MWeb: a Principled Framework for Modular Web Rule Bases and its Semantics

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

[1] Institute of Computer Science, FORTH-ICS, Crete, Greece
[2] Department of Computer Science, University of Crete, Greece
[3] CENTRIA, Departamento de Informatica, Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
analyti@ics.forth.gr, antoniou@ics.forth.gr, cd@di.fct.unl.pt

Abstract. We present a principled framework for modular web rule bases, called MWeb. According to this framework, each predicate defined in a rule base is characterized by its defining reasoning mode, scope, and exporting rule base list. Each predicate used in a rule base is characterized by its requesting reasoning mode and importing rule base list. For legal MWeb modular rule bases $\mathcal{S}$, the MWebAS and MWebWFS semantics of each rule base $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined model-theoretically. These semantics extend the *answer set semantics* (AS) and the *well-founded semantics with explicit negation* (WFSX) on ELPs, respectively, keeping all of their semantical and computational characteristics. Our framework supports: (i) local semantics and different points of view, (ii) local closed-world and open-world assumptions, (iii) scoped negation-as-failure, (iv) restricted propagation of local inconsistencies, and (v) monotonicity of reasoning, for "fully shared" predicates.

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

1 Introduction

The Semantic Web research [11] 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 [63, 51, 60, 59, 62] has started the discussion on *local closed-world assumptions* [36] and *scoped negation-as-failure* (otherwise, called *scoped default negation*) [41, 59]. 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 [10]. Moreover, the combination of weak and strong negation allows the distinction between open and closed predicates, as shown in [5]. 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 [16]. 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 $\mathcal{S}$ is a set of `MWeb` rule bases[4]. Each rule base $s \in \mathcal{S}$ can import knowledge about a predicate p from other rule bases in $\mathcal{S}$ 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 nonmonotonic reasoning forms on p are not allowed. On the other hand, a rule base that defines a predicate p can use nonmonotonic 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`, `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 $s \in S$ and extend, respectively, two major semantics for extended logic programs (ELPs), namely answer set semantics (`AS`) [31, 32] and well-founded semantics with explicit negation (`WFSX`) [53, 3, 1]. 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[5]. Additionally, it supports local semantics and different points of view, local closed-world and open-world

[4] 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.

[5] 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 $\mathcal{S}$.

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 $\neg p(\bar{c})$[6] 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 [62].

Initial ideas for our framework are presented in [19]. However, the operational semantics of modular rule bases, presented there, are not equivalent 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 [4]. 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 [4] 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 [4], is provided in the Related Work Section.

Below, we justify our choice for choosing the answer set semantics (`AS`) [31, 32] and the well-founded semantics with explicit negation (`WFSX`) [53, 3, 1] 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 nonmonotonic negation [68, 5, 69]. The integration of monotonic and nonmonotonic 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 nonmonotonic formalisms, like default logic and nonmonotonic modal logics (e.g. see [12]).

The study of weak negation resulted in two major semantics, the well-founded semantics [29] and the stable model semantics [30]. The study of strong negation introduced new problems and resulted in two major extensions of the previous semantics, well-founded semantics with explicit negation [53, 1] and answer set semantics [31]. For the non-paraconsistent case, both semantics obey to the "coherence principle",

[6] Note that $\neg p(\bar{c})$ denotes the strong (or explicit) negation of $p(\bar{c})$.

first specified in [53], which states that if something is known to be false then it is believed false (or, more formally, if $\neg L$ then $\sim L$). Other contending semantics, like the partial stable semantics for disjunctive programs (p-stable models) [56], 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 [52, 17]. 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 [17] 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 [56], another to WFSX, and the remaining one to strong negation p-stable models [2]. 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 [64], the Smodels [45], and the DLV system [44]). 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`. Section 8 reviews related work and conclusions are provided in Section 9. The proofs of all Propositions, Theorems, and Corollaries are found in the 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* [22] 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[7] 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"^^`d`. For example, "27"^^`xsd:integer` is a typed literal.

In our framework, *predicates names* are IRI references. Each *rule base* s is associated with a name Nam_s, which is also an IRI reference. A *constant* is an IRI reference or an RDF literal [57]. A *term* is a constant or a variable. An (`MWeb`) *atom* is a *simple atom* $p(t_1, ..., t_k)$ or a *qualified atom* $p(t_1, ..., t_k)@Nam_t$, where p is a predicate of arity k, t_i, for $i = 1, ..., k$, are terms and Nam_t is the name of a rule base t. An *objective literal* is either an atom A or the strong negation $\neg A$ of an atom A. A *default literal* is the weak negation $\sim L$ of an objective literal L. An (`MWeb`) *literal* is an objective or a default literal. An (`MWeb`) *rule* r is a formula of the form: $L \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots \sim L_n$, where L is a simple atom or the strong negation of a simple atom, and L_i (for $i \in \{1, ..., n\}$) is an objective literal. We say that r is *objective*, if no default literal appears in r. An (`MWeb`) *logic program* P is a set of rules. Note that if no qualified atom appears in P then P is an ELP.

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

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, $\overline{t}$ denotes a sequence of terms, $\overline{x}$ denotes a sequence of variables, and $\overline{c}$ denotes a sequence of constants.

Example 1. Consider a rule base s_1 with name: Nam_{s_1} =`http://gov.countryX`. Rule base s_1 is associated with a logic program[8] P_{s_1} that expresses immigration laws of an imaginary country `X`.

```
Enter(?p) ←   CountryEU(?c), citizenOf(?p,?c).
Enter(?p) ← ¬ CountryEU(?c), citizenOf(?p,?c), ¬ RequiresVisa(?c).
Enter(?p) ← ¬ CountryEU(?c), citizenOf(?p,?c), RequiresVisa(?c),
              HasVisa(?p).
```

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.

[7] The *Resource Description Framework* (RDF) is a framework for modeling meta-data about web resources, recommended by W3C [57].
[8] To improve readability, namespace prefixes have been eliminated from the example IRIs.

These rules are complemented with the following knowledge:

```
  CountryEU(Austria).                     ¬ RequiresVisa(Croatia).
¬ CountryEU(Djibuti).                       RequiresVisa(China).
¬ CountryEU(China).                         HasVisa(Chen).
  Person(?p) ← citizenOf(?p,?c).
  Country(?c) ← citizenOf(?p,?c).
```

Consider now another rule base s_2 with name Nam_{s_2} =`http://security.int` whose associated logic program P_{s_2} is the following:

```
citizenOf(Anne,Austria).                  citizenOf(Chen,China).
citizenOf(Boris,Croatia).                 citizenOf(Dil,Djibuti).
```

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

Depending on their reasoning mode, predicates defined in a rule base s are declared as `definite`, `open`, `positively closed`, `negatively closed`, or `normal`. In contrast to *normal* predicates, *definite*, *open*, and *closed* predicates impose restrictions on the use of weak negation in their defining rules. 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 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:

$$openRules_s(p) = \{\neg p(\overline{x}) \leftarrow cxt(\overline{x}), \sim p(\overline{x}), \quad p(\overline{x}) \leftarrow cxt(\overline{x}), \sim \neg p(\overline{x})\},$$

We refer to these rules, as the *contextual OWA rules* of p in s and to predicate cxt, as the *OWA context* of p in s. The contextual OWA rules of a predicate p in s provide a mechanism for making *local OWAs*. In particular, they express that if there exists $\overline{c}$ s.t. $cxt(\overline{c})$ is true in an intended model M of s then $p(\overline{c})$ or $\neg p(\overline{c})$ is true in M. If p is declared open in s without context information then p is called *freely open* in s, and we write the OWA rules as: $openRules_s(p) = \{\neg p(\overline{x}) \leftarrow \sim p(\overline{x}), \ \ p(\overline{x}) \leftarrow \sim \neg p(\overline{x})\}$.

Similarly, if a predicate p is declared positively 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:

$$posClosure_s(p) = \{\neg p(\overline{x}) \leftarrow cxt(\overline{x}), \sim p(\overline{x})\},$$

called *positive contextual CWA rule*, or

$$negClosure_s(p) = \{p(\overline{x}) \leftarrow cxt(\overline{x}), \sim \neg p(\overline{x})\},$$

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 $\overline{c}$ s.t. $cxt(\overline{c})$ is true in an intended model M of s, if $p(\overline{c})$ is believed to be false in M then $\neg p(\overline{c})$ is true in M. Similarly, the negative closure rule of a predicate p in s expresses that, for any $\overline{c}$ s.t. $cxt(\overline{c})$ is true in an intended model M of s, if $\neg p(\overline{c})$ is believed to be false in M then $p(\overline{c})$ is true in M. If p is declared positively or negatively closed in s without context information then p is called *freely positively* or *freely negatively closed* in s, respectively. That is, $posClosure_s(p) = \{\neg p(\overline{x}) \leftarrow \sim p(\overline{x})\}$ and $negClosure_s(p) = \{p(\overline{x}) \leftarrow \sim \neg p(\overline{x})\}$, respectively.

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, `Enter(Anne)` and `Enter(Chen)` are obtained from $P_{s_1} \cup P_{s_2}$, under both `AS` and `WFSX`. Interestingly, `Enter(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 `CountryEU` is open. By declaring `CountryEU` open in s_1 w.r.t. `Country`, the following two rules are added to the definition of `CountryEU`:

```
  CountryEU(?c) ← Country(?c), ∼ ¬ CountryEU(?c).
¬ CountryEU(?c) ← Country(?c), ∼ CountryEU(?c).
```

It is now concluded from $P_{s_1} \cup P_{s_2} \cup openRules_{s_1}$(`CountryEU`) under `AS` that `Enter(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, `Enter` is declared positively closed in s_1 w.r.t. `Person`. Then, the following rule is added to the definition of `Enter`: `¬ Enter(?p) ← Person(?p), ∼ 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 $P_{s_1} \cup\ P_{s_2} \cup openRules_{s_1}$(`CountryEU`) $\cup\ posClosure_{s_1}$(`Enter`), under both `AS` and `WFSX`, that `¬Enter(Dil)` holds, while all previous inferences remain unaffected.

However, if we add to rule base s_1 the fact "`¬ RequiresVisa(Djibuti).`" then it can be concluded from $P_{s_1} \cup\ P_{s_2} \cup openRules_{s_1}$(`CountryEU`) $\cup\ posClosure_{s_1}$(`Enter`), under both `AS` and `WFSX`, that `Enter(Dil)` 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 *defining reasoning mode* of p in s and takes the values `definite, open, posClosed, negClosed`, and `normal`. The latter reasoning mode is referred to as the *requesting reasoning mode* of p in s' and takes the values `definite, open, closed`, and `normal`. We also define the scopes (`global, local`, or `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 $\mathcal{S}$ be a modular rule base. As seen in Section 2, each rule base $s \in \mathcal{S}$ is associated with a name 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 $\mathcal{S}$, is integrated. Therefore, each rule base $s \in \mathcal{S}$ is also associated with an *interface* Int_s that contains two kinds of declarations, `defines` and `uses`, that have the following syntax:

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

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

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

`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 *UsesPred* clause. The user can state the rule bases from which s requests p, through the *from* clause. If this clause is omitted then s requests p from any providing 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 $\mathtt{m}$ and that another rule base $s' \in \mathcal{S}$ imports p from s in a requesting reasoning mode $\mathtt{m}'$ different than $\mathtt{m}$. Then, reasoning modes $\mathtt{m}$ (declared by exporter) and $\mathtt{m}'$ (declared by importer) are combined as shown in Table 1. Note that the final reasoning mode in which s' imports p from s equals $\mathit{least}(\mathtt{m}, \mathtt{m}')$, where $\mathtt{definite} < \mathtt{open} < \mathtt{closed} < \mathtt{normal}$. However, an error is caused if the exporting rule base s defines p in normal reasoning mode and the importing rule base s' declares that 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 1, 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 `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 `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 posClosed` p.   `uses open` p `from` 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 1, rule base s' imports p from s, as if p had been declared in s in open reasoning mode. □

Table 1. Combinations of defining and requesting reasoning modes

importer \ exporter	definite	open	closed	normal
normal	definite	open	closed	normal
closed	definite	open	closed	error
open	definite	open	open	error
definite	definite	definite	definite	error

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

`defines local negClosed` p. `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 $\mathcal{S}$) in normal reasoning mode. Then, according to Table 1, rule base s' imports p from s in closed reasoning mode. □

As we have already mentioned, it is possible that two rule bases $s, s' \in \mathcal{S}$ 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 2.

Table 2. Allowed combinations of scopes of the same predicate

scope in s' \ scope in s	global	local	internal
global	allowed	error	allowed
local	error	error	allowed
internal	allowed	allowed	allowed

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 $p = \mathit{pred\_name}/n$ is associated with a predicate name $\mathit{pred\_name} \in \mathcal{IRI}$ and a unique arity $n \in \mathbb{N}$ (as in Prolog systems). In this case, the definition of predicate indicator *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[9]. Rule base s_1, with Nam_{s_1} =<http://europa.eu>, defines the list of European Union countries (which does not include Croatia), stating that this list is positively closed w.r.t. the CWA context geo:Country. Rule base s_2, with Nam_{s_2} =<http://security.int>, provides international citizenship information and lists persons suspect of crimes. Rule base s_3, with Nam_{s_3} =<http://geography.int>, provides geographical information, stating a positively closed list of countries.

Finally, rule base s_4, with Nam_{s_4} =<http://gov.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 eu:CountryEU is defined in s_1, in positively closed reasoning mode, rule base s_4 imports eu:CountryEU from s_1 in open reasoning mode. Furthermore, note that s_4 imports sec:citizenOf and sec:Suspect from s_2 in definite reasoning mode. Even though sec:citizenOf is defined in s_2 in local scope, sec:citizenOf is also defined internally in s_4 and additional facts about this predicate are stated. This is allowed since the internal information of sec:citizenOf in s_4 is not made public. Finally, note the presence of a default qualified literal in the rules of P_{s_4}.

Note that s_2 is providing confidential information to any requester. Safety can be improved if s_2 specifies the authorized consumers of sec:citizenOf, as in:
defines global open sec:citizenOf visible to <http://gov.countryY>. $\square$

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 $\mathcal{IRI}$ and the set of RDF literals by $\mathcal{LIT}$. Additionally, we denote the set of variable symbols by *Var*. The sets *Var*, $\mathcal{IRI}$, and $\mathcal{LIT}$ are pairwise disjoint.

