$ is called inconsistent. Otherwise, $I$ is called consistent. As usual, $I$ can be seen, equivalently, as a function from $HB^S \rightarrow \{0, 1\}$, where: (i) $I(L) = 1$, if $L \in I$, and (ii) $I(L) = 0$, if $L \notin I$. Let $L \in HB^S$. We define: (i) $I(\sim L) = 1 - I(L)$, if $I$ is consistent, and (ii) $I(\sim L) = 1$, otherwise. $I$ also assigns a truth value to special literal $t$. In particular, we define $I(t) = 1$.

Let $I$ be a simple normal interpretation of $s$ w.r.t. $S$. It easy to see that for all $L \in HB^S$, $I(\sim L) = 1$ implies $I(\sim L) = 1$ (coherency), but not vice-versa.

**Definition 5.4. (Simple extended interpretation of a rule base w.r.t. a MRB)** A simple extended interpretation of $s$ w.r.t. $S$ is a set $I = T \cup \sim F$, where $T, F \subseteq HB^S$ s.t. either:

$$-T \cap \sim T = \emptyset \text{ and } T \cap F = \emptyset \text{ (consistency), or}$$

$$-T = F = HB^S. □$$

Let $T = T \cup \sim F$ be a simple extended interpretation of $s$ w.r.t. $S$. If $I = HB^S \cup \sim HB^S$ then $I$ is called inconsistent. Otherwise, $I$ is called consistent. As usual, $I$ can be seen, equivalently, as a function from $HB^S \cup \sim HB^S \rightarrow \{0, 1, 2, 1\}$, 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 truth values to special literals $t$ and $u$. In particular, we define $I(t) = 1$ and $I(u) = 1/2$. If $\sim T \subseteq F$ then $I$ is called coherent.
Let \( I \) be a simple normal or extended interpretation of \( s \) w.r.t. \( S \) and let \( S \subseteq \text{HB}_{n}^{S} \cup \neg\text{HB}_{n}^{S} \cup \{ t, u \} \). We define: \( I(S) = \min \{|I(L)| \mid L \in S \} \).

**Definition 5.5. (Dependencies of a rule base in a reasoning mode w.r.t. a MRB)** The dependencies of \( s \) in reasoning mode \( m \) w.r.t. \( S \), denoted by \( D_{s,a}^{m} \), is the minimal set of pairs \( \langle s', x \rangle \), where \( s' \in S \) and \( x \in \{ d, o, c, n \} \), that satisfies the following constraints:

1. \( \langle s, m \rangle \in D_{s,a}^{m} \),
2. If \( \langle s', x \rangle \in D_{s,a}^{m} \) and there exists \( p \in \text{Pred}^{p} \) s.t. \( x > |\text{mode}^{p}(p)| \) then \( \langle s', y \rangle \in D_{s,a}^{m} \), where \( y = |\text{mode}^{p}(p)| \),
3. If \( \langle s', x \rangle \in D_{s,a}^{m} \) and there exists \( p \in \text{Pred}^{p} \) s.t. \( \text{Nam}_{s'} \in \text{Import}^{p}(p) \), for \( s'' \in S \), then \( \langle s'', y \rangle \in D_{s,a}^{m} \), where \( y = \text{least}(x, \text{mode}^{p}(p), |\text{mode}^{p}(p)|) \). ⊢

Intuitively, if \( \langle s', x \rangle \in D_{s,a}^{m} \) and \( \langle s', x \rangle \neq \langle s, m \rangle \) then the meaning of the predicates in \( \text{Pred}_{s} \) in rule base \( S \) in reasoning mode \( m \) w.r.t. \( S \) depends on the meaning of the predicates in \( \text{Pred}_{s'} \) in rule base \( s' \) in reasoning mode \( x \) w.r.t. \( S \).

**Example 12.** Consider the modular rule base \( S \) of Example 6. It holds that: \( D_{s_a,c}^{m} = \{ \langle s_1, c \rangle, \langle s_3, d \rangle \} \). Thus, the meaning of \( \text{eu:CountryEU} \) and \( \text{geo:Country} \) in rule base \( s_1 \) in reasoning mode \( c \) depends on the meaning of the predicate \( \text{geo:Country} \) in rule base \( s_3 \) in reasoning mode \( d \). Similarly, it holds that: \( D_{s_a,n}^{m} = \{ \langle s_4, n \rangle, \langle s_1, c \rangle, \langle s_4, o \rangle, \langle s_1, o \rangle, \langle s_2, d \rangle, \langle s_3, d \rangle \} \). ⊲

**Definition 5.6. (Normal & extended interpretation of a rule base in a reasoning mode w.r.t. a MRB)** A normal (resp. extended) interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \) is a set \( I = \{ I_{s}^{p} \mid \langle s', x \rangle \in D_{s,a}^{m} \} \), such that:

1. for all \( \langle s', x \rangle \in D_{s,a}^{m} \), \( I_{s}^{p} \) is a simple normal (resp. extended) interpretation of \( s' \) w.r.t. \( S \), and
2. if there exists \( \langle s', x \rangle \in D_{s,a}^{m} \) s.t. \( I_{s}^{p} \) is inconsistent then for all \( \langle s', x \rangle \in D_{s,a}^{m} \), \( I_{s}^{p} \) is inconsistent. ⊢

Let \( I \) be a normal (resp. extended) interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \). In the case that all simple normal (resp. extended) interpretations in \( I \) are consistent then \( I \) is called consistent. Otherwise, \( I \) is called inconsistent. Note that there is only one inconsistent normal (resp. extended) interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \).

**Example 13.** Consider the modular rule base \( S \) of Example 6. Let \( I = \{ I_{s_1}^{p}, I_{s_3}^{p} \} \), where \( I_{s_1}^{p} = \{ \neg \text{eu:Country(Canada)} \} \) and \( I_{s_3}^{p} = \{ \text{geo:Country(Egypt)} \} \). Then, \( I \) is a (consistent) normal interpretation of \( s \) in reasoning mode \( c \) w.r.t. \( S \). Now, let \( J = \{ J_{s_1}^{p}, J_{s_3}^{p} \} \), where \( J_{s_1}^{p} = \{ \neg \text{eu:Country(Canada)} \}, \sim \text{eu:Country(Egypt)} \} \) and \( J_{s_3}^{p} = \{ \text{geo: Country(Egypt)} \} \). Then, \( J \) is a (consistent) extended interpretation of \( s \) in reasoning mode \( c \) w.r.t. \( S \). ⊲

Below, we order the normal and extended interpretations of a rule base \( s \) in reasoning mode \( m \) w.r.t. \( S \), according to truth ordering \( (\leq_{t}) \) and knowledge ordering \( (\leq_{k}) \) [Fitting 1991].
Definition 5.7. (Truth and knowledge ordering of normal & extended interpretations) Let \( I \) and \( J \) be normal (resp. extended) interpretations of \( s \) in reasoning mode \( \mathfrak{m} \) w.r.t. \( S \). We say that:

- \( J \) extends \( I \) w.r.t. truth \((l \leq t) J \) iff:
  
  \[
  \text{For all } \langle s',x \rangle \in D^S_{S_a,n} \text{ and for all } L \in \mathbb{HB}^S, \quad I^p_s(L) \leq J^p_s(L).
  \]

- \( J \) extends \( I \) w.r.t. knowledge \((l \leq k) J \) iff:
  
  \[
  \text{For all } \langle s',x \rangle \in D^S_{S_a,n}, \quad I^p_s \subseteq J^p_s. \quad \Box
  \]

Note that if \( I \) and \( J \) are normal interpretations then \( I \leq_k J \) iff \( I \leq_t J \). However, this is not true for extended interpretations.

5.2 MWeb Answer Set & Well-Founded Entailment

In this subsection, we define the normal and extended answer sets of a rule base in a reasoning mode \( \mathfrak{m} \) w.r.t. a modular rule base. Based on these, we define MWebAS and MWebWFS entailment over a rule base w.r.t. a modular rule base.

Before we define the normal and extended answer sets of a rule base in a reasoning mode \( \mathfrak{m} \) w.r.t. a modular rule base, a few auxiliary definitions are provided. Let \( P \) be a logic program. We will denote by \([P]_S\) the set of rules in \( P \) instantiated over \( \mathcal{HUs} \).

Below, we define logic program satisfaction, as usual.

Definition 5.8. (Logic program satisfaction) Let \( I \) be a simple normal (resp. extended) interpretation of \( s \) w.r.t. \( S \). We say that \( I \) satisfies a logic program \( P \) \((I \models P)\) iff for all \( r \in [P]_S \), \( I(\text{Head}_r) \geq I(\text{Body}_r) \). \( \Box \)

For each \( s \in S \), we define\(^8\) four logic programs that correspond to the four reasoning modes of \( s \), that is definite, open, closed, and normal. These logic programs will be used in defining the MWebAS and MWebWFS semantics of \( s \) in a reasoning mode \( \mathfrak{m} \) w.r.t. \( S \).

\[
P^d_s = \{ r \in P_s \mid \text{mode}^p_s(\text{pred}(\text{Head}_r)) \neq \text{n} \}. \]

\[
P^o_s = \{ r \in P_s \mid \text{mode}^p_s(\text{pred}(\text{Head}_r)) \in \{ o, c^+, c^- \} \} \cup \{ \text{openRules}_s(p) \mid p \in \text{Pred}_s^o \text{ and } \text{mode}^p_s(p) \in \{ o, c^+, c^- \} \}. \]

\[
P^c_s = \{ r \in P_s \mid \text{mode}^p_s(\text{pred}(\text{Head}_r)) \in \{ c^+, c^- \} \} \cup \{ \text{postClosure}_s(p) \mid p \in \text{Pred}_s^c \text{ and } \text{mode}^p_s(p) = c^+ \} \cup \{ \text{negClosure}_s(p) \mid p \in \text{Pred}_s^c \text{ and } \text{mode}^p_s(p) = c^- \}. \]

\[
P^n_s = \{ r \in P_s \mid \text{mode}^p_s(\text{pred}(\text{Head}_r)) = \text{n} \}. \]

It holds that: \((P^d_s \cup P^o_s \cup P^c_s) \cap P^n_s = \emptyset\).

Intuitively, \( P^d_s \) is the set of rules in \( P_s \) whose head predicate is defined in definite, open, or closed reasoning mode. \( P^o_s \) is the set of rules in \( P_s \) whose head predicate \( p \) is defined in open or closed reasoning mode union the open rules of \( p \) in \( s \). \( P^c_s \) is

\(^8\)Rules \( \text{openRules}_s(p) \), \( \text{postClosure}_s(p) \), and \( \text{negClosure}_s(p) \) are defined in Section 2.
the set of rules in $P_s$ whose head predicate $p$ is defined in closed reasoning mode union the positive closure rule of $p$ in $s$, if $p$ is positively closed, or the negative closure rule of $p$ in $s$, if $p$ is negatively closed. Finally, $P^c_s$ is the set of rules in $P_s$ whose head predicate is defined in normal reasoning mode.

Example 14. Consider rule base $s_1$ of Example 6. It holds that:

$P^d_{s_1} = P_{s_1}$,

$P^p_{s_1} = P_{s_1} \cup \{ \neg \text{eu:CountryEU}(?x) \leftarrow \text{geo:Country}(?x), \sim \text{eu:CountryEU}(?x)., \}

\quad \text{eu:CountryEU}(?x) \leftarrow \text{geo:Country}(?x), \sim \text{eu:CountryEU}(?x)., \}

P^c_{s_1} = P_{s_1} \cup \{ \neg \text{eu:CountryEU}(?x) \leftarrow \text{geo:Country}(?x), \sim \text{eu:CountryEU}(?x)., \}

P^p_{s_1} = \{ \}.

The following $P/_{S}\text{AS}_I$ modulo transformation is used in defining the normal answer sets of a rule base in a reasoning mode w.r.t. a modular rule base. This is actually an adaptation of the $P/I$ modulo transformation of AS [Gelfond and Lifschitz 1990] (also known as Gelfond-Lifschitz transformation).