Definition 1 (Vocabulary). An (MWeb) *vocabulary* V is a triple $\langle \mathit{RBase}, \mathit{Pred}, \mathit{Const} \rangle$, where $\mathit{RBase} \subseteq \mathcal{IRI}$ is a set of rule base names, $\mathit{Pred} \subseteq \mathcal{IRI}$ is a set of predicate names, and $\mathit{Const} \subseteq \mathcal{IRI} \cup \mathcal{LIT}$ is a set of constant symbols. $\square$

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

Definition 2 (Atom). Let $V = \langle \mathit{RBase}, \mathit{Pred}, \mathit{Const} \rangle$ be a vocabulary. An *atom* over V is a *simple atom* $p(t_1, ..., t_n)$ or a *qualified atom* $p(t_1, ..., t_n)@\mathit{rbase}$, where $p \in \mathit{Pred}$, $\mathit{rbase} \in \mathit{RBase}$, $n = \mathit{arity}(p)$, and t_i is a term over V, for $i = 1, ..., n$. $\square$

Definition 3 (Literal). Let $V = \langle \mathit{RBase}, \mathit{Pred}, \mathit{Const} \rangle$ be a vocabulary. An *objective literal* over V is an atom A or the strong negation $\neg A$ of an atom A over V. A *default literal* over V is the weak negation $\sim L$ of an objective literal L over V. A *literal* over V is an objective or a default literal over V. $\square$

We denote the set of objective literals over V and the set of literals over V by $\mathit{Lit}^{\circ}(V)$ and $\mathit{Lit}(V)$, respectively. Let $L \in \mathit{Lit}(V)$. We define $\mathit{pred}(L) = p$, where p is

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

Rule base s_1

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

Rule base s_2

```
⟨http://security.int⟩
defines local open sec:citizenOf.
defines glocal open sec:Suspect.

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

Rule base s_3

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

Rule base s_4

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

Fig. 1. An `MWeb` modular rule base

the predicate symbol appearing in L. If L is built from a qualified atom $p(\bar{t})@rbase$, we define the *qualifying rule base* of L as $qual(L) = rbase$. Otherwise, $qual(L)$ is undefined.

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

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

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

Definition 4 (Logic program). Let $V = \langle RBase, Pred, Const\rangle$ be a vocabulary. A *rule* r over V is an expression $L_0 \leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n$, where: (i) $L_0 \in Lit^{\circ}(V)$ is a simple literal (i.e., $qual(L_0)$ is undefined), (ii) $L_i \in Lit^{\circ}(V) \cup \{\mathtt{t}, \mathtt{u}\}$, for $i = 1, ..., m$, and (iii) $L_i \in Lit^{\circ}(V)$, for $i = m+1, ..., n$. We define $Head_r = L_0$, $Body^+_r = \{L_1, ..., L_m\}$, $Body^-_r = \{L_{m+1}, ..., L_n\}$, and $Body_r = Body^+_r \cup \sim Body^-_r$. 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: "$L_0 \leftarrow \mathtt{t}$" by the fact "$L_0$.".

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

Definition 5 (Rule base). Let $V = \langle RBase, Pred, Const \rangle$ be a vocabulary. A *rule base* s over V is a triple $s = \langle Nam_s, P_s, Int_s \rangle$, where: (i) $Nam_s \in RBase$ is the *name* of s, (ii) P_s is a logic program over V, called the *logic program* of s, and (iii) $Int_s = \langle Def_s, Use_s \rangle$ is the *interface* of s, where:

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

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

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

We define[10]: $|\mathtt{d}| = \mathtt{d}$, $|\mathtt{o}| = \mathtt{o}$, $|\mathtt{c}^+| = |\mathtt{c}^-| = \mathtt{c}$, and $|\mathtt{n}| = \mathtt{n}$. Then, we impose the following total order: $\mathtt{d} < \mathtt{o} < \mathtt{c} < \mathtt{n}$, called *reasoning mode extension*. Additionally, we impose the following total order on predicate scopes: $\mathtt{int} < \mathtt{lc} < \mathtt{gl}$, called *predicate scope extension*.

Example 7. Consider rule base s_1 of Example 6. Then, $Def_{s_1} = \{\langle$ `eu:CountryEU`, `lc`, $\mathtt{c}^+$, `geo:Country`, $\{*\}\rangle\}$ and $Use_{s_1} = \{\langle$ `geo:Country`, `d`, $\{$`<http://geography.int>`$\}\rangle\}$. □

[10] This auxiliary definition is needed, because the defining reasoning modes of a predicate p are $\{\mathtt{d}, \mathtt{o}, \mathtt{c}^+, \mathtt{c}^-, \mathtt{n}\}$, 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 $\{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$.

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

Definition 6 (Legal rule base). A rule base $s = \langle Nam_s, P_s, Int_s \rangle$ is *legal* iff:
(*Legality Constraints*)

1. If $\langle p, sc, mod, cxt, Exp \rangle, \langle p, sc', mod', cxt', Exp' \rangle \in Def_s$ then $sc = sc'$, $mod = mod'$, $cxt = cxt'$, and $Exp = Exp'$.
2. If $\langle p, mod, Imp \rangle, \langle p, mod', Imp' \rangle \in Use_s$ then $mod = mod'$ and $Imp = Imp'$.
3. For all $r \in P_s$:
   (a) $pred(Head_r) \in Pred_s^{\mathtt{D}}$.
   (b) $Body_r \cap \{\mathtt{u}\} = \emptyset$.
   (c) for all $L \in Body_r - \{\mathtt{t}\}$, $\quad pred(L) \in Pred_s^{\mathtt{D}} \cup Pred_s^{\mathtt{U}}$.
4. For all $\langle p, sc, mod, cxt, Exp \rangle \in Def_s$:
   (a) if $mod \in \{\mathtt{o}, \mathtt{c}^+, \mathtt{c}^-\}$ and $cxt \in Pred$ then $cxt \in Pred_s$ and $arity(cxt) = arity(p)$,
   (b) if $cxt \in Pred_s^{\mathtt{D}}$ then $mode_s^{\mathtt{D}}(cxt) \in \{\mathtt{d}\}$,
   (c) if $cxt \in Pred_s^{\mathtt{U}}$ then $mode_s^{\mathtt{U}}(cxt) \in \{\mathtt{d}\}$,
   (d) if $mod \in \{\mathtt{d}, \mathtt{n}\}$ then $cxt = \mathtt{n/a}$.
5. If $p \in Pred_s^{\mathtt{D}}$ and $scope_s(p) = \mathtt{gl}$ then $mode_s^{\mathtt{D}}(p) \in \{\mathtt{d}, \mathtt{o}\}$.
6. If $p \in Pred_s^{\mathtt{D}}$ and $scope_s(p) = \mathtt{int}$ then $Export_s(p) = \{\}$.
7. If $p \in Pred_s^{\mathtt{D}} \cap Pred_s^{\mathtt{U}}$ then $mode_s^{\mathtt{U}}(p) \leq |mode_s^{\mathtt{D}}(p)|$.
8. For all $r \in P_s$, and for all $L \in Body_r$:
   if $qual(L) \in RBase$ then $qual(L) \in Import_s(pred(L))$ or $Import_s(pred(L)) = \{*\}$.
9. For all $r \in P_s$, and for all $L \in Body_r$:
   if $mode_s^{\mathtt{D}}(pred(Head_r)) \neq \mathtt{n}$ then:
   (a) $Body_r^- = \{\}$,
   (b) for all $L \in Body_r^+$, if $pred(L) \in Pred_s^{\mathtt{D}}$ then $mode_s^{\mathtt{D}}(pred(L)) \neq \mathtt{n}$, and
   (c) for all $L \in Body_r^+$, if $pred(L) \in Pred_s^{\mathtt{U}}$ then $mode_s^{\mathtt{U}}(pred(L)) \neq \mathtt{n}$. □

Let s be a legal rule base. Constraint 1 of Definition 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 `u` should not appear in the body of any rule $r \in P_s$. Further, for each predicate appearing in the body of r, there should be a corresponding `defines` or `uses` declaration. Constraint 4 expresses that each open or closed predicate p, defined in s, can be associated with a predicate cxt. This predicate cxt should be defined in s or requested by s and have the same arity as p. If cxt is defined in (resp. requested by) s then its defining (resp. requesting) reasoning mode should be definite. Constraint 5 expresses that each global predicate of s should be defined in definite or open reasoning mode. This is because reasoning on global predicates should be monotonic. Constraint 6 expresses that each internal predicate of s is not visible by other rule bases. Constraint 7 expresses that if a predicate p is both defined in s and requested by s then its defining reasoning mode in s should extend its requesting reasoning mode in s. Intuitively, this means that the use of weak negation in the imported definition of p should satisfy the constraints of the defining reasoning mode of p in s. Constraint 8 expresses that if a qualified literal L appears in the body of a rule $r \in P_s$ then rule base s should request $pred(L)$ from rule base $qual(L)$. Constraint 9 expresses that for each rule $r \in P_s$, if the defining reasoning mode of the predicate appearing in $Head_r$ is restricted (i.e., not `normal`) then: (i) no default literal should appear in $Body_r$, and (ii) the defining (resp. requesting) reasoning mode of each defined (resp. requested) predicate appearing in $Body_r$ should also be restricted.

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

Definition 7 (Modular rule base). A *modular rule base* $\mathcal{S}$ over a vocabulary V is a set of legal rule bases over V. $\square$

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

$$Export_s^{\mathcal{S}}(p) = \begin{cases} \{Nam_{s'} \mid s' \in \mathcal{S} - \{s\}\} & \text{if } Export_s(p) = \{*\} \\ Export_s(p) \cap \{Nam_{s'} \mid s' \in \mathcal{S}\} & \text{otherwise} \end{cases}$$

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

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

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

$$Import_s^{\mathcal{S}}(p) = \begin{cases} ExportingTo_{\mathcal{S}}(p, s) & \text{if } Import_s(p) = \{*\} \\ Import_s(p) \cap ExportingTo_{\mathcal{S}}(p, s) & \text{otherwise,} \end{cases}$$

where $ExportingTo_{\mathcal{S}}(p, s) = \{Nam_{s'} \mid s' \in \mathcal{S},\ Nam_s \in Export_{s'}^{\mathcal{S}}(p)\}$.
Intuitively, $ExportingTo_{\mathcal{S}}(p, s)$ denotes the rule bases in $\mathcal{S}$ that are willing to export p to s. Additionally, $Import_s^{\mathcal{S}}(p)$ denotes the rule bases in $\mathcal{S}$ from which s imports p. We refer to $Import_s^{\mathcal{S}}(p)$ as the *importing rule base list* of p in s w.r.t. $\mathcal{S}$. Note that: For all $p \in Pred_s$, $Nam_s \notin Export_s^{\mathcal{S}}(p)$ and $Nam_s \notin Import_s^{\mathcal{S}}(p)$.

Example 10. For the modular rule base $\mathcal{S}$ of Example 6, $ExportingTo_{\mathcal{S}}(\texttt{sec:citizenOf}, s_4) = \{s_2\}$. Additionally, $Import_{s_4}^{\mathcal{S}}(\texttt{sec:citizenOf}) = \{s_2\}$. $\square$

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

Definition 8 (Legal modular rule base). A modular rule base $\mathcal{S}$ is *legal* iff: (*Legality Constraints*)

1. If $s \in \mathcal{S}$ then s is a legal rule base.
2. If $s, s' \in \mathcal{S}$ and $s \neq s'$ then $Nam_s \neq Nam_{s'}$.
3. For all $s, s' \in \mathcal{S}$ s.t. $s \neq s'$, and for all $p \in Pred_s^{\mathtt{D}}$:
   if $scope_s(p) = \texttt{int}$ then $p \notin Pred_{s'}^{\mathtt{U}}$ or $Nam_s \notin Import_{s'}^{\mathcal{S}}(p)$.

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

Let $\mathcal{S}$ be a legal modular rule base. Constraint 1 of Definition 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 rule base $s \in \mathcal{S}$ defines internally a predicate p then another rule base $s' \in \mathcal{S}$ cannot request p from s. Constraint 4 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `normal` reasoning mode and requested by another rule base $s' \in \mathcal{S}$ from s then its requesting reasoning mode in s' should also be `normal`. This is because the use of weak negation in the definition of p in s is unrestricted. Constraint 5 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `local` scope then it can be defined by another rule base $s' \in \mathcal{S}$ only in `internal` scope. This is because internal predicates are invisible to other rule bases. Constraint 6 expresses that if a predicate p is defined in a rule base $s \in \mathcal{S}$ in `global` scope then it can be defined by another rule base $s' \in \mathcal{S}$ only in `global` or `internal` scope. Constraint 7 expresses that if a rule base $s \in \mathcal{S}$ requests a predicate p from a *specific* rule base s' then s' should be a rule base of $\mathcal{S}$ that defines p and is willing to export p to s. That is, $\mathit{Import}_s(p) = \{*\}$ or $\mathit{Import}_s(p) = \mathit{Import}_s^{\mathcal{S}}(p)$.

Example 11. Modular rule base $\mathcal{S}$ of Example 6 is legal. □

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

5 Model-theoretic Semantics for Modular Rule Bases

In this section, we propose the `MWeb` *answer set semantics* (`MWebAS`) and the `MWeb` *well-founded semantics* (`MWebWFS`) of modular rule bases. We will show that these semantics extend the answer set semantics (`AS`) [31] and the well-founded semantics with explicit negation (`WFSX`) [53, 1] 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* $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$.

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 the constants appearing in its constituent rule bases.

Definition 9 (Herbrand universe of a MRB). Let $\mathcal{S} = \{s_1, ..., s_n\}$. The *Herbrand universe* of $\mathcal{S}$ is defined as: $\mathtt{HU}_{\mathcal{S}} = \mathtt{HU}_{s_1} \cup ... \cup \mathtt{HU}_{s_n}$, where $\mathtt{HU}_{s_i}$ (for $i = 1, ..., n$) is the set of constants appearing in P_{s_i}. □

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

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

Definition 10 (Herbrand base of a rule base w.r.t. a MRB). The *Herbrand base* of s w.r.t. $\mathcal{S}$ is defined as:

$$\mathtt{HB}_s^{\mathcal{S}} = \{[p]_{\mathcal{S}} \mid p \in Pred_s\} \ \cup \ \{[p@rbase]_{\mathcal{S}} \mid p \in Pred_s^{\mathtt{U}}, \ rbase \in Import_s^{\mathcal{S}}(p)\}. \ \square$$

Definition 11 (Simple normal interpretation of a rule base w.r.t. a MRB). A *simple normal interpretation* of s w.r.t. $\mathcal{S}$ is a set $I \subseteq \mathtt{HB}_s^{\mathcal{S}}$ s.t. $I \cap \neg I = \emptyset$ (*consistency*) or $I = \mathtt{HB}_s^{\mathcal{S}}$. □

Let I be a simple normal interpretation of s w.r.t. $\mathcal{S}$. If $I = \mathtt{HB}_s^{\mathcal{S}}$ then I is called *inconsistent*. Otherwise, I is called *consistent*. As usual, I can be seen, equivalently, as a function from $\mathtt{HB}_s^{\mathcal{S}} \rightarrow \{0, 1\}$, where: (i) $I(L) = 1$, if $L \in I$, and (ii) $I(L) = 0$, if $L \notin I$. Let $L \in \mathtt{HB}_s^{\mathcal{S}}$. 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 $\mathtt{t}$. In particular, we define $I(\mathtt{t}) = 1$.

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

Definition 12 (Simple extended interpretation of a rule base w.r.t. a MRB). A *simple extended interpretation* of s w.r.t. $\mathcal{S}$ is a set $I = T \cup \sim F$, where $T, F \subseteq \mathtt{HB}_s^{\mathcal{S}}$ s.t. either:

- $T \cap \neg T = \emptyset$ and $T \cap F = \emptyset$ (*consistency*), or
- $T = F = \mathtt{HB}_s^{\mathcal{S}}$. □

Let $I = T \cup \sim F$ be a simple extended interpretation of s w.r.t. $\mathcal{S}$. If $I = \mathtt{HB}_s^{\mathcal{S}} \cup \sim \mathtt{HB}_s^{\mathcal{S}}$ then I is called *inconsistent*. Otherwise, I is called *consistent*. As usual, I can be seen, equivalently, as a function from $\mathtt{HB}_s^{\mathcal{S}} \cup \sim \mathtt{HB}_s^{\mathcal{S}} \rightarrow \{0, 1/2, 1\}$, where: (i) $I(L) = 1$, if $L \in I$, (ii) $I(L) = 0$, if $L \notin I$ and $\sim L \in I$, and (iii) $I(L) = 1/2$, if $L \notin I$ and $\sim L \notin I$. I also assigns truth values to special literals $\mathtt{t}$ and $\mathtt{u}$. In particular, we define $I(\mathtt{t}) = 1$ and $I(\mathtt{u}) = 1/2$. If $\neg T \subseteq F$ then I is called *coherent*.

Let I be a simple normal or extended interpretation of s w.r.t. $\mathcal{S}$ and let $S \subseteq \mathtt{HB}_s^{\mathcal{S}} \cup \sim \mathtt{HB}_s^{\mathcal{S}} \cup \{\mathtt{t}, \mathtt{u}\}$. We define: $I(S) = min\{I(L) \mid L \in S\}$.

Definition 13 (Dependencies of a rule base in a reasoning mode w.r.t. a MRB). The *dependencies* of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, denoted by $D_{s,\mathtt{m}}^{\mathcal{S}}$, is the minimal set of pairs $\langle s', \mathtt{x} \rangle$, where $s' \in \mathcal{S}$ and $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, that satisfies the following constraints:

1. $\langle s, \mathtt{m} \rangle \in D_{s,\mathtt{m}}^{\mathcal{S}}$,

2. If $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$ and there exists $p \in Pred^{\mathtt{D}}_{s'}$ s.t. $\mathtt{x} > |mode^{\mathtt{D}}_{s'}(p)|$ then $\langle s', \mathtt{y}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$, where $\mathtt{y} = |mode^{\mathtt{D}}_{s'}(p)|$,
3. If $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$ and there exists $p \in Pred^{\mathtt{U}}_{s'}$ s.t. $Nam_{s''} \in Import^{\mathcal{S}}_{s'}(p)$, for $s'' \in \mathcal{S}$, then $\langle s'', \mathtt{y}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$, where $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p), |mode^{\mathtt{D}}_{s''}(p)|)$. $\square$

Intuitively, if $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$ and $\langle s', \mathtt{x}\rangle \neq \langle s, \mathtt{m}\rangle$ then the meaning of the predicates in $Pred_s$ in rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ depends on the meaning of the predicates in $Pred^{\mathtt{D}}_{s'}$ in rule base s' in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$.

Example 12. Consider the modular rule base $\mathcal{S}$ of Example 6. It holds that: $D^{\mathcal{S}}_{s_1,\mathtt{c}} = \{\langle s_1, \mathtt{c}\rangle, \langle s_3, \mathtt{d}\rangle\}$. Thus, the meaning of `eu:CountryEU` and `geo:Country` in rule base s_1 in reasoning mode `c` depends on the meaning of the predicate `geo:Country` in rule base s_3 in reasoning mode `d`. Similarly, it holds that: $D^{\mathcal{S}}_{s_4,\mathtt{n}} = \{\langle s_4, \mathtt{n}\rangle, \langle s_4, \mathtt{c}\rangle, \langle s_4, \mathtt{o}\rangle, \langle s_1, \mathtt{o}\rangle, \langle s_2, \mathtt{d}\rangle, \langle s_3, \mathtt{d}\rangle\}$. $\square$

Definition 14 (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 $\mathtt{m}$ w.r.t. $\mathcal{S}$ is a set $\mathsf{I} = \{I^{\mathtt{x}}_{s'} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$, such that:

1. for all $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$, $I^{\mathtt{x}}_{s'}$ is a simple normal (resp. extended) interpretation of s' w.r.t. $\mathcal{S}$, and
2. if there exists $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$ s.t. $I^{\mathtt{x}}_{s'}$ is inconsistent then for all $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$, $I^{\mathtt{x}}_{s'}$ is inconsistent. $\square$

Let I be a normal (resp. extended) interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. 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 $\mathtt{m}$ w.r.t. $\mathcal{S}$.