Definition 5.9. (Transformation $P/_{S}\text{AS}_I$). Let $S$ be a modular rule base, let $s \in S$, and let $I$ be a simple normal interpretation of $s$ w.r.t. $S$. Let $P$ be a logic program. The logic program $P/_{S}\text{AS}_I$ is obtained from $[P]_S$ as follows:

1. Remove from $[P]_S$, all rules that contain in their body a default literal $\sim L$ s.t. $I(L) = 1$.
2. Remove from the body of the remaining rules, any default literal $\sim L$ s.t. $I(L) = 0$.

Similarly, we define the $P/_{S}\text{WFS}_I$ modulo transformation, which is is actually an adaptation of the $P/I$ modulo transformation of WFSX [Pereira and Alferes 1992] to our framework. The $P/_{S}\text{WFS}_I$ modulo transformation is used in defining the extended answer sets of a rule base in a reasoning mode w.r.t. a modular rule base.

Definition 5.10. (Transformation $P/_{S}\text{WFS}_I$). Let $I$ be a simple extended interpretation of $S$ w.r.t. $S$. Let $P$ be a logic program. The logic program $P/_{S}\text{WFS}_I$ is obtained from $[P]_S$ as follows:

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

Example 15. Consider the modular rule base $S$ of Example 6.

Let $P = \{ \text{gov:Enter}(?p) \leftarrow \text{eu:CountryEU}(?c), \text{sec:citizenOf}(?p,?c), \sim \text{sec:Suspect}(?p) \@ \text{Nam}_{s_2}. \}$.

Consider now the simple normal interpretation of $s_4$ w.r.t. $S$, $I = \{ \text{sec:Suspect}(\text{Peter}) \@ \text{Nam}_{s_2}, \sim \text{eu:CountryEU} (\text{Egypt}) \}$. Then, $P/_{S}\text{AS}_I = \{ \text{gov:Enter}(p) \leftarrow \text{eu:CountryEU}(c), \text{sec:citizenOf}(p,c). \mid p \in \text{HU}_S \setminus \{\text{Peter}\} \text{ and } c \in \text{HU}_S \}$. 

21
Additionally, consider the simple extended interpretation of $s_4$ w.r.t. $S$, $I = \{\text{sec:Suspect(Peter)}\odot\text{Nam}_s, \neg\text{eu:CountryEU(Egypt)}, \neg\text{eu:CountryEU(Egypt)}\}$. Then,
\[
P_{/\Lambda_{SS}}^s I = \{ \text{gov:Enter}(p) \leftarrow \text{eu:CountryEU(c)}, \text{sec: citizenOf}(p, c), u \mid p \in \text{HU} - \{\text{Peter}\} \text{ and } c \in \text{HU} - \{\text{Egypt}\}\}.
\]

Let $N$ be a normal (resp. extended) interpretation of $s$ in reasoning mode $m$ w.r.t. $S$. Below, we define the \textit{minimal model} of $s$ in reasoning mode $m$ w.r.t. $S$ and $N$.

**Definition 5.11.** (Minimal model of a rule base in a reasoning mode w.r.t. a MRB and a normal or extended interpretation) Let $N = \{N_s^x \mid \langle s', x \rangle \in D_{s,n}^S\}$ be a normal (resp. extended) interpretation of $s$ in reasoning mode $m$ w.r.t. $S$. The \textit{minimal model} of $s$ in reasoning mode $m$ w.r.t. $S$ and $N$, denoted by $\text{least}_{S,N}^m(s)$, is the minimal (w.r.t. $\leq_t$) normal (resp. extended) interpretation of $s$, $M = \{M_s^x \mid \langle s', x \rangle \in D_{s,n}^S\}$, such that:

For all $\langle s', x \rangle \in D_{s,n}^S$:

1. For all $p \in \text{Pred}_{s'}^p$, s.t. $x > |\text{mode}_{s'}^p(p)|$, and for all $L \in [p]_S$:
   \[
   M_s^x(L) = M_{s'}^x(L), \text{ where } y = |\text{mode}_{s'}^p(p)|,
   \]

2. For all $p \in \text{Pred}_{s''}^p$, and for all $s'' \in S$ s.t. $\text{Nam}_{s''} \in \text{Import}_{s''}^S(p)$:
   a. for all $L \in [p]_S$:
      \[
      M_s^x(L) \geq M_{s''}^y(L), \text{ where } y = \text{least}(x, \text{mode}_{s''}^y(p), |\text{mode}_{s''}^y(p)|),
      \]
   b. for all $L \in [p@\text{Nam}_{s''}]_S$:
      \[
      M_s^x(L) = M_{s''}(\text{simple}(L)), \text{ where } y = \text{least}(x, \text{mode}_{s''}^y(p), |\text{mode}_{s''}^y(p)|),
      \]

3. $M_s^x \models P_{s'/\Lambda_{SS}}^x N_{s'}^x$ (resp. $M_s^x \models P_{s'/\Lambda_{SS}}^x N_{s'}^x$).

Intuitively, Definition 5.11 expresses that if $L$ is a literal defined in $s'$ at reasoning mode $y$ then the truth value of $L$ according to $M_{s'}^y$, for $x \geq |y|$, is equal to the truth value of $L$ according to $M_{s'}^y$. If $L$ is a simple (resp. qualified) literal imported in $s'$ from a rule base $s''$ then the truth value of $L$ according to $M_{s''}^y$ is greater than or equal (resp. equal) to the truth value of $L$ according to $M_{s'}^y$, where $y = \text{least}(x, \text{mode}_{s''}^y(p), |\text{mode}_{s''}^y(p)|)$ and $p = \text{pred}(L)$. Additionally, it holds that: $M_s^x \models P_{s'/\Lambda_{SS}}^x N_{s'}^x$ (resp. $M_s^x \models P_{s'/\Lambda_{SS}}^x N_{s'}^x$), for $x \in \{d, o, c, n\}$. Thus, for $M_s^x$, we consider the logic program $P_{s'}^x$ which contains the rules that define the definite, open, and (positively or negatively) closed predicates in $s$. For $M_s^x$, we consider the logic program $P_{s''}^x$, which contains the rules that define the open and (positively or negatively) closed predicates $p$ in $s$, along with the OWA rules of $p$. For $M_{s''}$, we consider the logic program $P_{s''}^x$, which contains the rules that define the (positively or negatively) closed predicates $p$ in $s$, along with the corresponding CWA rules of $p$. Finally, for $M_{s''}$, we consider the logic program $P_{s''}^x$, which contains the rules that define the normal predicates in $s$.

The following proposition states that Definition 5.11 is well-defined.

**Proposition 5.12.** Let $N$ be a normal (resp. extended) interpretation of $s$ in reasoning mode $m$ w.r.t. $S$. It always exists the minimal model of $s$ in reasoning mode $m$ w.r.t. $S$ and $N$. □
Below, we adapt the definition of the Coh operator in WFSX [Pereira and Alferes 1992] to our framework.

**Definition 5.13. (Coh operator)** Let \( L = \{ I^D_s \mid \langle s', x \rangle \in D^D_{s, a} \} \) be an extended interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \). We define \( \text{Coh}(l) = \{ \text{Coh}(I^D_s) \mid \langle s', x \rangle \in D^D_{s, a} \} \), where \( \text{Coh}(I^D_s) = I^D_s \cup \{ \neg L \mid L \in \text{HB}^S_s \text{ and } \neg L \in I^D_s \} \). □

Note that if \( l \) is an extended interpretation then all simple extended interpretations in \( \text{Coh}(l) \) are coherent.

**Example 16.** Consider the modular rule base \( S \) of Example 6. Additionally, consider the extended interpretation of \( s_1 \) in reasoning mode \( c \) w.r.t. \( S \), \( l = \{ I^D_{s_1} \}, \) where \( I^D_{s_1} = \{ \neg \text{eu:Country(Canada)}, \sim \text{eu:Country(Egypt)} \} \) and \( I^D_{s_2} = \{ \text{geo:Country(Egypt)} \} \). It holds that \( \text{Coh}(l) = \{ \text{Coh}(I^D_{s_1}), \text{Coh}(I^D_{s_2}) \} \), where \( \text{Coh}(I^D_{s_1}) = \{ \neg \text{eu:Country(Canada)}, \sim \text{eu:Country(Egypt)} \} \) and \( \text{Coh}(I^D_{s_2}) = \{ \text{geo:Country(Egypt)} \} \). □

We are now ready to define the normal and extended answer sets of a rule base in a reasoning mode \( m \) w.r.t. a modular rule base.

**Definition 5.14. (Normal & extended answer set of a rule base in a reasoning mode w.r.t. a MRB)** Let \( M \) be a normal (resp. extended) interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \). \( M \) is a normal (resp. extended) answer set of \( s \) in reasoning mode \( m \) w.r.t. \( S \), if \( M = \text{least}^n_s(s, M) \) (resp. \( M = \text{Coh}(\text{least}^e_s(s, M)) \)).

We denote the set of normal answer sets of \( s \) in reasoning mode \( m \) w.r.t. \( S \) by \( \mathcal{M}^n_{s, S}(s) \) and the set of extended answer sets of \( s \) in reasoning mode \( m \) w.r.t. \( S \) by \( \mathcal{M}^e_{s, S}(s) \). □

**Example 17.** Consider the modular rule base \( S \) of Example 6.

Let \( L = \neg \text{eu:CountryEU(Croatia)} \).

For all \( M \in \mathcal{M}^n_{s_1, S}(s_1) \), it holds that: \( M^o_{s_1}(L) = 1 \). Note that \( |\mathcal{M}^n_{s_1, S}(s_1)| = 1 \).

For all \( M \in \mathcal{M}^e_{s_1, S}(s_1) \), it holds that: \( M^o_{s_1}(L) \in \{0, 1\} \). Additionally, for all \( M \in \mathcal{M}^e_{s_1, S}(s_1) \), it holds that: \( M^o_{s_2}(L) \in \{0, 1\} \) and \( M^e_{s_2}(L) = M^o_{s_2}(L) \in \{0, 1\} \). Note that rule base \( s_4 \) requests predicate \( \text{eu:CountryEU} \) from rule base \( s_1 \) in open reasoning mode. Furthermore, it holds that:

\[-M^o_{s_1}(\text{gov:Enter(Anne)}) = 1. \text{ This is because:} \]

(i) \( M^o_{s_4}(\text{eu:CountryEU(Austria)}) = 1 \) and

(ii) \( M^e_{s_4}(\sim \text{sec:Suspect(Anne)}@http://security.int) = 1 \).

\[-M^o_{s_1}(\text{gov:Enter(Boris)}) = 1. \text{ This is because:} \]

(i) \( M^o_{s_4}(\text{eu:CountryEU(Croatia)}) = 1 \) or \( M^e_{s_4}(\sim \text{eu:CountryEU(Croatia)}) = 1 \),

(ii) \( M^o_{s_4}(\sim \text{gov:RequiresVisa(Croatia)}) = 1 \), and

(iii) \( M^e_{s_4}(\sim \text{sec:Suspect(Boris)}@http://security.int) = 1 \).

\[-M^o_{s_1}(\text{gov:Enter(Peter)}) = 0. \text{ This is because:} \]

\( M^e_{s_4}(\sim \text{sec:Suspect(Peter)}@http://security.int) = 0 \).

Note that, there exists \( M \in \mathcal{M}^e_{s_2, S}(s_4) \) such that: \( M^o_{s_1}(L) = M^o_{s_4}(L) = M^e_{s_4}(L) = 0 \) (resp. \( M^o_{s_1}(L) = M^o_{s_4}(L) = M^e_{s_4}(L) = 1 \)).
For all $M \in \mathcal{M}_{EAS}^E(s_1)$, it holds that: $M_s^c(L) = 1$. Note that $|\mathcal{M}_{EAS}^E(s_1)| = 1$.

For all $M \in \mathcal{M}_{EAS}^E(s_1)$, it holds that: $M_s^c(L) \in \{0, 1, 2, 1\}$. Additionally, for all $M \in \mathcal{M}_{EAS}^E(s_1)$, it holds that: $M_s^c(L) \in \{0, 1, 2, 1\}$ and $M_s^c(L) = M_s^c(L) \in \{0, 1, 2, 1\}$. Furthermore, it holds that: $M_s^c(gov: Enter(Anne)) = 1$, $M_s^c(gov: Enter(Boris)) = 0$. Note that for $M \in \mathcal{M}_{EAS}^E(s_1)$ s.t. $M_s^c(eu: Country EU(Croatia)) = 1/2$, it holds that $M_s^c(gov: Enter(Boris)) = 1/2$.