Example 13. Consider the modular rule base $\mathcal{S}$ of Example 6. Let $\mathsf{I} = \{I^{\mathtt{c}}_{s_1}, I^{\mathtt{d}}_{s_3}\}$, where $I^{\mathtt{c}}_{s_1} = \{\neg$ `eu:Country(Canada)`$\}$ and $I^{\mathtt{d}}_{s_3} = \{$`geo:Country(Egypt)`$\}$. Then, I is a (consistent) normal interpretation of s in reasoning mode `c` w.r.t. $\mathcal{S}$. Now, let $\mathsf{J} = \{J^{\mathtt{c}}_{s_1}, J^{\mathtt{d}}_{s_3}\}$, where $J^{\mathtt{c}}_{s_1} = \{\neg$ `eu:Country(Canada)`, $\sim$ `eu:Country(Egypt)`$\}$ and $I^{\mathtt{d}}_{s_3} = \{$`geo: Country(Egypt)`$\}$. Then, J is a (consistent) extended interpretation of s in reasoning mode `c` w.r.t. $\mathcal{S}$. $\square$

Below, we order the normal and extended interpretations of a rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, according to *truth ordering* ($\leq_{\mathtt{t}}$) and *knowledge ordering* ($\leq_{\mathtt{k}}$) [27].

Definition 15 (Truth and knowledge ordering of normal & extended interpretations). Let I and J be normal (resp. extended) interpretations of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. We say that:

– J *extends* I w.r.t. *truth* ($\mathsf{I} \leq_{\mathtt{t}} \mathsf{J}$) iff:

$$\text{For all } \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}} \text{ and for all } L \in \mathtt{HB}^{\mathcal{S}}_{s'}, \quad I^{\mathtt{x}}_{s'}(L) \leq J^{\mathtt{x}}_{s'}(L).$$

– J *extends* I w.r.t. *knowledge* ($\mathsf{I} \leq_{\mathtt{k}} \mathsf{J}$) iff:

$$\text{For all } \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}, \quad I^{\mathtt{x}}_{s'} \subseteq J^{\mathtt{x}}_{s'}. \ \square$$

Note that if I, J are normal interpretations then $\mathsf{I} \leq_{\mathtt{t}} \mathsf{J}$ iff $\mathsf{I} \leq_{\mathtt{k}} \mathsf{J}$. 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 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 w.r.t. a modular rule base, a few auxiliary definitions are provided. Let P be a logic program. We will denote by $[P]_{\mathcal{S}}$ the set of rules in P instantiated over $\mathtt{HU}_{\mathcal{S}}$.

Below, we define logic program satisfaction, as usual.

Definition 16 (Logic program satisfaction). Let I be a simple normal (resp. extended) interpretation of s w.r.t. $\mathcal{S}$. We say that I *satisfies* a logic program P ($I \models P$) iff for all $r \in [P]_{\mathcal{S}},\ I(Head_r) \geq I(Body_r)$. □

For each $s \in \mathcal{S}$, we define[11] 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 m w.r.t. S.

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

$$\begin{aligned} P_s^{\mathtt{o}} = \ & \{r \in P_s \mid mode_s^{\mathtt{D}}(pred(Head_r)) \in \{\mathtt{o}, \mathtt{c}^+, \mathtt{c}^-\}\} \cup \\ & \{openRules_s(p) \mid p \in Pred_s^{\mathtt{D}} \text{ and } mode_s^{\mathtt{D}}(p) \in \{\mathtt{o}, \mathtt{c}^+, \mathtt{c}^-\}\}. \end{aligned}$$

$$\begin{aligned} P_s^{\mathtt{c}} = \ & \{r \in P_s \mid mode_s^{\mathtt{D}}(pred(Head_r)) \in \{\mathtt{c}^+, \mathtt{c}^-\}\} \cup \\ & \{posClosure_s(p) \mid p \in Pred_s^{\mathtt{D}} \text{ and } mode_s^{\mathtt{D}}(p) = \mathtt{c}^+\} \cup \\ & \{negClosure_s(p) \mid p \in Pred_s^{\mathtt{D}} \text{ and } mode_s^{\mathtt{D}}(p) = \mathtt{c}^-\}. \end{aligned}$$

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

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

Intuitively, $P_s^{\mathtt{d}}$ is the set of rules in P_s whose head predicate is defined in definite, open, or closed reasoning mode. $P_s^{\mathtt{o}}$ 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. $P_s^{\mathtt{c}}$ is 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, $P_s^{\mathtt{n}}$ 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:

$$\begin{aligned} P_{s_1}^{\mathtt{d}} &= P_{s_1}, \\ P_{s_1}^{\mathtt{o}} &= P_{s_1} \cup \{\neg\mathtt{eu{:}CountryEU(?x)} \leftarrow \mathtt{geo{:}Country(?x)}, \sim \mathtt{eu{:}CountryEU(?x)}., \\ &\qquad\qquad \mathtt{eu{:}CountryEU(?x)} \leftarrow \mathtt{geo{:}Country(?x)}, \sim\neg\mathtt{eu{:}CountryEU(?x)}.\}, \\ P_{s_1}^{\mathtt{c}} &= P_{s_1} \cup \{\neg\mathtt{eu{:}CountryEU(?x)} \leftarrow \mathtt{geo{:}Country(?x)}, \sim \mathtt{eu{:}CountryEU(?x)}.\}, \\ P_{s_1}^{\mathtt{n}} &= \{\}. \ \square \end{aligned}$$

The following $P/_{\mathcal{S}}^{\mathtt{mAS}} I$ *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 [31] (also known as *Gelfond-Lifschitz transformation*).

[11] Rules $openRules_s(p)$, $posClosure_s(p)$, and $negClosure_s(p)$ are defined in Section 2.

Definition 17. (Transformation $P/^{\mathtt{mAS}}_{\mathcal{S}}I$). Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let I be a simple normal interpretation of s w.r.t. $\mathcal{S}$. Let P be a logic program. The logic program $P/^{\mathtt{mAS}}_{\mathcal{S}}I$ is obtained from $[P]_{\mathcal{S}}$ as follows:

1. Remove from $[P]_{\mathcal{S}}$, 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 $P/^{\mathtt{mWFS}}_{\mathcal{S}}I$ *modulo* transformation, which is is actually an adaptation of the P/I modulo transformation of `WFSX` [53] to our framework. The $P/^{\mathtt{mWFS}}_{\mathcal{S}}I$ *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 18. (Transformation $P/^{\mathtt{mWFS}}_{\mathcal{S}}I$). Let I be a simple extended interpretation of s w.r.t. $\mathcal{S}$. Let P be a logic program. The logic program $P/^{\mathtt{mWFS}}_{\mathcal{S}}I$ is obtained from $[P]_{\mathcal{S}}$ as follows:

1. Remove from $[P]_{\mathcal{S}}$, all rules that contain in their body an objective literal L s.t. $I(\neg L) = 1$ or a default literal $\sim L$ s.t. $I(L) = 1$.
2. Remove from the body of 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 $\mathcal{S}$ of Example 6.

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

Consider now the simple *normal* interpretation of s_4 w.r.t. $\mathcal{S}$, $I = \{$`sec:Suspect(Peter)@`Nam_{s_2}, $\neg$ `eu:CountryEU(Egypt)`$\}$. Then,

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

Additionally, consider the simple *extended* interpretation of s_4 w.r.t. $\mathcal{S}$, $I = \{$`sec:Suspect(Peter)@`Nam_{s_2}, $\neg$ `eu:CountryEU(Egypt)`, $\sim$ `eu:CountryEU(Egypt)`$\}$. Then,

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

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

Definition 19 (Minimal model of a rule base in a reasoning mode w.r.t. a MRB and a normal or extended interpretation). Let $\mathsf{N} = \{N^{\mathtt{x}}_{s'} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$ be a normal (resp. extended) interpretation of s in reasoning mode `m` w.r.t. $\mathcal{S}$. The *minimal model* of s in reasoning mode `m` w.r.t. $\mathcal{S}$ and N, denoted by $least^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{N})$, is the minimal (w.r.t. $\leq_{\mathtt{t}}$) normal (resp. extended) interpretation of $\mathcal{S}$, $\mathsf{M} = \{M^{\mathtt{x}}_{s'} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$, such that:
For all $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$:

1. For all $p \in Pred^{\mathtt{D}}_{s'}$ s.t. $\mathtt{x} > |mode^{\mathtt{D}}_{s'}(p)|$, and for all $L \in [p]_{\mathcal{S}}$:
   $M^{\mathtt{x}}_{s'}(L) = M^{\mathtt{y}}_{s'}(L)$, where $\mathtt{y} = |mode^{\mathtt{D}}_{s'}(p)|$,

2. For all $p \in Pred^{\mathtt{U}}_{s'}$, and for all $s'' \in \mathcal{S}$ s.t. $Nam_{s''} \in Import^{\mathcal{S}}_{s'}(p)$:
   (a) for all $L \in [p]_{\mathcal{S}}$:
   $M^{\mathtt{x}}_{s'}(L) \geq M^{\mathtt{y}}_{s''}(L)$, where $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p), |mode^{\mathtt{D}}_{s''}(p)|)$,
   (b) for all $L \in [p@Nam_{s''}]_{\mathcal{S}}$:
   $M^{\mathtt{x}}_{s'}(L) = M^{\mathtt{y}}_{s''}(simple(L))$, where $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p), |mode^{\mathtt{D}}_{s''}(p)|)$,

3. $M^{\mathtt{x}}_{s'} \models P^{\mathtt{x}}_{s'} /^{\mathtt{mAS}}_{\mathcal{S}} N^{\mathtt{x}}_{s'}$ (resp. $M^{\mathtt{x}}_{s'} \models P^{\mathtt{x}}_{s'} /^{\mathtt{mWFS}}_{\mathcal{S}} N^{\mathtt{x}}_{s'}$). □

Intuitively, Definition 19 expresses that if L is a literal defined in s' at reasoning mode $\mathtt{y}$ then the truth value of L, according to $M^{\mathtt{x}}_{s'}$, for $\mathtt{x} \geq |\mathtt{y}|$, is equal to the truth value of L, according to $M^{|\mathtt{y}|}_{s'}$. If L is a simple (resp. qualified) literal imported in s' from a rule base s'' then the truth value of L, according to $M^{\mathtt{x}}_{s'}$, is greater than or equal (resp. equal) to the truth value of L, according to $M^{\mathtt{y}}_{s''}$, where $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p), |mode^{\mathtt{D}}_{s''}(p)|)$ and $p = pred(L)$. Additionally, it holds that: $M^{\mathtt{x}}_{s'} \models P^{\mathtt{x}}_{s'} /^{\mathtt{mAS}}_{\mathcal{S}} N^{\mathtt{x}}_{s'}$ (resp. $M^{\mathtt{x}}_{s'} \models P^{\mathtt{x}}_{s'} /^{\mathtt{mWFS}}_{\mathcal{S}} N^{\mathtt{x}}_{s'}$), for $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. Thus, for $M^{\mathtt{d}}_{s'}$, we consider the logic program $P^{\mathtt{d}}_{s'}$ which contains the rules that define the definite, open, and (positively or negatively) closed predicates in s. For $M^{\mathtt{o}}_{s'}$, we consider the logic program $P^{\mathtt{o}}_{s'}$ 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 $M^{\mathtt{c}}_{s'}$, we consider the logic program $P^{\mathtt{c}}_{s'}$ 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 $M^{\mathtt{n}}_{s'}$, we consider the logic program $P^{\mathtt{n}}_{s'}$ which contains the rules that define the normal predicates in s.

The following proposition states that Definition 19 is well-defined.

Proposition 1. Let N be a normal (resp. extended) interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. It always exists the minimal model of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ and N. □

Below, we adapt the definition of the *Coh* operator in `WFSX` [53] to our framework.

Definition 20 (*Coh* operator). Let $\mathsf{I} = \{I^{\mathtt{x}}_{s'} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$ be an extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. We define $Coh(\mathsf{I}) = \{Coh(I^{\mathtt{x}}_{s'}) \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$, where $Coh(I^{\mathtt{x}}_{s'}) = I^{\mathtt{x}}_{s'} \cup \{\sim L \mid L \in \mathtt{HB}^{\mathcal{S}}_{s'} \text{ and } \neg L \in I^{\mathtt{x}}_{s'}\}$. □

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

Example 16. Consider the modular rule base $\mathcal{S}$ of Example 6. Additionally, consider the extended interpretation of s_1 in reasoning mode $\mathtt{c}$ w.r.t. $\mathcal{S}$, $\mathsf{I} = \{I^{\mathtt{c}}_{s_1}, I^{\mathtt{d}}_{s_3}\}$, where $I^{\mathtt{c}}_{s_1} = \{\neg$ `eu:Country(Canada)`, $\sim$ `eu:Country(Egypt)`$\}$ and $I^{\mathtt{d}}_{s_3} = \{$`geo:Country(Egypt)`$\}$. It holds that $Coh(\mathsf{I}) = \{Coh(I^{\mathtt{c}}_{s_1}), Coh(I^{\mathtt{d}}_{s_3})\}$, where $Coh(I^{\mathtt{c}}_{s_1}) = \{\neg$ `eu:Country(Canada)`, $\sim$ `eu:Country(Egypt)`, $\sim$ `eu:Country(Canada)`$\}$ and $Coh(I^{\mathtt{d}}_{s_3}) = \{$`geo:Country(Egypt)`, $\sim\neg$ `geo:Country(Egypt)`$\}$. □

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

Definition 21 (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 $\mathtt{m}$ w.r.t. $\mathcal{S}$. M is a *normal* (resp. *extended*) *answer set* of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, if $\mathsf{M} = least^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$ (resp. $\mathsf{M} = Coh(least^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M}))$).

We denote the set of normal answer sets of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ by $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ and the set of extended answer sets of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ by $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$. $\square$

Example 17. Consider the modular rule base $\mathcal{S}$ of Example 6.
Let $L = \neg\mathtt{eu:CountryEU(Croatia)}$.

For all $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s_1)$, it holds that: $M^{\mathtt{c}}_{s_1}(L) = 1$. Note that $|\mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s_1)| = 1$. For all $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{o},\mathcal{S}}(s_1)$, it holds that: $M^{\mathtt{o}}_{s_1}(L) \in \{0, 1\}$. Additionally, for all $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{n},\mathcal{S}}(s_4)$, it holds that: $M^{\mathtt{o}}_{s_4}(L) \in \{0, 1\}$ and $M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) \in \{0, 1\}$. Note that rule base s_4 requests predicate `eu:CountryEU` from rule base s_1 in open reasoning mode. Furthermore, it holds that:

- $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Anne)}) = 1$. This is because:
  (i) $M^{\mathtt{n}}_{s_4}(\mathtt{eu:CountryEU(Austria)}) = 1$ and
  (ii) $M^{\mathtt{n}}_{s_4}(\sim\mathtt{sec:Suspect(Anne)@}\langle \textit{http://security.int}\rangle) = 1$.
- $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) = 1$. This is because:
  (i) $M^{\mathtt{n}}_{s_4}(\mathtt{eu:CountryEU(Croatia)}) = 1$ or $M^{\mathtt{n}}_{s_4}(\neg\mathtt{eu:CountryEU(Croatia)}) = 1$,
  (ii)$M^{\mathtt{n}}_{s_4}(\neg\mathtt{gov:RequiresVisa(Croatia)}) = 1$, and
  (iii) $M^{\mathtt{n}}_{s_4}(\sim\mathtt{sec:Suspect(Boris)@}\langle \textit{http://security.int}\rangle) = 1$.
- $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Peter)}) = 0$. This is because:
  $M^{\mathtt{n}}_{s_4}(\sim\mathtt{sec:Suspect(Peter)@}\langle \textit{http://security.int}\rangle) = 0$.

Note that, there exists $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{n},\mathcal{S}}(s_4)$ such that: $M^{\mathtt{o}}_{s_1}(L) = M^{\mathtt{o}}_{s_4}(L) = M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) = 0$ (resp. $M^{\mathtt{o}}_{s_1}(L) = M^{\mathtt{o}}_{s_4}(L) = M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) = 1$).

For all $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{c},\mathcal{S}}(s_1)$, it holds that: $M^{\mathtt{c}}_{s_1}(L) = 1$. Note that $|\mathcal{M}^{\mathtt{EAS}}_{\mathtt{c},\mathcal{S}}(s_1)| = 1$. For all $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{o},\mathcal{S}}(s_1)$), it holds that: $M^{\mathtt{o}}_{s_1}(L) \in \{0, 1/2, 1\}$. Additionally, for all $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4)$, it holds that: $M^{\mathtt{o}}_{s_4}(L) \in \{0, 1/2, 1\}$ and $M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) \in \{0, 1/2, 1\}$. Furthermore, it holds that: $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Anne)}) = 1$, $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) \in \{1/2, 1\}$, and $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Peter)}) = 0$. Note that for $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4)$ s.t. $M^{\mathtt{n}}_{s_4}(\mathtt{eu:CountryEU(Croatia)}) = 1/2$, it holds that $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) = 1/2$.

Note that:

- There exists $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4)$ such that: $M^{\mathtt{o}}_{s_4}(\neg L) = M^{\mathtt{c}}_{s_4}(\neg L) = M^{\mathtt{n}}_{s_4}(\neg L) = 1$, and $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) = 1$.
- There exists $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4)$ such that: $M^{\mathtt{o}}_{s_4}(L) = M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) = 1/2$, and $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) = 1/2$.
- There exists $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4)$ such that: $M^{\mathtt{o}}_{s_1}(L) = M^{\mathtt{o}}_{s_4}(L) = M^{\mathtt{c}}_{s_4}(L) = M^{\mathtt{n}}_{s_4}(L) = 1$, and $M^{\mathtt{n}}_{s_4}(\mathtt{gov:Enter(Boris)}) = 1$. $\square$

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

Proposition 2. Let M be a consistent normal or extended interpretation of rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Let $s' \in \mathcal{S}$ and let $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$. Additionally, let $\mathsf{M}' = \{M^{\mathtt{y}}_{s''} \in \mathsf{M} \mid \langle s'', \mathtt{y}\rangle \in D^{\mathcal{S}}_{s',\mathtt{x}}\}$. It holds that:

1. If $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{x},\mathcal{S}}(s')$.
2. If $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{x},\mathcal{S}}(s')$. $\square$

The following proposition states that if there exists an inconsistent normal answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ then this is the only normal answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Additionally, $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ is either empty, or it consists of the inconsistent extended answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$[12].

Proposition 3. If there exists $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent then:

1. $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$, and
2. $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}'\}$, where M' is inconsistent, or $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\}$. $\square$

Example 18. Consider the modular rule base $\mathcal{S}$ of Example 6 and assume that we add to rule base s_1 the fact `¬eu:CountryEU(Austria)`. Then, for all $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s_1) = \{\mathsf{M}\}$, where M is inconsistent and $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s_1) = \{\mathsf{M}'\}$, 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:

```
 gov:Enter(?p) ←  eu:CountryEU(?c), sec:citizenOf(?p,?c).
¬gov:Enter(?p) ←  eu:CountryEU(?c), sec:citizenOf(?p,?c).
```

Then, $\mathcal{M}^{\mathtt{AS}}_{\mathtt{n},\mathcal{S}}(s_4) = \{\mathsf{M}\}$, where M is inconsistent, whereas $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{n},\mathcal{S}}(s_4) = \{\}$. This is because $P^{\mathtt{n}}_{s_4} /^{\mathtt{mWFS}}_{\mathcal{S}} \mathtt{HB}^{\mathcal{S}}_{s_4} = \{$`¬ gov:RequiresVisa(Croatia).`, `sec:citizenOf(Peter, Greece).`$\}$. Thus, $\mathsf{M}' \neq \mathit{Coh}(\mathit{least}^{\mathtt{n}}_{\mathcal{S}}(s_4, \mathsf{M}'))$, where M' is the inconsistent extended interpretation of s_4 in reasoning mode $\mathtt{n}$ w.r.t. $\mathcal{S}$. $\square$

Based on Proposition 3, we define a contradictory rule base in a reasoning mode w.r.t. a modular rule base.

Definition 22 (Contradictory rule base in a reasoning mode w.r.t. a MRB). If $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$ s.t. M is inconsistent then rule base s is called *contradictory* in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. $\square$

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

Proposition 4. If there exists $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent then (i) rule base s is contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, and (ii) $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$. $\square$

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

Proposition 5. Assume that rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ is contradictory. It holds that:

[12] Recall that there is only one inconsistent normal (resp. extended) interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$.

1. If $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}\}$ then rule base s in reasoning mode $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ w.r.t. $\mathcal{S}$ is also contradictory.
2. If $s' \in \mathcal{S}$ and $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\langle s, \mathtt{m} \rangle \in D^{\mathcal{S}}_{s',\mathtt{x}}$ then rule base s' in reasoning mode $\mathtt{x}$ 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 $\neg$`eu:CountryEU(Austria)`. Then, for all $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, rule bases s_1 and s_4 in reasoning mode $\mathtt{x}$ are contradictory, while rule bases s_2 and s_3 in reasoning mode $\mathtt{x}$ are not. This is because rule base s_4 imports predicate `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 `gov:Enter(Anne)` and $\neg$`gov:Enter(Anne)`. It holds that rule base s_4 in reasoning mode $\mathtt{n}$ is contradictory. However, rule base s_4 in reasoning modes $\mathtt{d}, \mathtt{o}, \mathtt{c}$ is non-contradictory. This is because contradictory information is isolated to normal reasoning mode of rule base s_4, due to Definition 13. Similarly, for all $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, rule bases s_1, s_2, and s_3 in reasoning mode $\mathtt{x}$ are non-contradictory. □

Let $\mathcal{S}$ be a modular rule base, let $s, s' \in \mathcal{S}$, and let $\mathtt{m}, \mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. It is possible that there is no normal (resp. extended) answer set of rule base s in reasoning mode $\mathtt{m}$, even though there is a normal (resp. extended) answer set of rule base s' in reasoning mode $\mathtt{x}$. 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 $\neg$`gov:Enter(Anne)`. It holds that there is no normal or extended answer set of rule base s_4 in reasoning mode $\mathtt{n}$ w.r.t. $\mathcal{S}$. This is because, as we show in Example 17, `gov:Enter(Anne)` is also derived from the rules. However, there is a normal and a extended answer set of rule base s_4 in reasoning modes $\mathtt{d}, \mathtt{o}, \mathtt{c}$ w.r.t. $\mathcal{S}$. This is because, due to Definition 13, the semantics of rule base s_4 in reasoning modes $\mathtt{d}, \mathtt{o}, \mathtt{c}$ are defined independently to reasoning mode $\mathtt{n}$. Similarly, for all $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, there is a normal and a extended answer set of rule bases s_1, s_2, and s_3 in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$. This is because rule bases s_1, s_2, and s_3 do not import predicate `gov:Enter` from rule base s_4. □

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

Proposition 6. It holds that: $|minimal_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))| \leq 1$. □

Based on Proposition 6, we define the *modular well-founded model* of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, as follows:

Definition 23 (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 $\mathtt{m}$ w.r.t. $\mathcal{S}$, $\mathtt{mWF}^{\mathcal{S}}_{\mathtt{s},\mathtt{m}}$, as follows:

1. $\mathtt{mWF}^{\mathcal{S}}_{s,\mathtt{m}} = least_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))$, if $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) \neq \emptyset$, and
2. $\mathtt{mWF}^{\mathcal{S}}_{s,\mathtt{m}}$ is the inconsistent extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, otherwise. □

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

Definition 24 (MWebAS & MWebWFS entailment). Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let:

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

We say that:

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

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

The following corollary follows directly from Definition 23 and Proposition 6.

Corollary 1. Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let:

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

It holds that: $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $M_s^{\mathtt{m}}(L) = 1$, where $\mathsf{M} = \mathtt{mWF}_{s,\mathtt{m}}^{\mathcal{S}}$. $\square$

6 Transformational Semantics for Modular Rule Bases

In this section, we show how a modular rule base $\mathcal{S}$ can be transformed into a set of extended logic programs (ELPs) whose conclusions capture the MWeb model-theoretic semantics of $\mathcal{S}$. In particular, for each $s \in \mathcal{S}$, four extended logic programs (ELPs) are generated, one for each reasoning mode: `definite, open, closed`, and `normal`. We denote these ELPs by $\Pi_{s,\mathcal{S}}^{\mathtt{d}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{o}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{c}}$, and $\Pi_{s,\mathcal{S}}^{\mathtt{n}}$, respectively. We assume that the legality of $\mathcal{S}$ has already being verified, by checking that: (i) each $s \in \mathcal{S}$ is a legal rule base (see Definition 6) and (ii) $\mathcal{S}$ is a legal modular rule base (see Definition 8).

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

Let $s \in \mathcal{S}$. To proceed, we need to define a total and injective function, $\tau_s^{\mathtt{x}}(.)$, for $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, from $\mathtt{HB}_s^{\mathcal{S}}$ to the set of ELP literals, appearing in the generated ELPs $\Pi_{s,\mathcal{S}}^{\mathtt{d}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{o}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{c}}$, and $\Pi_{s,\mathcal{S}}^{\mathtt{n}}$. Specifically, let $A = p(\bar{t}) \in \mathtt{HB}_s^{\mathcal{S}}$ be a simple atom and let $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. We define: (a) $\tau_s^{\mathtt{x}}(A) = Nam_s{:}\mathtt{x}\_p(\bar{t})$ and (b) $\tau_s^{\mathtt{x}}(\neg A) = \neg Nam_s{:}\mathtt{x}\_p(\bar{t})$. Additionally, let $A = p(\bar{t})@Nam_t \in \mathtt{HB}_s^{\mathcal{S}}$ be a qualified atom and let $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. We define: (a) $\tau_s^{\mathtt{x}}(A) = Nam_s{:}\mathtt{x}\_p@Nam_t(\bar{t})$ and (b) $\tau_s^{\mathtt{x}}(\neg A) = \neg Nam_s{:}\mathtt{x}\_p@Nam_t(\bar{t})$. For an atom $A \in \mathtt{HB}_s^{\mathcal{S}}$, we define: $\tau_s^{\mathtt{x}}(\sim A) = \sim\tau_s^{\mathtt{x}}(A)$.

In the generated ELPs, the symbols $Nam_s{:}\mathtt{x}\_p$ and $Nam_s{:}\mathtt{x}\_p@Nam_t$, for $s, t \in \mathcal{S}$, $p \in Pred_s$, and $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, correspond to predicate names[13]. Intuitively, the IRI of the owner rule base, Nam_s, and the IRI of the qualification rule base, Nam_t, control the evaluation scope of the web predicate p, while $\mathtt{x}$ indicates the reasoning mode.

Our transformational approach proceeds as follows: First, for all $s \in \mathcal{S}$, the auxiliary ELPs $P^{\mathtt{d}}_{s,\mathcal{S}}$, $P^{\mathtt{o}}_{s,\mathcal{S}}$, $P^{\mathtt{c}}_{s,\mathcal{S}}$, and $P^{\mathtt{n}}_{s,\mathcal{S}}$ are generated, which initially contain the facts "$\mathtt{domain}(c)$.", where $c \in \mathtt{HU}_{\mathcal{S}}$.

Let $s \in \mathcal{S}$ with $Nam_s = n_s$. The `defines` declarations of s are translated as follows:

1. Let p be a predicate that is defined as `definite` in s (that is, $mode_s^{\mathtt{D}}(p) = \mathtt{d}$). 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 rule, denoted by $r^{\mathtt{d}}$:

$$\tau_s^{\mathtt{d}}(L_0) \leftarrow \tau_s^{\mathtt{d}}(L_1), \ldots, \tau_s^{\mathtt{d}}(L_m).$$

Rule $r^{\mathtt{d}}$ is added to ELP $P^{\mathtt{d}}_{s,\mathcal{S}}$ and corresponds to the case where the reasoning mode of s is `definite`. Additionally, the following rules are generated:

$$\begin{array}{ll} n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{d}\_p(\overline{x}). & \neg n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{d}\_p(\overline{x}). \\ n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{d}\_p(\overline{x}). & \neg n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{d}\_p(\overline{x}). \\ n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{d}\_p(\overline{x}). & \neg n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{d}\_p(\overline{x}). \end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$ and $k = arity(p)$.
The rules in the first line of the above set of rules are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$ and correspond to the case where the reasoning mode of s is `open`. The rules in the second line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$ and correspond to the case where the reasoning mode of s is `closed`. The rules in the third line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$ and correspond to the case where the reasoning mode of s is `normal`.

2. Let $p \in Pred_s^{\mathtt{D}}$ be a predicate that is defined as `open` w.r.t. a predicate cxt in s (that is, $mode_s^{\mathtt{D}}(p) = \mathtt{o}$ 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 $r^{\mathtt{d}}$ and $r^{\mathtt{o}}$, respectively:

$$\begin{array}{c} \tau_s^{\mathtt{d}}(L_0) \leftarrow \tau_s^{\mathtt{d}}(L_1), \ldots, \tau_s^{\mathtt{d}}(L_m). \\ \tau_s^{\mathtt{o}}(L_0) \leftarrow \tau_s^{\mathtt{o}}(L_1), \ldots, \tau_s^{\mathtt{o}}(L_m). \end{array}$$

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

$$\begin{array}{ll} \neg n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim n_s{:}\mathtt{o}\_p(\overline{x}). & n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim\neg n_s{:}\mathtt{o}\_p(\overline{x}). \\ \neg n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{o}\_p(\overline{x}). & n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_p(\overline{x}). \\ \neg n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{o}\_p(\overline{x}). & n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_p(\overline{x}). \end{array}$$

[13] In order to avoid name clashes, it is assumed that IRIs always appear between delimiters "⟨" and "⟩".

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$ and $k = arity(p)$.
Rule $r^{\mathtt{o}}$ and the rules in the first line of the above set of rules are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$. The rules in the second line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$. The rules in the third line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

3. Let $p \in Pred^{\mathtt{D}}_s$ be a predicate that is defined as *freely* `open` in s (that is, $mode^{\mathtt{D}}_s(p) = \mathtt{o}$ and $context_s(p) = \mathtt{n/a}$). This case is treated similarly to the previous case with the difference that literals $n_s{:}\mathtt{o}\_cxt(\overline{x})$ are eliminated from the generated rules.
4. Let $p \in Pred^{\mathtt{D}}_s$ be a predicate that is defined as `positively closed` w.r.t. a predicate cxt in s (that is, $mode^{\mathtt{D}}_s(p) = \mathtt{c}^+$ 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 $r^{\mathtt{d}}$, $r^{\mathtt{o}}$, and $r^{\mathtt{c}}$, respectively:

$$\tau^{\mathtt{d}}_s(L_0) \leftarrow \tau^{\mathtt{d}}_s(L_1), \ldots, \tau^{\mathtt{d}}_s(L_m).$$
$$\tau^{\mathtt{o}}_s(L_0) \leftarrow \tau^{\mathtt{o}}_s(L_1), \ldots, \tau^{\mathtt{o}}_s(L_m).$$
$$\tau^{\mathtt{c}}_s(L_0) \leftarrow \tau^{\mathtt{c}}_s(L_1), \ldots, \tau^{\mathtt{c}}_s(L_m).$$

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

$$\neg n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim n_s{:}\mathtt{o}\_p(\overline{x}). \qquad n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim\neg n_s{:}\mathtt{o}\_p(\overline{x}).$$
$$\neg n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{c}\_cxt(\overline{x}), \sim n_s{:}\mathtt{c}\_p(\overline{x}).$$
$$\neg n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{c}\_p(\overline{x}). \qquad n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{c}\_p(\overline{x}).$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$ and $k = arity(p)$.
Rule $r^{\mathtt{o}}$ and the rules in the first line of the above set of rules are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$. Rule $r^{\mathtt{c}}$ and the rule in the second line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$. The rules in the third line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.
5. Let $p \in Pred^{\mathtt{D}}_s$ be a predicate that is defined as *freely* `positively closed` in s (that is, $mode^{\mathtt{D}}_s(p) = \mathtt{c}^+$ and $context_s(p) = \mathtt{n/a}$). This case is treated similarly to the previous case with the difference that literals $n_s{:}\mathtt{o}\_cxt(\overline{x})$ and $n_s{:}\mathtt{c}\_cxt(\overline{x})$ are eliminated from the generated rules.