Note that:

1. There exists $M \in \mathcal{M}_{EAS}^E(s_1)$ such that: $M_s^c(\neg L) = M_s^c(\neg L) = M_s^c(\neg L) = 1$, and $M_s^c(gov: Enter(Boris)) = 1$.

2. There exists $M \in \mathcal{M}_{EAS}^E(s_1)$ such that: $M_s^c(L) = M_s^c(L) = M_s^c(L) = 1/2$, and $M_s^c(gov: Enter(Boris)) = 1/2$.

3. There exists $M \in \mathcal{M}_{EAS}^E(s_1)$ such that: $M_s^c(L) = M_s^c(L) = M_s^c(L) = M_s^c(L) = 1$, and $M_s^c(gov: Enter(Boris)) = 1$. □

The following proposition relates the normal (resp. extended) answer sets of different rule bases and reasoning modes.

**Proposition 5.15.** Let $M$ be a consistent normal or extended interpretation of rule base $s$ in reasoning mode $m$ w.r.t. $S$. Let $s' \in S$ and let $x \in \{d, o, c, n\}$ s.t. $\langle s', x \rangle \in D_{s,x}^S$. Additionally, let $M' = \{M'_s \in M \mid \langle s'', y \rangle \in D_{s,x}^S\}$. It holds that:

1. If $M \in \mathcal{M}_{EAS}^E(s)$ then $M' \in \mathcal{M}_{EAS}^E(s')$.
2. If $M \notin \mathcal{M}_{EAS}^E(s)$ then $M' \notin \mathcal{M}_{EAS}^E(s')$. □

The following proposition states that if there exists an inconsistent normal answer set of $s$ in reasoning mode $m$ w.r.t. $S$ then this is the only normal answer set of $s$ in reasoning mode $m$ w.r.t. $S$. Additionally, $\mathcal{M}_{EAS}^E(s)$ is either empty, or it consists of the inconsistent extended answer set of $s$ in reasoning mode $m$ w.r.t. $S$.

**Proposition 5.16.** If there exists $M \in \mathcal{M}_{EAS}^E(s)$ s.t. $M$ is inconsistent then:

1. $\mathcal{M}_{EAS}^E(s) = \{M\}$, and
2. $\mathcal{M}_{EAS}^E(s) = \{M\}$, where $M'$ is inconsistent, or $\mathcal{M}_{EAS}^E(s) = \{\}$. □

**Example 18.** Consider the modular rule base $S$ of Example 6 and assume that we add to rule base $s_1$ the fact $\neg eu: Country EU(Austria)$. Then, for all $m \in \{d, o, c, n\}$, $\mathcal{M}_{EAS}^E(s_1) = \{M\}$, where $M$ is inconsistent and $\mathcal{M}_{EAS}^E(s_1) = \{M\}$, where $M'$ is inconsistent. Assume now that we replace the two rules with head $gov: Enter(\ ?p)$ of rule base $s_4$ with the rules:

```
```

Then, $\mathcal{M}_{EAS}^E(s_4) = \{M\}$, where $M$ is inconsistent, whereas $\mathcal{M}_{EAS}^E(s_4) = \{\}$. This is because $P_{s_4}^{\neg HB^S} \models \neg sec: citizenOf(Peter, Croatia)$, sec: citizenOf(Croatia, Peter},

\(^a\)Recall that there is only one inconsistent normal (resp. extended) interpretation of $s$ in reasoning mode $m$ w.r.t. $S$.\(^b\)
Thus, $\mathcal{M}' \neq \text{Coh}(\text{least}_2^C(s_4, \mathcal{M}'))$, where $\mathcal{M}'$ is the inconsistent extended interpretation of $s_4$ in reasoning mode $n$ w.r.t. $\mathcal{S}$. □

Based on Proposition 5.16, we define a contradictory rule base in a reasoning mode w.r.t. a modular rule base.

**Definition 5.17. (Contradictory rule base in a reasoning mode w.r.t. a MRB)** If $\mathcal{M}_{es, s}^e = \{\mathcal{M}\}$ s.t. $\mathcal{M}$ is inconsistent then rule base $s$ is called contradictory in reasoning mode $m$ w.r.t. $\mathcal{S}$. □

The following proposition states that if there exists an inconsistent extended answer set of $s$ in reasoning mode $m$ w.r.t. $\mathcal{S}$ then this is the only extended answer set of $s$ in reasoning mode $m$ w.r.t. $\mathcal{S}$ and $s$ is contradictory in reasoning mode $m$ w.r.t. $\mathcal{S}$.

**Proposition 5.18.** If there exists $\mathcal{M} \in \mathcal{M}_{es, s}^e(s)$ s.t. $\mathcal{M}$ is inconsistent then (i) rule base $s$ is contradictory in reasoning mode $m$ w.r.t. $\mathcal{S}$, and (ii) $\mathcal{M}_{es, s}^e(s) = \{\mathcal{M}\}$. □

The following proposition shows that inconsistency, local to rule base $s$ in reasoning mode $m$ w.r.t. $\mathcal{S}$, propagates to (i) all reasoning modes of $s$, if $m \in \{d, o, c\}$, and (ii) to rule bases $s' \in \mathcal{S}$ and reasoning modes $x \in \{d, o, c, n\}$ s.t. $\langle s, m \rangle \in D_{s', x}^\mathcal{S}$. All other cases remain unaffected from the local inconsistency.

**Proposition 5.19.** Assume that rule base $s$ in reasoning mode $m$ w.r.t. $\mathcal{S}$ is contradictory. It holds that:

1. If $m \in \{d, o, c\}$ then rule base $s$ in reasoning mode $x \in \{d, o, c, n\}$ w.r.t. $\mathcal{S}$ is also contradictory.

2. If $s' \in \mathcal{S}$ and $x \in \{d, o, c, n\}$ s.t. $\langle s, m \rangle \in D_{s', x}^\mathcal{S}$ then rule base $s'$ in reasoning mode $x$ w.r.t. $\mathcal{S}$ is contradictory. □

**Example 19.** Consider the modular rule base $\mathcal{S}$ of Example 6 and assume that we add to rule base $s_1$ the fact $\neg \text{eu:CountryEU(Austria)}$. Then, for all $x \in \{d, o, c, n\}$, rule bases $s_1$ and $s_4$ in reasoning mode $x$ are contradictory, while rule bases $s_2$ and $s_3$ in reasoning mode $x$ are not. This is because rule base $s_4$ imports predicate $\text{eu:CountryEU}$ from rule base $s_1$, while rule bases $s_2$ and $s_3$ do not.

Consider now the modular rule base $\mathcal{S}$ of Example 6 and assume that we add to rule base $s_4$ the facts $\text{gov:Enter(Anne)}$ and $\neg \text{gov:Enter(Anne)}$. It holds that rule base $s_4$ in reasoning mode $n$ is contradictory. However, rule base $s_4$ in reasoning modes $d, o, c$ is non-contradictory. This is because contradictory information is isolated to normal reasoning mode of rule base $s_4$, due to Definition 5.5. Similarly, for all $x \in \{d, o, c, n\}$, rule bases $s_1$, $s_2$, and $s_3$ in reasoning mode $x$ are non-contradictory. □

Let $\mathcal{S}$ be a modular rule base, let $s, s' \in \mathcal{S}$, and let $m, x \in \{d, o, c, n\}$. It is possible that there is no normal (resp. extended) answer set of rule base $s$ in reasoning mode $m$, even though there is a normal (resp. extended) answer set of rule base $s'$ in reasoning mode $x$. This is demonstrated in the following example:

**Example 20.** Consider the modular rule base $\mathcal{S}$ of Example 6 and assume that we add to rule base $s_4$ the fact $\neg \text{gov:Enter(Anne)}$. It holds that there is no normal
or extended answer set of rule base \( s_4 \) in reasoning mode \( m \) w.r.t. \( S \). This is because, as we show in Example 17, \( \text{gov:Enter(Anne)} \) is also derived from the rules. However, there is a normal and an extended answer set of rule base \( s_4 \) in reasoning modes \( d,o,c \) w.r.t. \( S \). This is because, due to Definition 5.5, the semantics of rule base \( s_4 \) in reasoning modes \( d,o,c \) are defined independently to reasoning mode \( n \). Similarly, for all \( x \in \{d,o,c,n\} \), there is a normal and an extended answer set of rule bases \( s_1, s_2, \) and \( s_3 \) in reasoning mode \( x \) w.r.t. \( S \). This is because rule bases \( s_1, s_2, \) and \( s_3 \) do not import predicate \( \text{gov:Enter} \) from rule base \( s_4 \).

The following proposition expresses that if \( \mathcal{M}_{m,S}^{EAS}(s) \neq \emptyset \) then there is a unique minimal w.r.t. \( \leq_k \) (thus, least) extended answer set of \( s \) in reasoning mode \( m \) w.r.t. \( S \).

**Proposition 5.20.** It holds that: \( |\text{minimal}_{\leq_k}(\mathcal{M}_{m,S}^{EAS}(s))| \leq 1. \)

Based on Proposition 5.20, we define the modular well-founded model of \( s \) in reasoning mode \( m \) w.r.t. \( S \), as follows:

**Definition 5.21.** (Well-founded model of a rule base in a reasoning mode w.r.t. a modular rule base) We define the modular well-founded model of \( s \) in reasoning mode \( m \) w.r.t. \( S \), \( \mathcal{M}_{m,S}^{WFS} \), as follows:

1. \( \mathcal{M}_{m,S}^{WFS} = \text{least}_{\leq_k}(\mathcal{M}_{m,S}^{EAS}(s)) \), if \( \mathcal{M}_{m,S}^{EAS}(s) \neq \emptyset \), and
2. \( \mathcal{M}_{m,S}^{WFS} \) is the inconsistent extended interpretation of \( s \) in reasoning mode \( m \) w.r.t. \( S \), otherwise.

Below, we define \( \mathcal{M}_{WFS}^{AS} \) and \( \mathcal{M}_{WFS}^{WFS} \) entailment over a rule base \( s \) w.r.t. a modular rule base \( S \).

**Definition 5.22.** (\( \mathcal{M}_{WFS}^{AS} \) & \( \mathcal{M}_{WFS}^{WFS} \) entailment) Let \( S \) be a modular rule base and let \( s \in S \). Let:

1. \( p \in \text{Pred}_m \), if \( M^\#_m(\text{Pred}_m(p)) \), and \( L \in [p]_S \cup \sim[p]_S \), or
2. \( p \in \text{Pred}_m \), if \( M^\#_m(\text{Pred}_m(p)) \), and \( L \in [p]_S \cup \sim[p]_S \), or
3. \( p \in \text{Pred}_m, M^\#_m(\text{Pred}_m(p), \text{Nam}_L \in \text{Import}_m^\#(p), \text{and } L \in [p \land \text{Nam}_L]_S \cup \sim[p \land \text{Nam}_L]_S \).

We say that:

\[ s \text{ entails } L \text{ w.r.t. } S \text{ under } \mathcal{M}_{WFS}^{AS} \text{ (} s \models^{\mathcal{M}_{WFS}^{AS}} L \text{) iff for all } M \in \mathcal{M}_{m,S}^{EAS}(s), M^\#_m(L) = 1. \]

\[ s \text{ entails } L \text{ w.r.t. } S \text{ under } \mathcal{M}_{WFS}^{WFS} \text{ (} s \models^{\mathcal{M}_{WFS}^{WFS}} L \text{) iff for all } M \in \mathcal{M}_{m,S}^{EAS}(s), M^\#_m(L) = 1. \]

**Example 21.** Consider the modular rule base \( S \) of Example 6. For \( \text{SEM} \in \{mAS,mWFS\} \), it holds \( s_1 \models^\text{SEM}_{S} \sim \text{eu:CountryEU(Croatia)} \), while \( s_4 \not\models^\text{SEM}_{S} \sim \text{eu:CountryEU(Croatia)} \). This is because, while rule base \( s_1 \) declares \( \text{eu:CountryEU(} \) in positively closed reasoning mode, rule base \( s_4 \) imports \( \text{eu:CountryEU from } s_1 \) in open reasoning mode. Additionally, for \( \text{SEM} \in \{mAS,mWFS\} \), it holds \( s_4 \models^\text{SEM}_{S} \text{gov:Enter(Anne)} \), and \( s_4 \models^\text{SEM}_{S} \sim \text{gov:Enter(Peter)} \). Moreover, it holds \( s_4 \models^\text{SEM}_{S} \text{gov:Enter(Boris)} \), while \( s_4 \not\models^\text{SEM}_{S} \text{gov:Enter(Boris)} \). This is because, \( \mathcal{M}_{WFS}^{AS} \), in contrast to \( \mathcal{M}_{WFS}^{WFS} \), supports case-based analysis on the truth values of \( \text{eu:CountryEU(Croatia)} \) and \( \sim \text{eu:CountryEU(Croatia)} \) in rule base \( s_4 \).