6. Let $p \in Pred^{\mathtt{D}}_s$ be a predicate that is defined as `negatively closed` w.r.t. a predicate cxt in s (that is, $mode^{\mathtt{D}}_s(p) = \mathtt{c}^-$ and $context_s(p) = cxt$). This case is similar to Case 4. Specifically, if P_s contains a rule r:

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

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

$$\tau^{\mathtt{d}}_s(L_0) \leftarrow \tau^{\mathtt{d}}_s(L_1), \ldots, \tau^{\mathtt{d}}_s(L_m).$$
$$\tau^{\mathtt{o}}_s(L_0) \leftarrow \tau^{\mathtt{o}}_s(L_1), \ldots, \tau^{\mathtt{o}}_s(L_m).$$
$$\tau^{\mathtt{c}}_s(L_0) \leftarrow \tau^{\mathtt{c}}_s(L_1), \ldots, \tau^{\mathtt{c}}_s(L_m).$$

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

$$\neg n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim n_s{:}\mathtt{o}\_p(\overline{x}). \qquad n_s{:}\mathtt{o}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{o}\_cxt(\overline{x}), \sim\neg n_s{:}\mathtt{o}\_p(\overline{x}).$$
$$n_s{:}\mathtt{c}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{c}\_cxt(\overline{x}), \sim\neg n_s{:}\mathtt{c}\_p(\overline{x}).$$
$$\neg n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow \neg n_s{:}\mathtt{c}\_p(\overline{x}). \qquad n_s{:}\mathtt{n}\_p(\overline{x}) \leftarrow n_s{:}\mathtt{c}\_p(\overline{x}).$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$ and $k = arity(p)$.

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

7. Let $p \in \mathit{Pred}^{\mathtt{D}}_s$ be a predicate that is defined as *freely* `negatively closed` in s (that is, $\mathit{mode}^{\mathtt{D}}_s(p) = \mathtt{c}^-$ and $\mathit{context}_s(p) = \mathtt{n/a}$). This case is treated similarly to the previous case with the difference that literals n_s:$\mathtt{o}\_\mathit{cxt}(\overline{x})$ and n_s:$\mathtt{c}\_\mathit{cxt}(\overline{x})$ are eliminated from the generated rules.
8. Let $p \in \mathit{Pred}^{\mathtt{D}}_s$ be a predicate that is defined as `normal` in s (that is, $\mathit{mode}^{\mathtt{D}}_s(p) = \mathtt{n}$). If P_s contains a rule r:

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

then r is translated into the following rule, denoted by $r^{\mathtt{n}}$:

$$\tau^{\mathtt{n}}_s(L_0) \leftarrow \tau^{\mathtt{n}}_s(L_1), \ldots, \tau^{\mathtt{n}}_s(L_m), \sim \tau^{\mathtt{n}}_s(L_{m+1}), \ldots, \sim \tau^{\mathtt{n}}_s(L_n).$$

Rule $r^{\mathtt{n}}$ is added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

The `uses` declarations of rule base s generate rules that respect Definition 19(2) (see also Table 1). In particular:

1. Let $p \in \mathit{Pred}^{\mathtt{U}}_s$ be a predicate that is requested from a rule base $t \in \mathcal{S}$ (i.e. $\mathit{Nam}_t \in \mathit{Import}^{\mathcal{S}}_s(p)$) s.t. $\mathit{least}(\mathit{mode}^{\mathtt{U}}_s(p), |\mathit{mode}^{\mathtt{D}}_t(p)|) = \mathtt{d}$ (see elements `definite` of Table 1). Then, the following rules are generated:

$$\begin{array}{ll}
n_s{:}\mathtt{d}_p(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{d}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{o}_p(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{o}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{c}_p(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{c}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{n}_p(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{n}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}).
\end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = \mathit{arity}(p)$, and $n_t = \mathit{Nam}_t$.

Additionally, the following rules are generated:

$$\begin{array}{ll}
n_s{:}\mathtt{d}_p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{d}_p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{o}_p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{o}_p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{c}_p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{c}_p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{n}_p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{n}_p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}).
\end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = \mathit{arity}(p)$, and $n_t = \mathit{Nam}_t$.

The rules in the first line of the above two sets of rules are added to ELP $P^{\mathtt{d}}_{s,\mathcal{S}}$. The rules in the second line are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$. The rules in the third line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$, and the rules in the fourth line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

2. Let $p \in \mathit{Pred}^{\mathtt{U}}_s$ be a predicate that is requested from a rule base $t \in \mathcal{S}$ (i.e. $\mathit{Nam}_t \in \mathit{Import}^{\mathcal{S}}_s(p)$) s.t. $\mathit{least}(\mathit{mode}^{\mathtt{U}}_s(p), |\mathit{mode}^{\mathtt{D}}_t(p)|) = \mathtt{o}$ (see elements `open` of Table 1). Then, the following rules are generated:

$$\begin{array}{ll}
n_s{:}\mathtt{d}_p(\overline{x}) \leftarrow n_t{:}\mathtt{d}_p(\overline{x}). & \neg n_s{:}\mathtt{d}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d}_p(\overline{x}). \\
n_s{:}\mathtt{o}_p(\overline{x}) \leftarrow n_t{:}\mathtt{o}_p(\overline{x}). & \neg n_s{:}\mathtt{o}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o}_p(\overline{x}). \\
n_s{:}\mathtt{c}_p(\overline{x}) \leftarrow n_t{:}\mathtt{o}_p(\overline{x}). & \neg n_s{:}\mathtt{c}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o}_p(\overline{x}). \\
n_s{:}\mathtt{n}_p(\overline{x}) \leftarrow n_t{:}\mathtt{o}_p(\overline{x}). & \neg n_s{:}\mathtt{n}_p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o}_p(\overline{x}).
\end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

Additionally, the following rules are generated:

$$\begin{array}{ll} n_s{:}\mathtt{d\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d\_}p(\overline{x}). & \neg n_s{:}\mathtt{d\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d\_}p(\overline{x}). \\ n_s{:}\mathtt{o\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{o\_}p(\overline{x}). & \neg n_s{:}\mathtt{o\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o\_}p(\overline{x}). \\ n_s{:}\mathtt{c\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{o\_}p(\overline{x}). & \neg n_s{:}\mathtt{c\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o\_}p(\overline{x}). \\ n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{o\_}p(\overline{x}). & \neg n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o\_}p(\overline{x}). \end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

The rules in the first line of the above two sets of rules are added to ELP $P^{\mathtt{d}}_{s,\mathcal{S}}$. The rules in the second line are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$. The rules in the third line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$, and the rules in the fourth line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

3. Let $p \in Pred^{\mathtt{U}}_s$ be a predicate that is requested from a rule base $t \in \mathcal{S}$ (i.e. $Nam_t \in Import^{\mathcal{S}}_s(p)$) s.t. $least(mode^{\mathtt{U}}_s(p), |mode^{\mathtt{D}}_t(p)|) = \mathtt{c}$ (see elements `closed` of Table 1). Then, the following rules are generated:

$$\begin{array}{ll} n_s{:}\mathtt{d\_}p(\overline{x}) \leftarrow n_t{:}\mathtt{d\_}p(\overline{x}). & \neg n_s{:}\mathtt{d\_}p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d\_}p(\overline{x}). \\ n_s{:}\mathtt{o\_}p(\overline{x}) \leftarrow n_t{:}\mathtt{o\_}p(\overline{x}). & \neg n_s{:}\mathtt{o\_}p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o\_}p(\overline{x}). \\ n_s{:}\mathtt{c\_}p(\overline{x}) \leftarrow n_t{:}\mathtt{c\_}p(\overline{x}). & \neg n_s{:}\mathtt{c\_}p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{c\_}p(\overline{x}). \\ n_s{:}\mathtt{n\_}p(\overline{x}) \leftarrow n_t{:}\mathtt{c\_}p(\overline{x}). & \neg n_s{:}\mathtt{n\_}p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{c\_}p(\overline{x}). \end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

Additionally, the following rules are generated:

$$\begin{array}{ll} n_s{:}\mathtt{d\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{d\_}p(\overline{x}). & \neg n_s{:}\mathtt{d\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{d\_}p(\overline{x}). \\ n_s{:}\mathtt{o\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{o\_}p(\overline{x}). & \neg n_s{:}\mathtt{o\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{o\_}p(\overline{x}). \\ n_s{:}\mathtt{c\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{c\_}p(\overline{x}). & \neg n_s{:}\mathtt{c\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{c\_}p(\overline{x}). \\ n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{c\_}p(\overline{x}). & \neg n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{c\_}p(\overline{x}). \end{array}$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

The rules in the first line of the above two sets of rules are added to ELP $P^{\mathtt{d}}_{s,\mathcal{S}}$. The rules in the second line are added to ELP $P^{\mathtt{o}}_{s,\mathcal{S}}$. The rules in the third line are added to ELP $P^{\mathtt{c}}_{s,\mathcal{S}}$, and the rules in the fourth line are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

4. Let $p \in Pred^{\mathtt{U}}_s$ be a predicate that is requested from a rule base $t \in \mathcal{S}$ (i.e. $Nam_t \in Import^{\mathcal{S}}_s(p)$) s.t. $least(mode^{\mathtt{U}}_s(p), |mode^{\mathtt{D}}_t(p)|) = \mathtt{n}$ (see elements `normal` of Table 1). Then, the following rules are generated:

$$n_s{:}\mathtt{n\_}p(\overline{x}) \leftarrow n_t{:}\mathtt{n\_}p(\overline{x}). \qquad \neg n_s{:}\mathtt{n\_}p(\overline{x}) \leftarrow \neg n_t{:}\mathtt{n\_}p(\overline{x}).$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

Additionally, the following rules are generated:

$$n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow n_t{:}\mathtt{n\_}p(\overline{x}). \qquad \neg n_s{:}\mathtt{n\_}p@n_t(\overline{x}) \leftarrow \neg n_t{:}\mathtt{n\_}p(\overline{x}).$$

where $\overline{x} = \langle ?x_1, ..., ?x_k \rangle$, $k = arity(p)$, and $n_t = Nam_t$.

Generated rules are added to ELP $P^{\mathtt{n}}_{s,\mathcal{S}}$.

Let $s \in \mathcal{S}$. We are now ready to define the ELPs $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$, where $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. In particular, we define: $\Pi^{\mathtt{m}}_{s,\mathcal{S}} = \cup\{P^{\mathtt{x}}_{s',\mathcal{S}} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$.

Below, we define the MWebAS and MWebWFS semantics of a modular rule base $\mathcal{S}$ through the corresponding semantics of the ELPs $\Pi^{\mathtt{d}}_{s,\mathcal{S}}$, $\Pi^{\mathtt{o}}_{s,\mathcal{S}}$, $\Pi^{\mathtt{c}}_{s,\mathcal{S}}$, and $\Pi^{\mathtt{n}}_{s,\mathcal{S}}$, where $s \in \mathcal{S}$. 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 $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ and the answer sets (resp. partial stable models) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$, according to AS (resp. WFSX) semantics is *one-to-one*.

First, we provide the following definition: Let $\mathsf{M} = \{M^{\mathtt{x}}_{s'} \mid \langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$ be a normal (resp. extended) interpretation of a rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. We will denote by N_{M} the 2-valued (resp. 3-valued) interpretation[14] of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ s.t. (i) $N_{\mathsf{M}}(\tau^{\mathtt{x}}_{s'}(L)) = M^{\mathtt{x}}_{s'}(L)$, for all $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$ and $L \in \mathtt{HB}^{\mathcal{S}}_{s} \cup \sim\mathtt{HB}^{\mathcal{S}}_{s}$, (ii) $N_{\mathsf{M}}(\mathtt{domain}(c)) = 1$, for all $c \in \mathtt{HU}_{\mathcal{S}}$, and (iii) $N_{\mathsf{M}}(\mathtt{t}) = 1$ (resp. $N_{\mathsf{M}}(\mathtt{t}) = 1$ and $N_{\mathsf{M}}(\mathtt{u}) = 1/2$).

Theorem 1. Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. It holds that: M is a normal (resp. extended) answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ iff N_{M} is an answer set (resp. partial stable model[15]) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. $\Box$

Proposition 7. Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) \neq \emptyset$. It holds that: $\mathsf{M} = \mathtt{mWF}^{\mathcal{S}}_{s,\mathtt{m}}$ iff N_{M} is the well-founded model (according to WFSX semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. $\Box$

The following Corollary, which follows directly from Definition 24 and Theorem 1, shows that the meaning of a predicate p defined in a rule base $s \in \mathcal{S}$ according to mAS (resp. mWFS) semantics, is given by the literals $\tau^{\mathtt{m}}_{s}(L)$ in $\mathcal{C}_{\mathtt{AS}}\left(\Pi^{\mathtt{m}}_{s,\mathcal{S}}\right)$ (resp. $\mathcal{C}_{\mathtt{WFSX}}\left(\Pi^{\mathtt{m}}_{s,\mathcal{S}}\right)$), where $\mathtt{m} = mode^{\mathtt{D}}_{s}(p)$ and $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$.

Corollary 2. Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let:

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

It holds that:

- $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$ iff $\tau^{\mathtt{m}}_{s}(L) \in \mathcal{C}_{\mathtt{AS}}\left(\Pi^{\mathtt{m}}_{s,\mathcal{S}}\right)$.
- $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$ iff $\tau^{\mathtt{m}}_{s}(L) \in \mathcal{C}_{\mathtt{WFSX}}\left(\Pi^{\mathtt{m}}_{s,\mathcal{S}}\right)$. $\Box$

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, MWebAS is more informative than MWebWFS. However, MWebWFS has better computational properties. Additionally, we show that MWebAS and MWebWFS extends AS and WFS on ELPs.

[14] Here, we include also inconsistent interpretations I of an ELP P s.t. $I(L) = 1$, for all $L \in \mathtt{HB}_{P} \cup \sim\mathtt{HB}_{P}$, where $\mathtt{HB}_{P}$ denotes the *Herbrand Base* of P.

[15] Here, we consider an extended definition of the *partial stable models* of WFSX semantics [53, 3], where inconsistent partial stable models are also allowed.