The following corollary follows directly from Definition 5.21 and Proposition 5.20.
Corollary 5.23. Let $S$ be a modular rule base and let $s \in S$. Let:

1. $p \in \text{Pred}_s^p$, $\mathbf{m} = |\text{mode}_s^p(p)|$, and $L \in [p]_S \cup \neg[p]_S$, or
2. $p \in \text{Pred}_s^p - \text{Pred}_s^p$, $\mathbf{m} = \text{mode}_s(p)$, and $L \in [p]_S \cup \neg[p]_S$, or
3. $p \in \text{Pred}_s^p$, $\mathbf{m} = \text{mode}_s^p(p)$, and $L \in [p]_S \cup \neg[p]_S$.

It holds that: $s \models^\text{WFS}_S L$ if $M_p(L) = 1$, where $M = \text{mWFS}_S$.

6. Transformational Semantics for Modular Rule Bases

In this section, we show how a modular rule base $S$ can be transformed into a set of extended logic programs (ELPs) whose conclusions capture the Web model-theoretic semantics of $S$. In particular, for each $s \in S$, four extended logic programs (ELPs) are generated, one for each reasoning mode: definite, open, closed, and normal. We denote these ELPs by $\Pi^d_s$, $\Pi^o_s$, $\Pi^c_s$, and $\Pi^n_s$, respectively. We assume that the legality of $S$ has already been verified, by checking that: (i) each $s \in S$ is a legal rule base (see Definition 4.6) and (ii) $S$ is a legal modular rule base (see Definition 4.8).

A major advantage of our transformational approach is that extending the logic programs of a modular rule base $S$ or adding new rule bases does not require changing the form of the ELP rules that have already been generated.

Let $s \in S$. To proceed, we need to define a total and injective function, $\tau^s_x(\cdot)$, for $x \in \{d, o, c, n\}$, from $\text{HB}^d_s$ to the set of ELP literals, appearing in the generated ELPs $\Pi^d_s$, $\Pi^o_s$, $\Pi^c_s$, and $\Pi^n_s$. Specifically, let $A = p(t)$ be a simple atom and let $x \in \{d, o, c, n\}$. We define: (a) $\tau^s_x(A) = \text{Nam}_x \cdot x \cdot p(t)$ and (b) $\tau^s_x(\neg A) = \neg \text{Nam}_x \cdot x \cdot p(t)$. Additionally, let $A = p(t) \circ \text{Nam}_t$ be a qualified atom and let $x \in \{d, o, c, n\}$. We define: (a) $\tau^s_x(A) = \text{Nam}_x \cdot x \cdot p \circ \text{Nam}_t(t)$ and (b) $\tau^s_x(\neg A) = \neg \text{Nam}_x \cdot x \cdot p \circ \text{Nam}_t(t)$. For an atom $A \in \text{HB}^d_s$, we define: $\tau^s_x(\neg A) = \neg \tau^s_x(A)$.

In the generated ELPs, the symbols $\text{Nam}_x \cdot x \cdot p$ and $\text{Nam}_x \cdot x \cdot p \circ \text{Nam}_t$, for $s, t \in S$, $p \in \text{Pred}_s$, and $x \in \{d, o, c, n\}$, correspond to predicate names. Intuitively, the IRI of the owner rule base, $\text{Nam}_s$, and the IRI of the qualification rule base, $\text{Nam}_t$, control the evaluation scope of the web predicate $p$, while $x$ indicates the reasoning mode.

Our transformational approach proceeds as follows: First, for all $s \in S$, the auxiliary ELPs $\Pi^d_s$, $\Pi^o_s$, $\Pi^c_s$, and $\Pi^n_s$ are generated, which initially contain the facts “domain(c),” where $c \in \text{HB}_S$.

Let $s \in S$ with $\text{Nam}_s = n_s$. The defines declarations of $s$ are translated as follows:

1. Let $p$ be a predicate that is defined as definite in $s$ (that is, $\text{mode}_s^p(p) = d$).
   
   If $P_s$ contains a rule $r$:
   
   $$L_0 \leftarrow L_1, \ldots, L_m, \text{ where } \text{pred}(L_0) = p,$$

   then $r$ is translated into the following rule, denoted by $r^d$:
   
   $$\tau^s_x(L_0) \leftarrow \tau^s_x(L_1), \ldots, \tau^s_x(L_m).$$

   Rule $r^d$ is added to ELP $P^n_s$, and corresponds to the case where the reasoning mode of $s$ is definite. Additionally, the following rules are generated:

---

10In order to avoid name clashes, it is assumed that IRIs always appear between delimiters “⟨” and “⟩.”

27
Let $p \in Pred_s$ be a predicate that is defined as open w.r.t. a predicate $cxt$ in $s$ (that is, $mode_s^o(p) = o$ and $context_s(p) = cxt$). If $P_s$ contains a rule $r$:

$$L_0 \leftarrow L_1, \ldots, L_m, \text{ where } pred(L_0) = p,$$

then $r$ is translated into the following rules, denoted by $r^d$ and $r^o$, respectively:

$$\tau^d_s(L_0) \leftarrow \tau^d_s(L_1), \ldots, \tau^d_s(L_m).$$

$$\tau^o_s(L_0) \leftarrow \tau^o_s(L_1), \ldots, \tau^o_s(L_m).$$

Rule $r^d$ is added to ELP $P^{d}_{s, S}$. Additionally, the following rules are generated:

$$\neg n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$\neg n_s : c_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$n_s : c_p(\mathbf{P}) \leftarrow n_s : c_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$\neg n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

$$n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

where $\mathbf{P} = (?x_1, \ldots, ?x_k)$ and $k = arity(p)$.

The rules in the first line of the above set of rules are added to ELP $P^{o}_{s, S}$ and correspond to the case where the reasoning mode of $s$ is open. The rules in the second line are added to ELP $P^{c}_{s, S}$ and correspond to the case where the reasoning mode of $s$ is closed. The rules in the third line are added to ELP $P^{n}_{s, S}$ and correspond to the case where the reasoning mode of $s$ is normal.

(2) Let $p \in Pred_s$ be a predicate that is defined as open w.r.t. a predicate $cxt$ in $s$ (that is, $mode_s^o(p) = o$ and $context_s(p) = cxt$). If $P_s$ contains a rule $r$:

$$L_0 \leftarrow L_1, \ldots, L_m, \text{ where } pred(L_0) = p,$$

then $r$ is translated into the following rules, denoted by $r^d$ and $r^o$, respectively:

$$\tau^d_s(L_0) \leftarrow \tau^d_s(L_1), \ldots, \tau^d_s(L_m).$$

$$\tau^o_s(L_0) \leftarrow \tau^o_s(L_1), \ldots, \tau^o_s(L_m).$$

Rule $r^d$ is added to ELP $P^{d}_{s, S}$. Additionally, the following rules are generated:

$$\neg n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$\neg n_s : c_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$n_s : c_p(\mathbf{P}) \leftarrow n_s : c_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$\neg n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

$$n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

where $\mathbf{P} = (?x_1, \ldots, ?x_k)$ and $k = arity(p)$.

Rule $r^o$ and the rules in the first line of the above set of rules are added to ELP $P^{o}_{s, S}$. The rules in the second line are added to ELP $P^{c}_{s, S}$. The rules in the third line are added to ELP $P^{n}_{s, S}$.

(3) Let $p \in Pred_s$ be a predicate that is defined as freely open in $s$ (that is, $mode_s^o(p) = o$ and $context_s(p) = n/a$). This case is treated similarly to the previous case with the difference that literals $n_s : o_{cxt}(\mathbf{P})$ are eliminated from the generated rules.

(4) Let $p \in Pred_s$ be a predicate that is defined as positively closed w.r.t. a predicate $cxt$ in $s$ (that is, $mode_s^p(p) = c^+$ and $context_s(p) = cxt$). If $P_s$ contains a rule $r$:

$$L_0 \leftarrow L_1, \ldots, L_m, \text{ where } pred(L_0) = p,$$

then $r$ is translated into the following rules, denoted by $r^d$, $r^o$, and $r^c$, respectively:

$$\tau^d_s(L_0) \leftarrow \tau^d_s(L_1), \ldots, \tau^d_s(L_m).$$

$$\tau^o_s(L_0) \leftarrow \tau^o_s(L_1), \ldots, \tau^o_s(L_m).$$

$$\tau^c_s(L_0) \leftarrow \tau^c_s(L_1), \ldots, \tau^c_s(L_m).$$

Rule $r^d$ is added to ELP $P^{d}_{s, S}$. Additionally, the following rules are generated:

$$\neg n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$n_s : o_p(\mathbf{P}) \leftarrow n_s : o_{cxt}(\mathbf{P}), \neg n_s : o_p(\mathbf{P}).$$

$$\neg n_s : c_p(\mathbf{P}) \leftarrow n_s : c_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$n_s : c_p(\mathbf{P}) \leftarrow n_s : c_{cxt}(\mathbf{P}), \neg n_s : c_p(\mathbf{P}).$$

$$\neg n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

$$n_s : n_p(\mathbf{P}) \leftarrow n_s : c_p(\mathbf{P}).$$

28
where \( \bar{x} = \langle ?x_1, \ldots, ?x_k \rangle \) and \( k = \text{arity}(p) \).

Rule \( r^a \) and the rules in the first line of the above set of rules are added to ELP \( P_s^a \). Rule \( r^c \) and the rule in the second line are added to ELP \( P_s^c \). The rules in the third line are added to ELP \( P_s^m \).

(5) Let \( p \in \text{Pred}_s^a \) be a predicate that is defined as freely positively closed in \( s \) (that is, \( \text{mode}_s^a(p) = c^+ \) and \( \text{context}_s(p) = n/a \)). This case is treated similarly to the previous case with the difference that literals \( n_s: \circ \text{ext}(\bar{x}) \) and \( n_s: \circ \text{ext}(\bar{x}) \) are eliminated from the generated rules.

(6) Let \( p \in \text{Pred}_s^a \) be a predicate that is defined as negatively closed w.r.t. a predicate \( \text{ctx} \) in \( s \) (that is, \( \text{mode}_s^a(p) = c^- \) and \( \text{context}_s(p) = \text{ctx} \)). This case is similar to Case 4. Specifically, if \( P_s \) contains a rule \( r \):

\[
L_0 \leftarrow L_1, \ldots, L_m, \text{ where } \text{pred}(L_0) = p,
\]

then \( r \) is translated into the following rules, denoted by \( r^a \), \( r^c \), and \( r^e \), respectively:

\[
\begin{align*}
\tau^a_s(L_0) & \rightarrow \tau^a_s(L_1), \ldots, \tau^a_s(L_m). \\
\tau^c_s(L_0) & \rightarrow \tau^c_s(L_1), \ldots, \tau^c_s(L_m). \\
\tau^e_s(L_0) & \rightarrow \tau^e_s(L_1), \ldots, \tau^e_s(L_m).
\end{align*}
\]

Rule \( r^a \) is added to ELP \( P_s^a \). Additionally, the following rules are generated:

\[
\begin{align*}
-n_s: \circ \text{p}(\bar{x}) & \leftarrow n_s: \circ \text{ext}(\bar{x}), \sim n_s: \circ \text{p}(\bar{x}). \\
n_s: \circ \text{p}(\bar{x}) & \leftarrow n_s: \circ \text{ext}(\bar{x}), \sim n_s: \circ \text{p}(\bar{x}). \\
n_s: \circ \text{p}(\bar{x}) & \leftarrow n_s: \circ \text{ext}(\bar{x}), \sim n_s: \circ \text{p}(\bar{x}).
\end{align*}
\]

where \( \bar{x} = \langle ?x_1, \ldots, ?x_k \rangle \) and \( k = \text{arity}(p) \).

Rule \( r^a \) and the rules in the first line of the above set of rules are added to ELP \( P_s^a \). Rule \( r^c \) and the rules in the second line are added to ELP \( P_s^c \). Finally, the rules in the third line are added to \( P_s^m \).

(7) Let \( p \in \text{Pred}_s^a \) be a predicate that is defined as freely negatively closed in \( s \) (that is, \( \text{mode}_s^a(p) = c^- \) and \( \text{context}_s(p) = n/a \)). This case is treated similarly to the previous case with the difference that literals \( n_s: \circ \text{ext}(\bar{x}) \) and \( n_s: \circ \text{ext}(\bar{x}) \) are eliminated from the generated rules.

(8) Let \( p \in \text{Pred}_s^a \) be a predicate that is defined as normal in \( s \) (that is, \( \text{mode}_s^a(p) = n \)). If \( P_s \) contains a rule \( r \):

\[
L_0 \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n, \text{ where } \text{pred}(L_0) = p,
\]

then \( r \) is translated into the following rule, denoted by \( r^a \):

\[
\tau^a_s(L_0) \rightarrow \tau^a_s(L_1), \ldots, \tau^a_s(L_m), \sim \tau^a_s(L_{m+1}), \ldots, \sim \tau^a_s(L_n).
\]

Rule \( r^a \) is added to ELP \( P_s^a \).

The uses declarations of rule base \( s \) generate rules that respect Definition 5.11(2) (see also Table I). In particular:
(1) Let \( p \in \text{Pred}_1^d \) be a predicate that is requested from a rule base \( t \in \mathcal{S} \) (i.e. \( \text{Nam}_t \in \text{Import}_1^d(p) \)) s.t. \( \text{least}(\text{mode}_1^d(p), |\text{mode}_1^d(p)|) = d \) (see elements definite of Table I). Then, the following rules are generated:

\[
\begin{align*}
&n_1 \cdot d_p(t) \leftarrow n_1 \cdot d_p(t). \\
&n_2 \cdot o_p(t) \leftarrow n_2 \cdot d_p(t). \\
&n_3 \cdot c_p(t) \leftarrow n_3 \cdot d_p(t). \\
&n_4 \cdot n_p(t) \leftarrow n_4 \cdot d_p(t).
\end{align*}
\]

where \( \overline{t} = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

Additionally, the following rules are generated:

\[
\begin{align*}
&n_1 \cdot d_p \# n_1(t) \leftarrow n_1 \cdot d_p(t). \\
&n_2 \cdot o_p \# n_1(t) \leftarrow n_2 \cdot d_p(t). \\
&n_3 \cdot c_p \# n_1(t) \leftarrow n_3 \cdot d_p(t). \\
&n_4 \cdot n_p \# n_1(t) \leftarrow n_4 \cdot d_p(t).
\end{align*}
\]

where \( \overline{t} = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

The rules in the first line of the above two sets of rules are added to ELP \( \Pi^d_{s,S} \).

The rules in the second line are added to ELP \( \Pi^o_{s,S} \). The rules in the third line are added to ELP \( \Pi^d_{s,S} \), and the rules in the fourth line are added to ELP \( \Pi^o_{s,S} \).

(2) Let \( p \in \text{Pred}_1^d \) be a predicate that is requested from a rule base \( t \in \mathcal{S} \) (i.e. \( \text{Nam}_t \in \text{Import}_1^o(p) \)) s.t. \( \text{least}(\text{mode}_1^o(p), |\text{mode}_1^o(p)|) = o \) (see elements open of Table I). Then, the following rules are generated:

\[
\begin{align*}
&n_1 \cdot d_p(t) \leftarrow n_1 \cdot d_p(t). \\
&n_2 \cdot o_p(t) \leftarrow n_2 \cdot o_p(t). \\
&n_3 \cdot c_p(t) \leftarrow n_3 \cdot o_p(t). \\
&n_4 \cdot n_p(t) \leftarrow n_4 \cdot o_p(t).
\end{align*}
\]

where \( \overline{t} = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

Additionally, the following rules are generated:

\[
\begin{align*}
&n_1 \cdot d_p \# n_1(t) \leftarrow n_1 \cdot d_p(t). \\
&n_2 \cdot o_p \# n_1(t) \leftarrow n_2 \cdot o_p(t). \\
&n_3 \cdot c_p \# n_1(t) \leftarrow n_3 \cdot o_p(t). \\
&n_4 \cdot n_p \# n_1(t) \leftarrow n_4 \cdot o_p(t).
\end{align*}
\]

where \( \overline{t} = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

The rules in the first line of the above two sets of rules are added to ELP \( \Pi^d_{s,S} \).

The rules in the second line are added to ELP \( \Pi^o_{s,S} \). The rules in the third line are added to ELP \( \Pi^d_{s,S} \), and the rules in the fourth line are added to ELP \( \Pi^o_{s,S} \).

(3) Let \( p \in \text{Pred}_1^d \) be a predicate that is requested from a rule base \( t \in \mathcal{S} \) (i.e. \( \text{Nam}_t \in \text{Import}_1^c(p) \)) s.t. \( \text{least}(\text{mode}_1^c(p), |\text{mode}_1^c(p)|) = c \) (see elements closed
of Table I). Then, the following rules are generated:

\[
\begin{align*}
n_1 \cdot d_p(\tau) & \leftarrow n_1 \cdot d_p(\tau), \quad \neg n_1 \cdot d_p(\tau) \leftarrow \neg n_1 \cdot d_p(\tau), \\
n_1 \cdot o_p(\tau) & \leftarrow n_1 \cdot o_p(\tau), \quad \neg n_1 \cdot o_p(\tau) \leftarrow \neg n_1 \cdot o_p(\tau), \\
n_1 \cdot c_p(\tau) & \leftarrow n_1 \cdot c_p(\tau), \quad \neg n_1 \cdot c_p(\tau) \leftarrow \neg n_1 \cdot c_p(\tau), \\
n_1 \cdot n_p(\tau) & \leftarrow n_1 \cdot n_p(\tau), \quad \neg n_1 \cdot n_p(\tau) \leftarrow \neg n_1 \cdot c_p(\tau),
\end{align*}
\]

where \( \tau = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

Additionally, the following rules are generated:

\[
\begin{align*}
n_1 \cdot d_p @ n_1(\tau) & \leftarrow n_1 \cdot d_p(\tau), \quad \neg n_1 \cdot d_p @ n_1(\tau) \leftarrow \neg n_1 \cdot d_p(\tau), \\
n_1 \cdot o_p @ n_1(\tau) & \leftarrow n_1 \cdot o_p(\tau), \quad \neg n_1 \cdot o_p @ n_1(\tau) \leftarrow \neg n_1 \cdot o_p(\tau), \\
n_1 \cdot c_p @ n_1(\tau) & \leftarrow n_1 \cdot c_p(\tau), \quad \neg n_1 \cdot c_p @ n_1(\tau) \leftarrow \neg n_1 \cdot c_p(\tau), \\
n_1 \cdot n_p @ n_1(\tau) & \leftarrow n_1 \cdot n_p(\tau), \quad \neg n_1 \cdot n_p @ n_1(\tau) \leftarrow \neg n_1 \cdot c_p(\tau),
\end{align*}
\]

where \( \tau = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

The rules in the first line of the above two sets of rules are added to ELP \( P^d_{s, S} \). The rules in the second line are added to ELP \( P^c_{s, S} \). The rules in the third line are added to ELP \( P^e_{s, S} \), and the rules in the fourth line are added to ELP \( P^n_{s, S} \).

(4) Let \( p \in \text{Pred}^S \) be a predicate that is requested from a rule base \( t \in S \) (i.e. \( \text{Nam}_t \in \text{Import}^S_t(p) \) s.t. \( \text{least}(\text{mod}_t^S(p), |\text{mod}_t^S(p)|) = n \) (see elements normal of Table I)). Then, the following rules are generated:

\[
\begin{align*}
n_1 \cdot n_p(\tau) & \leftarrow n_1 \cdot n_p(\tau), \quad \neg n_1 \cdot n_p(\tau) \leftarrow \neg n_1 \cdot n_p(\tau),
\end{align*}
\]

where \( \tau = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

Additionally, the following rules are generated:

\[
\begin{align*}
n_1 \cdot n_p @ n_1(\tau) & \leftarrow n_1 \cdot n_p(\tau), \quad \neg n_1 \cdot n_p @ n_1(\tau) \leftarrow \neg n_1 \cdot n_p(\tau),
\end{align*}
\]

where \( \tau = \langle ?x_1, \ldots, ?x_k \rangle \), \( k = \text{arity}(p) \), and \( n_t = \text{Nam}_t \).

Generated rules are added to ELP \( P^n_{s, S} \).

Let \( s \in S \). We are now ready to define the ELPs \( \Pi^m_{s, S} \), where \( m \in \{d, o, c, n\} \). In particular, we define: \( \Pi^m_{s, S} = \cup (P^m_{s, S} | \langle s', x \rangle \in D^S_{s, a}) \).

Below, we define the MWebAS and MWebWFS semantics of a modular rule base \( S \) through the corresponding semantics of the ELPs \( \Pi^d_{s, S}, \Pi^c_{s, S}, \Pi^e_{s, S}, \Pi^n_{s, S} \), where \( s \in S \). In particular, the following Theorem shows that the correspondence between the normal (resp. extended) answer sets of a rule base \( s \in S \) in reasoning mode \( m \in \{d, o, c, n\} \) and the answer sets (resp. partial stable models) of \( \Pi^m_{s, S} \), according to AS (resp. WFSX) semantics is one-to-one.

First, we provide the following definition: Let \( M = \{ M^m_s | \langle s', x \rangle \in D^S_{s, a} \} \) be a normal (resp. extended) interpretation of a rule base \( s \) in reasoning mode \( m \) w.r.t. \( S \). We will denote by \( N_M \) the 2-valued (resp. 3-valued) interpretation \( 11 \) of \( \Pi^m_{s, S} \) s.t. (i) \( N_M(t^S_{\tau}(L)) = M^m_s(L) \), for all \( \langle s', x \rangle \in D^S_{s, a} \) and \( L \in \text{HB}^S_s \cup \text{~HB}^S_s \), (ii)

\[11\text{Here, we include also inconsistent interpretations } I \text{ of an ELP } P \text{ s.t. } I(L) = 1, \text{ for all } L \in \text{HB}_P \cup \text{~HB}_P, \text{ where } \text{HB}_P \text{ denotes the Herbrand Base of } P.\]
It holds that: it is an answer set (resp. partial stable model) of Π.

**Theorem 6.1.** Let $S$ be a modular rule base, let $s \in S$, and let $m \in \{d, o, c, n\}$.

It holds that: $M$ is a normal (resp. extended) answer set of $s$ in reasoning mode $m$ w.r.t. $S$ iff $M$ is an answer set (resp. partial stable model$^{12}$) of $\Pi^{m}_{s, S}$.

**Proposition 6.2.** Let $S$ be a modular rule base, let $s \in S$, and let $m \in \{d, o, c, n\}$ s.t. $M^{d}_{s, S}(s) \neq \emptyset$. It holds that: $M = m^{WFS}_{s, m}$ iff $N_{M}$ is the well-founded model (according to WFSX semantics) of $\Pi^{m}_{s, S}$. □

The following Corollary, which follows directly from Definition 5.22 and Theorem 6.1, shows that the meaning of a predicate $p$ defined in a rule base $s \in S$ according to $m^{AS}$ (resp. $m^{WFS}$) semantics, is given by the literals $\tau^{p}_{s}(L)$ in $C^{AS}_{s} (\Pi^{m}_{s, S})$ (resp. $C^{WFS}_{s} (\Pi^{m}_{s, S})$), where $m = mode^{p}_{s}(p)$ and $L \in [p]_{S} \cup \sim[p]_{S}$.

**Corollary 6.3.** Let $S$ be a modular rule base and let $s \in S$. Let:

1. $p \in Pred^{s}_{d}, m = mode^{s}_{d}(p)$, and $L \in [p]_{S} \cup \sim[p]_{S}$, or
2. $p \in Pred^{s}_{o} - Pred^{s}_{d}, m = mode^{s}_{o}(p)$, and $L \in [p]_{S} \cup \sim[p]_{S}$, or
3. $p \in Pred^{s}_{c}$, $m = mode^{s}_{c}(p)$, $Nam_{c} \in Import^{s}_{c}(p)$, and $L \in [p@Nam_{c}]_{S} \cup \sim[p@Nam_{c}]_{S}$.