Proposition 8. Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup \sim\mathtt{HB}_s^{\mathcal{S}}$. It holds that: if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ then $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. □

The following proposition provides the data complexities of MWebAS and MWebWFS 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 MWebAS and MWebWFS semantics of a rule base s w.r.t. a modular rule base $\mathcal{S}$, through appropriately defined ELPs $\Pi_{s,\mathcal{S}}^{\mathtt{d}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{o}}$, $\Pi_{s,\mathcal{S}}^{\mathtt{c}}$, and $\Pi_{s,\mathcal{S}}^{\mathtt{n}}$, evaluated under AS and WFSX semantics, respectively (see Section 6).

Proposition 9. Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup \sim\mathtt{HB}_s^{\mathcal{S}}$. It holds that: (i) the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ is data complete for co-NP and program complete for co-NEXPTIME, and (ii) the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ is data complete for P and program complete for EXPTIME. □

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

Proposition 10. Let $\mathcal{S}$ be a modular rule base. Additionally, let $p \in \mathit{Pred}_s^{\mathtt{D}}$ s.t. $\mathit{mode}_s^{\mathtt{D}}(p) = \mathtt{d}$, and let $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$. It holds that: $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. □

The following proposition shows that MWebAS and MWebWFS semantics extend AS and WFSX semantics on ELPs, respectively. Let P be an ELP. We denote by $\mathcal{C}_{\mathtt{SEM}}(P)$ the set of literals entailed from P under $\mathtt{SEM} \in \{\mathtt{AS}, \mathtt{WFSX}\}$. If P does not have a consistent *answer set* then AS semantics adopts an explosive approach by letting $\mathcal{C}_{\mathtt{AS}}(P) = \mathtt{HB}_P \cup \sim\mathtt{HB}_P$, where $\mathtt{HB}_P$ is the *Herbrand Base* of P[16] [31]. Similarly, if P does not have a consistent *well-founded model* (according to WFSX semantics) then $\mathcal{C}_{\mathtt{WFSX}}(P) = \mathtt{HB}_P \cup \sim\mathtt{HB}_P$ [53].

Proposition 11. Let s be a rule base s.t. $\mathit{Pred}_s^{\mathtt{U}} = \emptyset$ and for all $p \in \mathit{Pred}_s^{\mathtt{D}}$, $\mathit{mode}_s^{\mathtt{D}}(p) = \mathtt{n}$. Let $\mathcal{S} = \{s\}$, let $p \in \mathit{Pred}_s^{\mathtt{D}}$, and let $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$. It holds that: (i) $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $L \in \mathcal{C}_{\mathtt{AS}}(P_s)$, and (ii) $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $L \in \mathcal{C}_{\mathtt{WFSX}}(P_s)$. □

Let $\mathcal{S}$, $\mathcal{S}'$ be modular rule bases. We say that $\mathcal{S}'$ *expands* $\mathcal{S}$ if $\mathcal{S} \subseteq \mathcal{S}'$. 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 12. Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases s.t. $\mathcal{S} \subseteq \mathcal{S}'$ and for all $s \in \mathcal{S}$ and $p \in \mathit{Pred}_s^{\mathtt{U}}$, $\mathit{Import}_s^{\mathcal{S}}(p) = \mathit{Import}_s^{\mathcal{S}'}(p)$. Let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup \sim\mathtt{HB}_s^{\mathcal{S}}$. It holds that: $s \models_{\mathcal{S}}^{\mathtt{SEM}} L$ iff $s \models_{\mathcal{S}'}^{\mathtt{SEM}} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. □

[16] The *Herbrand Base* of an ELP P is the set of ground literals $p(c_1, ..., c_k)$ 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.

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 $\mathcal{S}$ is *extended* by extending the rule bases in $\mathcal{S}$ and by adding to $\mathcal{S}$ more rule bases. Now, a rule base s is extended by extending: (i) the logic program of s, (ii) the defined predicates p of s, along with their scope, defining reasoning mode, and exporting rule base list, and (iii) the used predicates of s, along with their requesting reasoning mode and importing rule base list. In other words, information and sharing of information in $\mathcal{S}$ is increased.

Definition 25 (Extending modular rule bases). Let $\mathcal{S}$, $\mathcal{S}'$ be modular rule bases. We say that $\mathcal{S}'$ *extends* $\mathcal{S}$ ($\mathcal{S} \leq \mathcal{S}'$) iff for all $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}'$:

i. $Nam_s = Nam_{s'}$, $P_s \subseteq P_{s'}$, $Pred^{\mathtt{D}}_s \subseteq Pred^{\mathtt{D}}_{s'}$, $Pred^{\mathtt{U}}_s \subseteq Pred^{\mathtt{U}}_{s'}$,

ii. For all $p \in Pred^{\mathtt{D}}_s$:
   (a) $scope_s(p) \leq scope_{s'}(p)$ and $Export^{\mathcal{S}}_s(p) \subseteq Export^{\mathcal{S}'}_{s'}(p)$,
   (b) $mode^{\mathtt{D}}_s(p) \leq |mode^{\mathtt{D}}_{s'}(p)|$,
   (c) if $|mode^{\mathtt{D}}_s(p)|, |mode^{\mathtt{D}}_{s'}(p)| \in \{\mathtt{o}, \mathtt{c}\}$ then $context_s(p) = context_{s'}(p)$, and

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

The following propositions are used for proving monotonicity of reasoning over the definite and open predicates of a modular rule base $\mathcal{S}$, in the case that $\mathcal{S}$ 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 $s \in \mathcal{S}$ are extended, (ii) some of the internal predicates of a rule base $s \in \mathcal{S}$ are becoming local or global, (iii) some of the local predicates of a rule base $s \in \mathcal{S}$ are becoming global, (iv) the exporting rule base list of some defining predicates of a rule base $s \in \mathcal{S}$ is extended, and (v) the importing rule base list of some used predicates of a rule base $s \in \mathcal{S}$ is also extended. Additionally, new rule bases may be added to $\mathcal{S}$. We refer to this kind of modular rule base extension, as *content upgrade extension*. Proposition 13 shows that the objective literals of definite and open predicates, entailed under `MWebAS` (resp. `MWebWFS`), increase after the content upgrade extension of $\mathcal{S}$ to $\mathcal{S}'$.

Proposition 13 (Content upgrade monotonicity). Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases such that for all $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}'$:

1. $Nam_s = Nam_{s'}$, $P_s \subseteq P_{s'}$, $Pred^{\mathtt{D}}_s \subseteq Pred^{\mathtt{D}}_{s'}$, $Pred^{\mathtt{U}}_s \subseteq Pred^{\mathtt{U}}_{s'}$,
2. For all $p \in Pred^{\mathtt{D}}_s$:
   $scope_s(p) \leq scope_{s'}(p)$, $mode^{\mathtt{D}}_s(p) = mode^{\mathtt{D}}_{s'}(p)$, $context_s(p) = context_{s'}(p)$, $Export^{\mathcal{S}}_s(p) \subseteq Export^{\mathcal{S}'}_{s'}(p)$, and
3. For all $p \in Pred^{\mathtt{U}}_s$:
   $mode^{\mathtt{U}}_s(p) = mode^{\mathtt{U}}_{s'}(p)$ and $Import^{\mathcal{S}}_s(p) \subseteq Import^{\mathcal{S}'}_{s'}(p)$.

Let $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$ s.t. $Nam_s = Nam_{s'}$. Let $p \in Pred^{\mathtt{D}}_s$ s.t. $mode^{\mathtt{D}}_s(p) \in \{\mathtt{d}, \mathtt{o}\}$ and let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$ then $s' \models^{\mathtt{SEM}}_{\mathcal{S}'} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. $\Box$

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 $s \in \mathcal{S}$, respectively, is increasing. Additionally, new rule bases may be added to $\mathcal{S}$. We refer to this kind of modular rule base extension, as *mode upgrade extension*. Proposition 14 shows that the objective literals of definite and open predicates, entailed under `MWebAS` (resp. `MWebWFS`), increase after the mode upgrade extension of $\mathcal{S}$ to $\mathcal{S}'$.

Proposition 14 (Mode upgrade monotonicity). Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases such that for all $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}'$:

1. $Nam_s = Nam_{s'}$, $P_s = P_{s'}$, $Pred_s^{\mathtt{D}} = Pred_{s'}^{\mathtt{D}}$, $Pred_s^{\mathtt{U}} = Pred_{s'}^{\mathtt{U}}$ and
2. For all $p \in Pred_s^{\mathtt{D}}$:
   (a) $scope_s(p) = scope_{s'}(p)$ and $Export_s^{\mathcal{S}}(p) = Export_{s'}^{\mathcal{S}'}(p)$,
   (b) $mode_s^{\mathtt{D}}(p) \leq |mode_{s'}^{\mathtt{D}}(p)|$,
   (d) if $|mode_s^{\mathtt{D}}(p)|$, $|mode_{s'}^{\mathtt{D}}(p)| \in \{\mathtt{o}, \mathtt{c}\}$ then $context_s(p) = context_{s'}(p)$.
3. For all $p \in Pred_s^{\mathtt{U}}$:
   $mode_s^{\mathtt{U}}(p) \leq mode_{s'}^{\mathtt{U}}(p)$ and $Import_s^{\mathcal{S}}(p) = Import_{s'}^{\mathcal{S}'}(p)$.

Let $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$ s.t. $Nam_s = Nam_{s'}$. Let $p \in Pred_s^{\mathtt{D}}$ s.t. $mode_s^{\mathtt{D}}(p), mode_{s'}^{\mathtt{D}}(p) \in \{\mathtt{d}, \mathtt{o}\}$ and let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models_{\mathcal{S}}^{\mathtt{SEM}} L$ then $s' \models_{\mathcal{S}'}^{\mathtt{SEM}} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. $\square$

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 13 and 14.

Theorem 2 (Monotonicity of definite and open predicates under modular rule base extension). Let $\mathcal{S}$, $\mathcal{S}'$ be modular rule bases s.t. $\mathcal{S} \leq \mathcal{S}'$. Let $s \in \mathcal{S}$ and let $s' \in \mathcal{S}'$ s.t. $Nam_s = Nam_{s'}$. Let $p \in Pred_s^{\mathtt{D}}$ s.t. $mode_s^{\mathtt{D}}(p), mode_{s'}^{\mathtt{D}}(p) \in \{\mathtt{d}, \mathtt{o}\}$ and let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models_{\mathcal{S}}^{\mathtt{SEM}} L$ then $s' \models_{\mathcal{S}'}^{\mathtt{SEM}} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. $\square$

7.2 c-Stratified Predicates over Modular Rule Bases

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. In this subsection, we define the `c`-stratified[17] predicates of s w.r.t. $\mathcal{S}$. Intuitively, the definition of `c`-stratified predicates adapts the definition of stratified normal programs [6] 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. $\mathcal{S}$ then inference on p from s w.r.t. $\mathcal{S}$ is fully determined. In particular, for any ground atom $p(\overline{c}) \in [p]_{\mathcal{S}}$, it holds that:
  $s \models_{\mathcal{S}}^{\mathtt{SEM}} p(\overline{c})$ or $s \models_{\mathcal{S}}^{\mathtt{SEM}} \neg p(\overline{c})$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.
– If a predicate p (i) is defined positively or negatively closed in s w.r.t. context cxt, and (ii) is `c`-stratified in s w.r.t. $\mathcal{S}$ then inference on p from s w.r.t. $\mathcal{S}$ is fully determined within context cxt. In particular, for any ground atom $p(\overline{c}) \in [p]_{\mathcal{S}}$, it holds that:
  $s \models_{\mathcal{S}}^{\mathtt{SEM}} p(\overline{c})$, or $s \models_{\mathcal{S}}^{\mathtt{SEM}} \neg p(\overline{c})$, or $s \models_{\mathcal{S}}^{\mathtt{SEM}} {\sim} cxt(\overline{c})$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.

[17] The prefix `c` in the term `c`-*stratified* stands for `closed` because this notion applies to positively or negatively closed predicates, only.

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

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

Let s be a rule base, we define the *extended predicates* of s, as follows: $Pred_s^{\neg} = Pred_s \cup \{\neg p \mid p \in Pred_s\}$. Let p be a predicate. We define $\neg(\neg p) = p$ and $|\neg p\,| = p$. Additionally, we define $\neg Pred = \{\neg p \mid p \in Pred\}$. Let $\mathcal{S}$ be a modular rule base. We define $N_{\mathcal{S}} = \{\langle \mathtt{m}, s, x\rangle \mid \mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}\},\ s \in \mathcal{S}, \text{ and } x \in Pred_s^{\neg}\}$.

We are now ready to define the *direct* `c`*-dependence*[18] relationship between triples $\langle \mathtt{m}, s, x\rangle \in N_{\mathcal{S}}$. Intuitively, $\langle \mathtt{m}, s, x\rangle$ *directly* `c`*-depends on* $\langle \mathtt{m}', s', x'\rangle$, if (i) the definition of x in rule base s in reasoning mode $\mathtt{m}$ depends on the definition of x' in rule base s' in reasoning mode $\mathtt{m}'$, or (ii) $s = s'$, $\mathtt{m} = \mathtt{m}' = \mathtt{c}$, 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 26 (Direct `c`-dependence). Let $\mathcal{S}$ be a modular rule base and let $s, s' \in \mathcal{S}$. Additionally, let $\langle \mathtt{m}, s, x\rangle$, $\langle \mathtt{m}', s', x'\rangle \in N_{\mathcal{S}}$. We say that $\langle \mathtt{m}, s, x\rangle$ *directly* `c`*-depends on* $\langle \mathtt{m}', s', x'\rangle$ w.r.t. $\mathcal{S}$ (denoted by $\langle \mathtt{m}, s, x\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{m}', s', x'\rangle$) iff:

1. It holds that:
   (a) there exists $r \in P_s$ s.t. $pred^{\neg}(Head_r) = x$, and there exists $L' \in Body_r^{+}$ s.t.:
      i. $pred^{\neg}(L') = x'$,
      ii. if $qual(L') \neq \mathtt{n/a}$ then $Nam_{s'} = qual(L')$,
      iii. if $s \neq s'$ then $Nam_{s'} \in Import_s^{\mathcal{S}}(pred(L'))$,
   (b) if $s = s'$ then $\mathtt{m}' = least(\mathtt{m}, mode_s^{\mathtt{D}}(|x'|))$, and
   (c) if $s \neq s'$ then $\mathtt{m}' = least(\mathtt{m}, mode_s^{\mathtt{U}}(|x'|), |mode_{s'}^{\mathtt{D}}(|x'|)|)$, or
2. It holds that:
   (a) $s = s'$, $\mathtt{m} = \mathtt{m}' = \mathtt{c}$, and $x' = \neg x$,
   (b) if $x \in Pred$ then $mode_s^{\mathtt{D}}(x) = \mathtt{c}^{-}$,
   (c) if $x \in \neg Pred$ then $mode_s^{\mathtt{D}}(|x|) = \mathtt{c}^{+}$. □

It is easy to see that if $\langle \mathtt{m}, s, x\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{m}', s', x'\rangle$ and $\mathtt{m}' = \mathtt{c}$ then $\mathtt{m} = \mathtt{c}$.

Example 22. Consider the modular rule base $\mathcal{S} = \{s_1, s_2\}$ 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, $\neg$`gov:Enter`$\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle$`d`, s_1, `sec:Suspect` $\rangle$ and $\langle$`c`, s_2, `gov:Enter` $\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}}$ $\langle$`c`, s_2, $\neg$`gov:Enter`$\rangle$. □

Let $\mathcal{S}$ be a modular rule base. The `c`*-dependency graph* of $\mathcal{S}$ is built by linking triples $\langle \mathtt{m}, s, x\rangle \in N_{\mathcal{S}}$.

Definition 27 (`c`-dependency graph of a MRB). Let $\mathcal{S}$ be a modular rule base.

[18] The prefix `c` in the term `c`-*dependence* stands for `closed` and is used to differentiate this kind of dependency from other kinds of dependencies.

Rule base s_1

⟨*http://security.int*⟩