It holds that:

$-s \models_{S} L \text{ if } \tau^{p}_{s}(L) \in C^{AS}_{s} (\Pi^{m}_{s, S})$.

$-s \models_{S} L \text{ if } \tau^{p}_{s}(L) \in C^{WFS}_{s} (\Pi^{m}_{s, S})$. □

**7. PROPERTIES OF THE MWEB ANSWER SET & MWEB WELL-FOUNDED SEMANTICS**

In this section, we present several properties of the proposed semantics. First, we show that, similarly to $AS$ and $WFSX$ on ELPs, $M^{AS}_{s, S}$ is more informative than $M^{WFS}_{s, S}$. However, $M^{WFS}_{s, S}$ has better computational properties. Additionally, we show that $M^{AS}_{s, S}$ and $M^{WFS}_{s, S}$ extends $AS$ and $WFS$ on ELPs.

**Proposition 7.1.** Let $S$ be a modular rule base, let $s \in S$, and let $L \in HB^{S}_{1} \cup \sim HB^{S}_{2}$. It holds that: if $s \models_{S} L$ then $s \models_{S} L$. □

The following proposition provides the data complexities of $M^{AS}_{s, S}$ and $M^{WFS}_{s, S}$ semantics. These complexities are the same as the complexities of the answer set (AS) and well-founded semantics with explicit negation (WFSX) on ELPs, respectively. This result follows from the fact that we can define the $M^{AS}_{s, S}$ and $M^{WFS}_{s, S}$ semantics of a rule base $s$ w.r.t. a modular rule base $S$, through appropriately defined ELPs $\Pi^{m}_{s, S}$, $\Pi^{n}_{s, S}$, $\Pi^{n}_{s, S}$, and $\Pi^{n}_{s, S}$, evaluated under $AS$ and $WFSX$ semantics, respectively (see Section 6).

**Proposition 7.2.** Let $S$ be a modular rule base, let $s \in S$, and let $L \in HB^{S}_{1} \cup \sim HB^{S}_{2}$. It holds that: (i) the problem of establishing if $s \models_{S} L$ is data

---

$^{12}$Here, we consider an extended definition of the partial stable models of WFSX semantics [Pereira and Alferes 1992; Alferes and Pereira 1996], where inconsistent partial stable models are also allowed.
complete for co-NP and program complete for co-NEXPTIME, and (ii) the problem of establishing if \( s \models_{\text{S}} L \) is data complete for P and program complete for EXPTIME.

Let \( S \) be a modular rule base. The following proposition shows that the \texttt{MWebAS} and \texttt{MWebWFS} semantics of the definite predicates of a rule base \( s \in S \) are equivalent.

**Proposition 7.3.** Let \( S \) be a modular rule base. Additionally, let \( p \in \text{Pred}_p^S \) s.t. \( \text{mode}_p^S(p) = d \), and let \( L \in [p]_S \cup \sim[p]_S \). It holds that: \( s \models_{\text{S}} L \) iff \( s \models_{\text{S}} L \).

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 \( C_{\text{SEM}}(P) \) the set of literals entailed from \( P \) under \( \text{SEM} \in \{\text{AS, WFSX}\} \). If \( P \) does not have a consistent answer set then \texttt{AS} semantics adopts an explosive approach by letting \( C_{\text{AS}}(P) = \text{HB}_P \cup \sim\text{HB}_P \), where \( \text{HB}_P \) is the Herbrand Base of \( P \) [13] [Gelfond and Lifschitz 1990]. Similarly, if \( P \) does not have a consistent well-founded model (according to \texttt{WFSX} semantics) then \( C_{\text{WFSX}}(P) = \text{HB}_P \cup \sim\text{HB}_P \) [Pereira and Alferes 1992].

**Proposition 7.4.** Let \( s \) be a rule base s.t. \( \text{Pred}_p^S = \emptyset \) and for all \( p \in \text{Pred}_p^S \), \( \text{mode}_p^S(p) = n \). Let \( S = \{s\} \), let \( p \in \text{Pred}_p^S \), and let \( L \in [p]_S \cup \sim[p]_S \). It holds that:
- (i) \( s \models_{\text{AS}} L \) iff \( L \in C_{\text{AS}}(P_s) \), and
- (ii) \( s \models_{\text{S}} L \) iff \( L \in C_{\text{WFSX}}(P_s) \).

Let \( S, S' \) be modular rule bases. We say that \( S' \) expands \( S \) if \( S \subseteq S' \). The following proposition shows that reasoning on the predicates of a modular rule base \( S \) remains the same after modular rule base expansion, if the set of rule bases from which knowledge about a predicate is imported in any rule base \( s \in S \) stays the same after the expansion of \( S \).

**Proposition 7.5.** Let \( S \) and \( S' \) be modular rule bases s.t. \( S \subseteq S' \) and for all \( s \in S \) and \( p \in \text{Pred}_p^S \), Import\(_s^S\)(p) = Import\(_s^{S'}\)(p). Let \( L \in \text{HB}_S^S \cup \sim\text{HB}_S^S \). It holds that:
- \( s \models_{\text{SEM}} L \) iff \( s \models_{\text{SEM}} L \), for \( \text{SEM} \in \{\text{AS, WFSX}\} \).

### 7.1 Monotonicity for Definite and Open Predicates under Modular Rule Base Extension

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 \( S \) is extended by extending the rule bases in \( S \) and by adding to \( S \) 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, 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 \( S \) is increased.

**Definition 7.6.** (Extending modular rule bases) Let \( S, S' \) be modular rule bases. We say that \( S' \) extends \( S \) (\( S \leq S' \)) iff for all \( s \in S \), there exists \( s' \in S' \):

\[ \text{The Herbrand Base of an ELP } P \text{ is the set of ground literals } p(c_1, ..., c_k) \text{ and } \neg p(c_1, ..., c_k), \]

where \( p \) is a predicate symbol appearing in \( P \), \( k \) is the arity of \( p \), and \( c_1, ..., c_k \) are constants appearing in \( P \).
i. Nam$_s$ = Nam$_{s'}$, $P_s$ $\subseteq$ $P_{s'}$, Pred$^0_s$ $\subseteq$ Pred$^0_{s'}$, Pred$^1_s$ $\subseteq$ Pred$^1_{s'}$.

ii. For all $p \in$ Pred$^2_s$:
   (a) scope$_s(p) \leq$ scope$_{s'}(p)$ and Export$^S_s(p) \subseteq$ Export$^{S'}_{s'}(p)$,
   (b) mode$^2_s(p) \leq$ mode$^2_{s'}(p)$,
   (c) if $[\text{mode}^2_s(p)], [\text{mode}^2_{s'}(p)] \in \{o, c\}$ then context$_s(p) = $ context$_{s'}(p)$, and

iii. For all $p \in$ Pred$^2_{s'}$:
   mode$^2_s(p) \leq$ mode$^2_{s'}(p)$ and Import$^S_s(p) \subseteq$ Import$^{S'}_{s'}(p)$.

The following propositions are used for proving monotonicity of reasoning over the definite and open predicates of a modular rule base $S$, in the case that $S$ is extended. Consider the special case of modular rule base extension, where (i) the logic program, the defined predicates, and the used predicates of a rule base $s \in S$ are extended, (ii) some of the internal predicates of a rule base $s \in S$ are becoming local or global, (iii) some of the local predicates of a rule base $s \in S$ are becoming global, (iv) the exporting rule base list of some defining predicates of a rule base $s \in S$ is extended, and (v) the importing rule base list of some used predicates of a rule base $s \in S$ is also extended. Additionally, new rule bases may be added to $S$. We refer to this kind of modular rule base extension, as content upgrade extension. Proposition 7.7 shows that the objective literals of definite and open predicates, entailed under $\text{MWAS}$ (resp. $\text{MWF}$), increase after the content upgrade extension of $S$ to $S'$.

**Proposition 7.7. (Content upgrade monotonicity)** Let $S$ and $S'$ be modular rule bases such that for all $s \in S$, there exists $s' \in S'$:

1. Nam$_s$ = Nam$_{s'}$, $P_s$ $\subseteq$ $P_{s'}$, Pred$^0_s$ $\subseteq$ Pred$^0_{s'}$, Pred$^1_s$ $\subseteq$ Pred$^1_{s'}$.

2. For all $p \in$ Pred$^2_s$:
   scope$_s(p) \leq$ scope$_{s'}(p)$, mode$^2_s(p) = $ mode$^2_{s'}(p)$, context$_s(p) = $ context$_{s'}(p)$,
   Export$^S_s(p) \subseteq$ Export$^{S'}_{s'}(p)$, and

3. For all $p \in$ Pred$^2_{s'}$:
   mode$^2_s(p) = $ mode$^2_{s'}(p)$ and Import$^S_s(p) \subseteq$ Import$^{S'}_{s'}(p)$.

Let $s \in S$ and $s' \in S'$ s.t. Nam$_s$ = Nam$_{s'}$. Let $p \in$ Pred$^2_s$ s.t. mode$^2_s(p) \in \{d, o\}$ and let $L \in [p]_S$. It holds that: if $s \models^\text{SEM} S L$ then $s' \models^\text{SEM} S L$, for $\text{SEM} \in \{\text{mAS}, \text{mWF}\}$.

Consider now the special case of modular rule base extension, where the defining or requesting reasoning mode of the predicates defined or used in a rule base $s \in S$, respectively, is increasing. Additionally, new rule bases may be added to $S$. We refer to this kind of modular rule base extension, as mode upgrade extension. Proposition 7.8 shows that the objective literals of definite and open predicates, entailed under $\text{MWAS}$ (resp. $\text{MWF}$), increase after the mode upgrade extension of $S$ to $S'$.

**Proposition 7.8. (Mode upgrade monotonicity)** Let $S$ and $S'$ be modular rule bases such that for all $s \in S$, there exists $s' \in S'$:

1. Nam$_s$ = Nam$_{s'}$, $P_s$ = $P_{s'}$, Pred$^0_s$ = Pred$^0_{s'}$, Pred$^1_s$ = Pred$^1_{s'}$ and

2. For all $p \in$ Pred$^2_s$:
   (a) scope$_s(p) = $ scope$_{s'}(p)$ and Export$^S_s(p) = $ Export$^{S'}_{s'}(p)$,
(b) \( \text{mode}^c_s(p) \leq |\text{mode}^c_s(p)| \),
(d) if \(|\text{mode}^c_s(p)|, |\text{mode}^c_s(p)| \in \{o, c\} \) then \( \text{context}_s(p) = \text{context}^{c}_s(p) \).

For all \( p \in \text{Pred}^c_s \):
\[
\text{mode}^c_s(p) \leq |\text{mode}^c_s(p)| \quad \text{and} \quad \text{Import}^c_s(p) = \text{Import}^c_s(p).
\]

Let \( s \in S \) and \( s' \in S' \) s.t. \( \text{Nam}_s = \text{Nam}_{s'} \). Let \( p \in \text{Pred}^c_s \) s.t. \( \text{mode}^c_s(p), \text{mode}^c_s(p) \in \{d, o\} \) and let \( L \in [p]_S \). It holds that: if \( s \models_{\text{SEM}}^S L \) then \( s' \models_{\text{SEM}}^{S'} L \), for \( \text{SEM} \in \{\text{mAS}, \text{mWFS}\} \). □

Below, we state the main theorem of this subsection, in which general extension of a modular rule base is considered. This follows directly from Propositions 7.7 and 7.8.

**Theorem 7.9. (Monotonicity of definite and open predicates under modular rule base extension)** Let \( S, S' \) be modular rule bases s.t. \( S \leq S' \). Let \( s \in S \) and let \( s' \in S' \) s.t. \( \text{Nam}_s = \text{Nam}_{s'} \). Let \( p \in \text{Pred}^c_s \) s.t. \( \text{mode}^c_s(p), \text{mode}^c_s(p) \in \{d, o\} \) and let \( L \in [p]_S \). It holds that: if \( s \models_{\text{SEM}}^S L \) then \( s' \models_{\text{SEM}}^{S'} L \), for \( \text{SEM} \in \{\text{mAS}, \text{mWFS}\} \). □

### 7.2 c-Stratified Predicates over Modular Rule Bases

Let \( S \) be a modular rule base and let \( s \in S \). In this subsection, we define the c-stratified\(^{14}\) predicates of \( s \) w.r.t. \( S \). Intuitively, the definition of c-stratified predicates adapts the definition of stratified normal programs [Apt et al. 1988] to our framework.

Our aim is to show that:

- If a predicate \( p \) (i) is defined freely positively or negatively closed in \( s \), and (ii) is c-stratified in \( s \) w.r.t. \( S \) then inference on \( p \) from \( s \) w.r.t. \( S \) is fully determined.

  In particular, for any ground atom \( p(\tau) \in [p]_S \), it holds that:
  
  \( s \models_{\text{SEM}}^S p(\tau) \) or \( s \models_{\text{SEM}}^{S} \neg p(\tau) \), for \( \text{SEM} \in \{\text{mAS}, \text{mWFS}\} \).

- If a predicate \( p \) (i) is defined positively or negatively closed in \( s \) w.r.t. context \( \text{ctx} \), and (ii) is c-stratified in \( s \) w.r.t. \( S \) then inference on \( p \) from \( s \) w.r.t. \( S \) is fully determined within context \( \text{ctx} \). In particular, for any ground atom \( p(\tau) \in [p]_S \), it holds that:
  
  \( s \models_{\text{SEM}}^S p(\tau) \), or \( s \models_{\text{SEM}}^{S} \neg p(\tau) \), or \( s \models_{\text{SEM}}^{\text{SEM}} \neg \text{ctx}(\tau) \), for \( \text{SEM} \in \{\text{mAS}, \text{mWFS}\} \).

First, we provide a few auxiliary definitions. Let \( L \) be an objective literal. We define the extended predicate of \( L \), as follows:

\[
\text{pred}^c(L) = \begin{cases} 
\text{pred}(L) & \text{if } L \text{ is an atom}, \\
\neg \text{pred}(L) & \text{if } L \text{ is the strong negation of an atom}.
\end{cases}
\]

Let \( s \) be a rule base, we define the extended predicates of \( s \), as follows: \( \text{Pred}^c_s = \text{Pred}_s \cup \{\neg p \mid p \in \text{Pred}_s\} \). Let \( p \) be a predicate. We define \( \neg (\neg p) = p \) and \( |\neg p| = p \).

Additionally, we define \( \neg \text{Pred} = \{\neg p \mid p \in \text{Pred}\} \). Let \( S \) be a modular rule base. We define \( N_S = \{(m, s, x) \mid m \in \{d, o, c\}, s \in S, \text{ and } x \in \text{Pred}_s\} \).

---

\(^{14}\)The prefix c in the term c-stratified stands for closed because this notion applies to positively or negatively closed predicates, only.
We are now ready to define the direct c-dependence\(^{15}\) relationship between triples \(\langle m, s, x \rangle \in N_S\). Intuitively, \(\langle m, s, x \rangle\) directly c-depend on \(\langle m', s', x' \rangle\), if (i) the definition of \(x\) in rule base \(s\) in reasoning mode \(m\) depends on the definition of \(x'\) in rule base \(s'\) in reasoning mode \(m'\), or (ii) \(s = s'\), \(m = m' = c\), and the definition of \(x\) in \(s\) depends on the definition of \(\neg x\) in \(s'\) due to the corresponding CWA added by our program transformation.

**Definition 7.10. (Direct c-dependence)** Let \(S\) be a modular rule base and let \(s, s' \in S\). Additionally, let \(\langle m, s, x \rangle, \langle m', s', x' \rangle \in N_S\). We say that \(\langle m, s, x \rangle\) directly \(c\)-depends on \(\langle m', s', x' \rangle\) w.r.t. \(S\) (denoted by \(\langle m, s, x \rangle \leftarrow^c_S \langle m', s', x' \rangle\)) iff:

1. It holds that:
   1. there exists \(r \in P_s\) s.t. \(\text{pred}^{-}(\text{Head}_r) = x\), and there exists \(L' \in \text{Body}_r\) s.t.:
      1. \(\text{pred}^{-}(L') = x'\),
      2. if \(\text{qual}(L') \neq \text{a/a} \) then \(\text{Nam}_{x'} = \text{qual}(L')\),
      3. if \(s \neq s'\) then \(\text{Nam}_{x'} \in \text{Import}^S_{s'}(\text{pred}(L'))\),
   2. if \(s = s'\) then \(m' = \text{least}(m, \text{mode}^0_m(|x'|))\), and
   3. if \(s = s'\) then \(m' = \text{least}(m, \text{mode}^0_m(|x'|), \text{mode}^0_{x'}(|x'|))\), or

2. It holds that:
   1. \(s = s'\), \(m = m' = c\), and \(x' = \neg x\),
   2. if \(x \in \text{Pred} \) then \(\text{mode}^0_m(|x|) = c^-\),
   3. if \(x \in \neg \text{Pred} \) then \(\text{mode}^0_m(|x|) = c^+\) \(\Box\)

It is easy to see that if \(\langle m, s, x \rangle \leftarrow^c_S \langle m', s', x' \rangle\) and \(m' = c\) then \(m = c\).

**Example 22.** Consider the modular rule base \(S = \{s_1, s_2\}\) of Figure 2. Rule base \(s_2\) expresses that any person that is not (explicitly) a suspect, according to rule base \(s_1\), can enter country \(Z\). Note that even though sec: Suspect is declared as freely open in \(s_1\), it is requested by \(s_2\) from \(s_1\) in definite reasoning mode. Additionally, note that gov: Enter is declared as negatively closed in \(s_2\). Based on these, it follows that \(\langle c, s_2, \neg\text{gov: Enter} \rangle \leftarrow^c_S \langle d, s_1, \text{sec: Suspect} \rangle\) and \(\langle c, s_2, \text{gov: Enter} \rangle \leftarrow^c_S \langle c, s_2, \neg\text{gov: Enter} \rangle\) \(\Box\)

Let \(S\) be a modular rule base. The c-dependency graph of \(S\) is built by linking triples \(\langle m, s, x \rangle \in N_S\).

**Definition 7.11. (c-dependency graph of a MRB)** Let \(S\) be a modular rule base.

The c-dependency graph of \(S\) is \(DG^c_S = (N_S, E_S)\), where: \(E_S = \{\langle n, n' \rangle \in N_S \times N_S \mid n \leftarrow^c_S n'\}\).

A node \(n \in N_S\) c-depends on a node \(n' \in N_S\) w.r.t. \(S\) (denoted by \(n \leftarrow^c_S n'\)) iff there is a path in \(DG^c_S\) from \(n\) to \(n'\). \(\Box\)

Let \(S\) be a modular rule base, let \(s \in S\), and let \(x \in \text{Pred}^s\). We define the c-dependencies of \(x\) in \(s\) w.r.t. \(S\), as follows:

15The prefix c in the term c-dependence stands for closed and is used to differentiate this kind of dependency from other kinds of dependencies.
There exists a mapping level for all \( \langle \text{rule base}, s \rangle \). Let \( s \) w.r.t. mode there is a mapping, called level closed reasoning mode. Intuitively, the definition of \( x \) in \( s \) depends (directly or indirectly) only on predicates, which are considered in definite or closed reasoning mode, and

(1) the definition of \( x \) in \( s \) depends (due to the corresponding local CWA) on the weak negation of the definition of \( \neg x' \) in \( s' \) then the level of \( \langle c, s', \neg x' \rangle \) is less than the level of \( \langle c, s', x' \rangle \).

Definition 7.12. (c-stratified predicate w.r.t. a MRB) Let \( S \) be a modular rule base, let \( s \in S \), and let \( p \) be a predicate defined in \( s \) in closed reasoning mode. Intuitively, \( p \) is c-stratified in \( s \) w.r.t. \( S \) iff:

(1) there is a mapping, called level, from \( c-D^S_{x,s} \) to \( \mathbb{N} \) s.t.:
   - for all \( \langle m', s', x' \rangle \in c-D^S_{x,s} \),
     - if \( \langle m', s', x' \rangle \leftarrow_s \langle m'', s'', x'' \rangle \) then the level of \( \langle m'', s'', x'' \rangle \) is less than or equal to the level of \( \langle m', s', x' \rangle \), and
     - if the definition of \( x' \) in \( s' \) in closed reasoning mode depends (due to the corresponding local CWA) on the weak negation of the definition of \( \neg x' \) in \( s' \) then the level of \( \langle c, s', \neg x' \rangle \) is less than the level of \( \langle c, s', x' \rangle \).

(2) there exists a mapping level :\( c-D^S_{x,s} \rightarrow \mathbb{N} \) s.t.:
   - for all \( \langle m', s', x' \rangle \in c-D^S_{x,s} \), and for all \( \langle m'', s'', x'' \rangle \in N_S \) s.t. \( \langle m', s', x' \rangle \leftarrow_s \langle m'', s'', x'' \rangle \):
     - if \( x' \in \text{Pred} \), \( s'' = c \), \( s' = s'' \), and \( x'' = \neg x' \) then:
       i. if \( x' \in \text{Pred}, \text{mode}^S_x(x') = c^- \) then \( \text{level}(\langle c, s', \neg x' \rangle) < \text{level}(\langle c, s', x' \rangle) \).

Fig. 2. A modular rule base demonstrating the concept of c-dependence

\[
c-D^S_{x,s} = \{ \langle c, s, x \rangle \} \cup \{ \langle m', s', x' \rangle \in N_S \mid \langle c, s, x \rangle \leftarrow_S \langle m', s', x' \rangle \}.
\]

Let \( S \) be a modular rule base, \( s \in S \), and let \( p \) be a predicate defined in \( s \) in closed reasoning mode. Intuitively, \( p \) is c-stratified in \( s \) w.r.t. \( S \) iff:

(1) the definition of \( x \) in \( s \) depends (directly or indirectly) only on predicates, which are considered in definite or closed reasoning mode, and

(2) there is a mapping, called level, from \( c-D^S_{x,s} \) to \( \mathbb{N} \) s.t.:
   - for all \( \langle m', s', x' \rangle \in c-D^S_{x,s} \),
     - if \( \langle m', s', x' \rangle \leftarrow_s \langle m'', s'', x'' \rangle \) then the level of \( \langle m'', s'', x'' \rangle \) is less than or equal to the level of \( \langle m', s', x' \rangle \), and
     - if the definition of \( x' \) in \( s' \) in closed reasoning mode depends (due to the corresponding local CWA) on the weak negation of the definition of \( \neg x' \) in \( s' \) then the level of \( \langle c, s', \neg x' \rangle \) is less than the level of \( \langle c, s', x' \rangle \).

Definition 7.12. (c-stratified predicate w.r.t. a MRB) Let \( S \) be a modular rule base, \( s \in S \), and let \( p \) be a predicate defined in \( s \) in closed reasoning mode. Intuitively, \( p \) is c-stratified in \( s \) w.r.t. \( S \) iff:

(1) For all \( \langle m', s', x' \rangle \in c-D^S_{x,s} \), it holds that \( m' \in \{ d, c \} \).

(2) There exists a mapping level :\( c-D^S_{x,s} \rightarrow \mathbb{N} \) s.t.:
   - for all \( \langle m', s', x' \rangle \in c-D^S_{x,s} \), and for all \( \langle m'', s'', x'' \rangle \in N_S \) s.t. \( \langle m', s', x' \rangle \leftarrow_s \langle m'', s'', x'' \rangle \):
     - \( \text{level}(\langle m'', s'', x'' \rangle) \leq \text{level}(\langle m', s', x' \rangle) \), and
     - if \( m' = m'' = c, s' = s'' \), and \( x'' = \neg x' \) then:
       i. if \( x' \in \text{Pred}, \text{mode}^S_x(x') = c^- \) then \( \text{level}(\langle c, s', \neg x' \rangle) < \text{level}(\langle c, s', x' \rangle) \).
ii. if $x' \in \neg \text{Pred}$, $\text{mode}^0_{s'}(|x'|) = c^+$ then $\text{level}(|c, s', \neg x'|) < \text{level}(|c, s', x'|)$.

**Example 23.** Consider the modular rule base $S = \{s_1, s_2\}$ of Figure 2 and let $x = \text{gov:Enter}$. It holds that: $c-D_{c,s}^S = \{(d, s_1, \text{sec:Suspect}), (c, s_2, \neg \text{gov:Enter})\}$. Note that there exists a mapping level $c-D_{c,s}^S \rightarrow \text{IN}$ that satisfies the conditions of Definition 7.12(2). Therefore, $\text{gov:Enter}$ is $c$-stratified in $s_2$ w.r.t. $S$. \hfill \Box

**Proposition 7.13.** Let $S$ be a modular rule base and let $s \in S$. Let $p \in \text{Pred}^0_s$ s.t. $p$ is $c$-stratified in $S$ w.r.t. $S$ and let $L = p(c_1, ..., c_n)$, where $c_i \in \text{HU}_S$, for $i = 1, ..., n$. Let $\text{SEM} \in \{\text{mAS, mWFS}\}$.

1. If $p$ is freely (positively or negatively) closed in $s$ then: $s \models_{\text{SEM}} L$ or $s \not\models_{\text{SEM}} \neg L$.
2. If $p$ is (positively or negatively) closed in $s$ w.r.t. context $\text{ext}$ then:
   - $s \models_{\text{SEM}} L$, or $s \not\models_{\text{SEM}} \neg L$, or $s \models_{\text{SEM}} \neg \text{ctx}(c_1, ..., c_n)$. \hfill \Box

Let $S$ be a modular rule base. The following proposition shows that the $\text{mWebAS}$ and $\text{mWebWFS}$ semantics of the $c$-stratified predicates of a rule base $s \in S$ are equivalent in the case that $M_{c,S}^\text{AS}(s) \neq \emptyset$.

**Proposition 7.14.** Let $S$ be a modular rule base and let $s \in S$ s.t. $M_{c,S}^\text{AS}(s) \neq \emptyset$. Additionally, let $p \in \text{Pred}^0_s$ s.t. $p$ is $c$-stratified in $S$ w.r.t. $S$, and let $L \in [p]_s \cup \neg [p]_s$. It holds that: $s \models_{\text{SEM}} L$ iff $s \not\models_{\text{SEM}} \neg L$. \hfill \Box

Let $S$ be a modular rule base and let $s \in S$. As the following example shows the result of Proposition 7.14 does not hold if $M_{c,S}^\text{AS}(s) = \emptyset$.

**Example 24.** Consider the modular rule base $S = \{s\}$ of Figure 3. Note that $q$ is $c$-stratified in $S$ w.r.t. $S$, whereas $p$ is not. The latter is true because $(c, s, \neg p) \leftarrow^S (c, s, p)$ and $(c, s, p) \leftarrow^S (c, s, \neg p)$. Additionally, note that $M_{c,S}^\text{AS}(s) = \emptyset$, while $M_{c,S}^\text{AS}(s) \neq \emptyset$. Thus, $s \models_{\text{SEM}} L$, for all $L \in [\text{ex:q}]_s \cup \neg [\text{ex:q}]_s$, whereas $s \not\models_{\text{SEM}} \neg L$, just for $L \in \{\neg \text{ex:q}(d), \neg \text{ex:q}(c), \text{ex:q}(c), \text{ex:q}(d)\}$. \hfill \Box

![Rule base](http://example.org)

defines local negClosed ex:p.
defines local negClosed ex:q.

\[ \neg \text{ex:p}(c) \leftarrow \text{ex:p}(c). \]
\[ \neg \text{ex:q}(d). \]

Fig. 3. A rule base $s$ with no normal answer sets in reasoning mode $c$

Closing this Section, we would like to state that even if a predicate $p$ is $c$-stratified in $S$ w.r.t. $S$, the addition of a new rule base $s'$ to $S$ will require the conditions of Definition 7.12 to be re-checked, replacing $S$ with the new modular rule base $S'$. As future work, it would be interesting to examine whether there are reasonable restrictions of our framework that allow to test whether $p$ remains $c$-stratified in $S$ w.r.t. $S'$ with only local checks.
8. CONCLUSIONS

In this paper, we presented a principled framework for modular web rule bases, called \textit{MWeb}. According to this framework, each predicate $p$ defined in a rule base $s$ is characterized by its defining reasoning mode (\textit{definite}, \textit{open}, \textit{positively closed}, \textit{negatively closed}, or \textit{normal}), scope (\textit{global}, \textit{local}, or \textit{internal}), and exporting rule base list. Each predicate $p$ used in a rule base $s$ is characterized by its requesting reasoning mode (\textit{definite}, \textit{open}, \textit{closed}, or \textit{normal}), and importing rule base list. For legal \textit{MWeb} modular rule bases $\mathcal{S}$, the \textit{MWebAS} and \textit{MWebWFS} semantics of each $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined model-theoretically. These semantics extend the \textit{AS} [Gelfond and Lifschitz 1990; 1991] and \textit{WFSX} [Pereira and Alferes 1992; Alferes and Pereira 1996; Alferes et al. 1995] semantics on ELPs, respectively, keeping all of their semantical and computational characteristics. Additionally, the \textit{MWebAS} and \textit{MWebWFS} semantics of a rule base $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ can be defined (equivalently), through the \textit{AS} and \textit{WFSX} semantics of a set of ELPs, one for each $s \in \mathcal{S}$ and reasoning mode $d$, $o$, $c$, and $n$.

Our \textit{MWeb} framework for modular rule bases $\mathcal{S}$ supports:

— reasoning in four modes, \textit{definite}, \textit{open}, \textit{closed}, and \textit{normal}, which indicate, respectively, that weak negation is not accepted at all, only OWAs are accepted, both CWAs and OWAs are accepted, and weak negation is fully accepted,

— localized semantics and different points of view, as each rule base $s \in \mathcal{S}$ is associated with its own local models which, possibly, are in conflict with the local models of other rule bases in $\mathcal{S}$,

— the coexistence of local closed-world and local open-world assumptions, in the rule bases of $\mathcal{S}$, through the contextual CWA and contextual OWA rules. Note that the CWA context (resp. OWA context) of a closed (resp. open) predicate $p$ in a rule base $s \in \mathcal{S}$ can be used to delimit the application of the respective CWA (resp. OWA) to constants appearing in $s$, only,

— scoped negation-as-failure and scoped literal evaluation, through the use of qualified literals, the local predicate scope, and the restricted evaluation of literals, using only rule bases in $\mathcal{S}$,

— restricted propagation of local inconsistencies, making possible reasoning even in the presence of an inconsistency, local to a web rule base and reasoning mode,

— monotonicity of reasoning, for definite, open (and thus, also \textit{global}) predicates, in the case that $\mathcal{S}$ is extended,

— directed semantic relations, since if a rule base $s \in \mathcal{S}$ imports information from another rule base $s' \in \mathcal{S}$, this affects reasoning in $s$ but not in $s'$, and

— full determination of inference over a predicate $p$ in a rule base $s \in \mathcal{S}$ (resp. within a context $cxt$), if $p$ is defined positively or negatively closed in $s$ (resp. w.r.t. $cxt$) and $p$ is $c$-stratified in $s$ w.r.t. $\mathcal{S}$.

We want to emphasize that under the \textit{MWebWFS} semantics, declaring a predicate $p$ as \textit{open} or requesting a predicate $p$ in open reasoning mode, we do not obtain more entailments for $p$ than declaring $p$ as \textit{definite} or requesting $p$ in definite reasoning mode. This is because \textit{MWebWFS}, similarly to \textit{WFS}, cannot reason by cases and take advantage of the OWA rules. The reasoning mode \textit{open} makes sense for
We have included it in the MWebWFS semantics, for the sake of uniform presentation.

The MWeb framework has been implemented in the XSB system [Sagonas et al. 1994] and it is freely available from http://centria.di.fct.unl.pt/~cd/mweb. The system compiles an MWeb interface and corresponding logic program files into an XSB module, resorting to the tabling capabilities of this Prolog system. Both the MWebWFS and the MWebAS semantics are supported by the implementation, and can be selected by the user at query time. The MWebWFS implementation is more efficient because it uses natively XSB’s features to compute the well-founded semantics, and because of the lower theoretical complexity of MWebWFS for propositional programs. The MWebAS implementation requires the use of the external Smodels answer set solver [Niemelä and Simons 1997] which is called at query time by the MWeb toplevel shell whenever the query cannot be answered by the MWebWFS semantics (i.e., when the query is “undefined”). The MWeb distribution comes with a set of examples exploring the representation capabilities of the system. Additionally, it provides a preliminary integration of the MWeb framework with RDFS and ERDF ontologies [Analyti et al. 2008].

More and more knowledge is becoming available in the Semantic Web [KAON2], both in the form of RDFS and OWL ontologies. The Resource Description Framework Schema (RDFS) [Klyne and Carroll 2004; Hayes 2004] is a basic ontology language which can define class and property hierarchies, as well as properties with domain and ranges. For more complex domains, the Web Ontology Language (OWL-2) [OWL2 2009] provides declarative constructs to express complex concepts and statements about properties with the semantics provided by the description logic language SROIQ [Horrocks et al. 2006]. In both situations, rules are needed to overcome the limitations of both RDFS and OWL, which are extensively addressed in the literature (see for instance [Eiter et al. 2008]). The OWL-2 RL profile [Motik et al. 2009] is a syntactic subset of OWL-2 that is amenable to implementation using rule based technologies. Since reasoning with RDFS ontologies and OWL-2 RL ontologies can be achieved through logic programming [Ianni et al. 2009; Motik et al. 2009; Reynolds 2009], integration of MWeb rule bases with RDFS and OWL 2 RL ontologies can be achieved through an easy extension of our theory. However, the details of this extension are left for future work. Another interesting extension of our framework is the support of equality and expressive built-in predicates.

The MWeb framework proposes the complete separation of the interface part, which can be freely exchanged in the Web, containing the defines and uses declarations, and the associated logic program, containing the predicate definitions, which might be private (sensible data, etc.). However, it is outside the scope of the paper, how these mechanisms must be implemented in practice with the full generality required by the Semantic Web. In fact, trust and authorization could be much improved by security languages, such as the PeerTrust language [Gavriloae et al. 2004].

Other kinds of rules are expected to coexist in the Semantic Web, namely the so-called reactive, event-condition-action, or production rules. The RIF Working

---

16 The integration with the RDFS is illustrated in the current available implementation of MWeb, showing in practice the generality and appropriateness of the followed approach.
Group is defining a dialect for the language and semantics of production rules (RIF-PRD [de Sainte Marie et al. 2009]) respecting the Object Management Group’s adopted specification “Production Rule Representation” [OMG-PRR 2007], which is endorsed by major software companies. The condition part of the RIF-PRD production rules uses a logical syntax which is compatible with the one proposed here for the MWeb framework. In this way, all the knowledge that can be safely extracted from our MWeb rule bases can be plugged-in into RIF-PRD and used in the condition part for firing rules. The side-effects of the production rules can be easily reflected into our MWeb rule bases. However, these dynamic aspects are not covered in this paper.

Future work also concerns handling of local inconsistencies by, possibly, adapting existing paraconsistent semantics for extended logic programs [Alferes et al. 1995; Damásio and Pereira 1998; Sakama and Inoue 1995] to our MWeb framework. Additionally, we plan to define a notion of m-equivalent MWeb rule bases such that, for any modular rule base $S$ and $s \in S$, if $s$ is replaced in $S$ by an m-equivalent rule base $s'$ then the MWeb semantics of the other rule bases in $S$ will remain unaffected. This problem is related to the work in [Oikarinen and Janhunen 2006]. Another direction of research is the definition of modular Extended RDF (ERDF) ontologies, extending the Extended RDF framework, presented in [Analyti et al. 2008], with the modularity concepts proposed in this paper.

Closing, we would like to mention that the modularity 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]. The current proposal is a step in this direction.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: http://www.acm.org/pubs/citations/journals/jn/2010-V-N/p1-URLend.

Acknowledgments: This work has been benefited from discussions with Luís Moniz Pereira and José Júlio Alferes. 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 (cf. http://rewerse.net)

Received January 2009; revised January 2010; accepted May 2010

REFERENCES


Alferes, J. J. and Pereira, L. M. 1992. On Logic Program Semantics with Two Kinds of Nega-