```
defines local open sec:Person.
defines glocal open sec:Suspect.

sec:Person(Anne).
sec:Person(Boris).
sec:Suspect(Peter).
```

Rule base s_2

⟨*http://gov.countryZ*⟩

```
defines local negClosed gov:Enter
        w.r.t. sec:Person.
uses definite sec:Suspect from
        ⟨http://security.int⟩.
uses definite sec:Person from
        ⟨http://security.int⟩.

¬ gov:Enter(?p) ← sec:Suspect(?p).
```

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

- The c-*dependency graph* of $\mathcal{S}$ is $DG^{\mathtt{c}}_{\mathcal{S}} = (N_{\mathcal{S}}, E_{\mathcal{S}})$, where: $E_{\mathcal{S}} = \{\langle n, n' \rangle \in N_{\mathcal{S}} \times N_{\mathcal{S}} \mid n \leftarrow^{\mathtt{c}}_{\mathcal{S}} n'\}$.
- A node $n \in N_{\mathcal{S}}$ c-*depends* on a node $n' \in N_{\mathcal{S}}$ w.r.t. $\mathcal{S}$ (denoted by $n \dashleftarrow^{\mathtt{c}}_{\mathcal{S}} n)$ iff there is a path in $DG^{\mathtt{c}}_{\mathcal{S}}$ from n to n'. □

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

$$\mathtt{c}\text{-}D^{\mathcal{S}}_{x,s} = \{\langle \mathtt{c}, s, x \rangle\} \cup \{\langle \mathtt{m}', s', x' \rangle \in N_{\mathcal{S}} \mid \langle \mathtt{c}, s, x \rangle \dashleftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{m}', s', x' \rangle\}.$$

Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let p be a predicate defined in s in closed reasoning mode. Intuitively, p is c-stratified in s w.r.t. $\mathcal{S}$ 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 $\mathtt{c}\text{-}D^{\mathcal{S}}_{x,s}$ to $\mathbb{N}$ s.t.:
   for all $\langle \mathtt{m}', s', x' \rangle \in \mathtt{c}\text{-}D^{\mathcal{S}}_{x,s}$,
   (a) if $\langle \mathtt{m}', s', x' \rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{m}'', s'', x'' \rangle$ then the level of $\langle \mathtt{m}'', s'', x'' \rangle$ is less than or equal to the level of $\langle \mathtt{m}', s', x' \rangle$, and
   (b) 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 $\langle \mathtt{c}, s', \neg x' \rangle$ is less than the level of $\langle \mathtt{c}, s', x' \rangle$.

Definition 28 (c-stratified predicate w.r.t. a MRB). Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $p \in Pred^{\mathtt{D}}_s$ s.t. $mode^{\mathtt{D}}_s(p) \in \{\mathtt{c}^+, \mathtt{c}^-\}$. Let $x = p$, if $mode^{\mathtt{D}}_s(p) = \mathtt{c}^-$, and let $x = \neg p$, if $mode^{\mathtt{D}}_s(p) = \mathtt{c}^+$. We say that p is c-*stratified* in s w.r.t. $\mathcal{S}$ iff:

1. For all $\langle \mathtt{m}', s', x' \rangle \in \mathtt{c}\text{-}D^{\mathcal{S}}_{x,s}$, it holds that $\mathtt{m}' \in \{\mathtt{d}, \mathtt{c}\}$.
2. There exists a mapping $level : \mathtt{c}\text{-}D^{\mathcal{S}}_{x,s} \to \mathbb{N}$ s.t.:
   For all $\langle \mathtt{m}', s', x' \rangle \in \mathtt{c}\text{-}D^{\mathcal{S}}_{x,s}$, and for all $\langle \mathtt{m}'', s'', x'' \rangle \in N_{\mathcal{S}}$ s.t. $\langle \mathtt{m}', s', x' \rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{m}'', s'', x'' \rangle$:
   (a) $level(\langle \mathtt{m}'', s'', x'' \rangle) \leq level(\langle \mathtt{m}', s', x' \rangle)$, and
   (b) if $\mathtt{m}' = \mathtt{m}'' = \mathtt{c}$, $s' = s''$, and $x'' = \neg x'$ then:
      i. if $x' \in Pred$, $mode^{\mathtt{D}}_{s'}(x') = \mathtt{c}^-$ then $level(\langle \mathtt{c}, s', \neg x' \rangle) < level(\langle \mathtt{c}, s', x' \rangle)$.
      ii. if $x' \in \neg Pred$, $mode^{\mathtt{D}}_{s'}(|x'|) = \mathtt{c}^+$ then $level(\langle \mathtt{c}, s', \neg x' \rangle) < level(\langle \mathtt{c}, s', x' \rangle)$.

Example 23. Consider the modular rule base $\mathcal{S} = \{s_1, s_2\}$ of Figure 2 and let $x =$gov:Enter. It holds that: c-$D^{\mathcal{S}}_{x,s} = \{\langle$d, s_1, sec:Suspect $\rangle, \langle$c, s_2, $\neg$gov:Enter$\rangle\}$. Note that there exists a mapping *level* :c-$D^{\mathcal{S}}_{x,s} \rightarrow I\!N$ that satisfies the conditions of Definition 28(2). Therefore, gov:Enter is c-stratified in s_2 w.r.t. $\mathcal{S}$. □

Proposition 15. Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let $p \in Pred^{\mathtt{D}}_s$ s.t. p is c-stratified in s w.r.t. $\mathcal{S}$ and let $L = p(c_1, ..., c_n)$, where $c_i \in \mathtt{HU}_{\mathcal{S}}$, for $i = 1, ..., n$. Let $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.

1. If p is freely (positively or negatively) closed in s then: $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$ or $s \models^{\mathtt{SEM}}_{\mathcal{S}} \neg L$.
2. If p is (positively or negatively) closed in s w.r.t. context cxt then: $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$, or $s \models^{\mathtt{SEM}}_{\mathcal{S}} \neg L$, or $s \models^{\mathtt{SEM}}_{\mathcal{S}} {\sim}cxt(c_1, ..., c_n)$. □

Let $\mathcal{S}$ be a modular rule base. The following proposition shows that the MWebAS and MWebWFS semantics of the c-stratified predicates of a rule base $s \in \mathcal{S}$ are equivalent in the case that $\mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s) \neq \{\}$.

Proposition 16. Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$ s.t. $\mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s) \neq \{\}$. Additionally, let $p \in Pred^{\mathtt{D}}_s$ s.t. p is c-stratified in s w.r.t. $\mathcal{S}$, and let $L \in [p]_{\mathcal{S}} \cup {\sim}[p]_{\mathcal{S}}$. It holds that: $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$ iff $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$. □

Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. As the following example shows the result of Proposition 16 does not hold if $\mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s) = \{\}$.

Example 24. Consider the modular rule base $\mathcal{S} = \{s\}$ of Figure 3. Note that q is c-stratified in s w.r.t. $\mathcal{S}$, whereas p is not. The latter is true because $\langle \mathtt{c}, s, \neg p\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{c}, s, p\rangle$ and $\langle \mathtt{c}, s, p\rangle \leftarrow^{\mathtt{c}}_{\mathcal{S}} \langle \mathtt{c}, s, \neg p\rangle$. Additionally, note that $\mathcal{M}^{\mathtt{AS}}_{\mathtt{c},\mathcal{S}}(s) = \{\}$, while $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{c},\mathcal{S}}(s) \neq \{\}$. Thus, $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$, for all $L \in [\mathtt{ex:q}]_{\mathcal{S}} \cup {\sim}[\mathtt{ex:q}]_{\mathcal{S}}$, whereas $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$, just for $L \in \{\neg$ ex:q(d), $\sim$ ex:q(d), $\sim\neg$ ex:q(c), ex:q(c)$\}$. □

Rule base s

$\langle http://example.org \rangle$
defines local negClosed ex:p. defines local negClosed ex:q.
¬ex:p(c) ← ex:p(c). ¬ 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. $\mathcal{S}$, the addition of a new rule base s' to $\mathcal{S}$ will require the conditions of Definition 28 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. $\mathcal{S}'$ with only local checks.

8 Related Work

Below we review related work and compare it with our `MWeb` framework and semantics. A detailed comparison of the `MWebAS` and `MWebWFS` semantics of modular rule bases, as presented here, with an older version [4] is provided at the end of this Section.

The concept of local CWAs was first introduced in [26]. In that work, the semantics of a local CWA about a sentence Φ appearing in an information source s is that: for all variable substitutions θ, if the ground sentence $\Phi\theta$ is true in the world then $\Phi\theta$ is entailed by s. Thus, any ground sentence $\Phi\theta$ that is not entailed by s is known to be false. Formally: $LCWA(\Phi) \equiv (s \models \Phi\theta) \vee (s \models \neg\Phi\theta)$. As seen in Proposition 15, this notion of local CWA also appears in our framework, in the case that a predicate p is declared freely closed in a rule base s and p is `c`-stratified in s w.r.t. a modular rule base $\mathcal{S}$, where θ is a variable substitution from $\mathtt{HU}_{\mathcal{S}}$.

In [36], local CWAs, with the semantics defined in [26], are applied for agent planning in the Semantic Web. In particular, [36] considers independent information sources over the web, containing (i) knowledge, expressed in DAML+OIL [38] or the SHOE language [35], and (ii) explicit local CWAs. If an agent needs information about a predicate p that is not contained in its knowledge base then p is passed to the Semantic Web Mediator of the agent. The Semantic Web Mediator queries relevant information sources about p and stops if: (i) an answer is found, or (ii) the local CWAs of a source s determine that s has complete knowledge about p. In that work, strong negation is not considered and modularity issues are rather trivial, as information sources do not interact.

A form of local CWA w.r.t. a context is proposed in [18], where the local CWA is applied on the predicates of a *single* data source s, containing only positive facts. In that work, a context is a first-order formula over the predicates of s. The semantics of the proposed local CWA syntax is defined in first-order logic. Rules, strong negation, and modularity issues are not considered, in that work.

An alternative proposal for local CWAs is present in the `dlvhex` system [25]. This answer-set programming system has features, like high-order atoms and external atoms, which are very flexible. For instance, assuming that a generic external atom $KB[C](X)$ is available for querying a concept C in a knowledge base KB then a CWA can be stated as follows: $C'(X) \leftarrow concept(C), concept(C'), cwa(C, C'), o(X),$ $\sim KB[C](X)$, where $concept(C)$ is a predicate which holds for all concepts C, the predicate $cwa(C, C')$ states that C' is the complement of C under the closed world assumption, and $o(X)$ is a predicate that holds for all individuals occurring in KB. Strong negation and modularity issues are not considered, in that work.

The combination of open-world and closed-world reasoning, in the same framework, is also proposed in [5], where the stable model semantics of Extended RDF (ERDF) ontologies is developed. Intuitively, an ERDF ontology is the combination of (i) an ERDF graph G containing (implicitly existentially quantified) positive and negative information, and (ii) an ERDF program P containing derivation rules, with possibly all connectives $\sim$, $\neg$, $\supset$, $\wedge$, $\vee$, $\forall$, $\exists$ in the body of a rule, and strong negation $\neg$ in the head of a rule. However, modularity issues are not considered there, and local CWAs and OWAs are not declared w.r.t. a context.

Flora-2 [70] is a rule-based object-oriented knowledge base system for reasoning with semantic information on the Web. It is based on F-logic [42] and supports metaprogramming, non-monotonic multiple inheritance, logical database updates, encapsulation, modules with dynamically assigned content, and qualified literals. Module indicators in qualified literals can be module names or variables that get bound to a module name at run time. In Flora-2, reasoning mode and predicate scope issues are ignored. Additionally, strong negation is not supported. Simple literals appearing in a file, that is loaded to a module, are assumed to be qualified by the module name. The semantics of a modular rule base $\mathcal{S}$ is defined, based on the F-logic semantics [42] of an equivalent rule base with no modules. In particular, each qualified atom $subject[predicate \rightarrow object]@Nam_s$ (where Nam_s is a module name) is translated to $subject[predicate\#Nam_s \rightarrow object]$, where $predicate\#Nam_s$ is a new predicate name.

TRIPLE [67] is a rule language for the Semantic Web that supports modules (called, *models* there), qualified literals, and dynamic module transformation. Its syntax is based on F-Logic [42] and supports a fragment of RDFS and first-order logic. All variables must be explicitly quantified, either existentially or universally. Arbitrary formulas can be used in the body, while the head of the rules is restricted to atoms or conjunctions of molecules. Module indicators in qualified literals can be module names, variables, or skolem functions, as well as conjunction and difference of module indicators. However, the latter two cases do not add expressive power, as they can be equivalently defined through qualified literal conjunctions and the use of weak negation. The semantics of a modular rule base is defined based on the *well-founded semantics* (`WFS`) [29] of an equivalent logic program. In that work, reasoning mode, predicate scope, and visibility issues are ignored. Additionally, strong negation is not supported.

In [55], a modularity framework for rule bases is proposed and its `AS` semantics is defined. However, in that work, the dependency graph G between the rule bases of a modular rule base $\mathcal{S}$ (formed based on the rule bases' `import` statements) should be acyclic, facilitating distributed evaluation. The answer sets of a module $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined based on the answer sets of the modules that are lower than s in the dependency graph G. In that work, reasoning mode and predicate scope issues are ignored. Additionally, strong negation is not supported. Further, it is assumed that exported predicates are provided to any requesting rule base and imported predicates may be requested, through qualified literals, from any providing rule base.

Another modularity framework for rule bases is proposed in [54], where weakly negated rule literals should be qualified and depend (directly or indirectly) on qualified literals, only. In that work, reasoning mode, predicate scope, and visibility issues are ignored. Additionally, strong negation is not supported. The *contextually bounded* `AS` and *contextually bounded* `WFS` semantics of a modular rule base $\mathcal{S}$ are defined, through the `AS` and `WFS` semantics of an equivalent logic program $\mathcal{S}_{CB}$. $\mathcal{S}_{CB}$ consists of the rules of each rule base $s \in \mathcal{S}$ (called *contexts*, there) union the rules $p@Nam_s(t_1, ..., t_n) \leftarrow Body$, where $p(t_1, ..., t_n) \leftarrow Body$ is a rule defined in a rule base $s \in \mathcal{S}$. Another proposal made in [54] is to qualify all simple atoms appearing in a rule base s by the name of s. The resulting rules union the original rules of each rule base $s \in \mathcal{S}$ form a normal logic program $\mathcal{S}_{CC}$. Then, the *contextually closed* `AS` and *contextually closed* `WFS` semantics of a modular rule base $\mathcal{S}$ are defined through the `AS` and `WFS` semantics of $\mathcal{S}_{CC}$.

Modularity for rule bases is also considered in [15], where the *contextual* `AS` and the *contextual* `WFS` semantics of a modular rule base $\mathcal{S}$ are defined model-theoretically. However, in that work, reasoning mode, predicate scope, and visibility issues are ignored. Simple literals appearing in a rule base s (called *context*, there) are assumed to be qualified by the name of s. Intuitively, we can say that, if (i) all predicates of the rule bases in $\mathcal{S}$ are defined in `normal` reasoning mode, (ii) all literals appearing in the body of the rules of the rule bases in $\mathcal{S}$ are qualified, (iii) predicate scope and visibility issues are ignored, and (iv) $D^{\mathcal{S}}_{s,\mathtt{n}} = \mathcal{S}$ then the `MWebAS` and *contextual* `AS` semantics of a rule base $s \in \mathcal{S}$ coincide. However, this property is not true for `MWebWFS` and *contextual* `WFS`. Indeed, in contrast to `MWebWFS`, contextual `WFS` is not coherent. Further, we would like to note that since in our theory, in general $D^{\mathcal{S}}_{s,\mathtt{m}} \subseteq \mathcal{S}$, for an $s \in \mathcal{S}$ and $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, it is possible that s has a normal answer set in reasoning mode `m` w.r.t. $\mathcal{S}$, even though $\mathcal{S}$ does not have a *stable contextual model*, according to *contextual* `AS` semantics.

We want to note that all modularity frameworks in [55, 54, 15] achieve monotonicity of reasoning in the case that a modular rule base $\mathcal{S}$ is *expanded* with additional rule bases (while original rule bases in $\mathcal{S}$ remain the same). Our framework also achieves this kind of monotonicity, as expressed in Proposition 12. However, our framework achieves also a more general kind of monotonicity for `global` predicates that is described in Theorem 2.

In [7], a general framework for modules in rule-based languages is proposed, which does not adhere to a particular rule language and semantics. Each module consists of a set of private rules, a set of public rules, and a set of dependency relationships between a body part of a rule and a head part of another. The authors propose a single algebraic operation, applied to considered modules, until a single module is derived. This is the *scoped import* operation $s \rtimes_S s'$, where S is a set of pointers to body parts of rules in s'. In particular, $s \rtimes_S s'$ makes a copy s'' of s and transfers all the rules of s' to the private rules of s'', while dependency relationships are readjusted. For our case, where deductive rules are considered, the dependency relationships between the rule parts of a module are derived dynamically, based on predicate names. Note that, as scoped import transfers all rules from the operand modules to the resulting module, the existence of a global model is a requirement. Further, in that work, rules (and not defined predicates) are declared as `public` or `private`.

A framework for modular logic programs is also considered in [23], where each module is seen as a generalized quantifier and is associated with a logic program, a set of input predicates, and a single output predicate. The output predicate of a module s actually corresponds to a `local` predicate in our framework, while other predicates defined in s correspond to `internal` predicates. Similarly, the input predicates of a module s correspond to `used` predicates in our framework. A module can be called from a main logic program or from other modules. However, the respective *call graph* between modules should be acyclic. Weak negation is considered and defined semantics extends the stable model semantics [30]. However, strong negation is not considered, reasoning mode issues are ignored, while the visibility mechanism is simple.

A related direction of research is on modularity frameworks for Description Logic (DL) ontologies [8] (for an overview and qualitative comparison of these frameworks, see [33, 9]). In *E-connections* for DLs [43], a set of DL ontologies $s_1, ..., s_n$ with disjoint vocabularies is connected via a set of *link relations* $\mathcal{E} = \bigcup_{i,j \leq n} E_{i,j}$. Link relations in $E_{i,j}$ are used in s_i, for forming the concepts: $\exists E_{i,j}.C_j$ and $\forall E_{i,j}.C_j$, where C_j is a concept of s_j. In *Distributed Description Logics* (DDLs) [13, 66], a set of DL ontologies $s_1, ..., s_n$ with disjoint vocabularies is connected via (i) a set of *bridge rules* of the form $C_i \stackrel{\sqsubseteq}{\rightarrow} C_j$ or $C_i \stackrel{\sqsupseteq}{\rightarrow} C_j$, where C_i and C_j are concepts of s_i and s_j, respectively, (ii) a set of *partial individual correspondences* $a_i \mapsto b_j$, where a_i and b_j are objects of s_i and s_j, respectively, and (iii) a set of *complete individual correspondences* $a_i \stackrel{=}{\mapsto} \{b_j^1, ..., b_j^n\}$, where a_i is an object of s_i and $b_j^1, ..., b_j^n$ are objects of s_j. The problem of local inconsistency polluting the inferences of all modules in a modular representation is handled in the version of DDL [66] through a special interpretation, called *hole*, whose role is to interpret even inconsistent local T-boxes. In our work, we achieve the same result since the semantics of a rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ depends only on the pairs of rules bases s' and reasoning modes $\mathtt{x}$, $\langle s', \mathtt{x} \rangle$, in $D^{\mathcal{S}}_{s',\mathtt{x}}$ (and not on all pairs in $\mathcal{S} \times \{\mathtt{d},\mathtt{o},\mathtt{c},\mathtt{n}\}$). In *Package-based Description Logics* (P-DL) [9], a set of DL ontologies $s_1, ..., s_n$ (called *packages*) is connected through the use of common terms. Each term and axiom is associated with a single *home package*. A package s_i may use *foreign terms*, that is terms whose *home* is another package s_j.

Despite their differences, modular DL ontologies and MWeb modular rule bases share some common goals, such as encapsulation, localized semantics, partial knowledge reuse, and directed semantic relations. Yet, there exist some major differences between the two approaches. Modular rule bases share a common Herbrand Universe, whereas each DL module has its own local domain of interpretation. Additionally, although DL ontologies are more expressive than rules in certain aspects [34], rules provide for more expressive forms of module interconnection. Moreover, MWeb modular rule bases provide full support for negation (weak and strong).

Finally, we would like to mention a general framework for multi-context reasoning, proposed in [14], that allows to combine arbitrary monotonic and non-monotonic logics. Information flow between the different contexts is achieved through a set of nonmonotonic bridge rules. In that work, several notions for acceptable belief states for the multicontext system are investigated.

As a fair account, we should say that though many of the previously described works do not support strong negation, this can be added as an easy extension of their theories, for Answer Set based semantics. The treatment of strong negation in well-founded based semantics is not so easy and immediate.

8.1 Comparison with older version of the MWebAS and MWebWFS semantics of modular rule bases

Below, we compare the MWebAS and MWebWFS semantics of modular rule bases, as presented here, with an older version [4]. In [4], a normal (resp. extended) interpretation of a modular rule base $\mathcal{S}$[19] is defined as a set $\mathsf{M} = \{M^{\mathtt{x}}_{s'} \mid s' \in \mathcal{S},\ \mathtt{x} \in \{\mathtt{d},\mathtt{o},\mathtt{c},\mathtt{n}\}\}$, where $M^{\mathtt{x}}_{s'}$ is a simple normal (resp. extended) interpretation of rule base s' w.r.t. $\mathcal{S}$. The minimal model of $\mathcal{S}$ w.r.t. a normal (resp. extended) interpretation of $\mathcal{S}$, N, denoted by $least(\mathcal{S}, \mathsf{N})$, is defined as the least (w.r.t. $\leq_{\mathtt{t}}$) normal (resp. extended) interpretation of $\mathcal{S}$, $\mathsf{M} = \{M^{\mathtt{x}}_{s'} \mid s' \in \mathcal{S}$ and $\mathtt{x} \in \{\mathtt{d},\mathtt{o},\mathtt{c},\mathtt{n}\}\}$, that satisfies the conditions of Definition 19, with the difference that: (i) in 19.1 and 19.2(b), symbol $=$ is replaced by $\geq$, and (ii) in 19.2(a) and 19.2(b), expression $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p), |mode^{\mathtt{D}}_{s''}(p)|)$ is replaced by $\mathtt{y} = least(\mathtt{x}, mode^{\mathtt{U}}_{s'}(p))$. Then, a normal (resp. extended) answer set

[19] Note that in [4], a normal (resp. extended) interpretation is defined for the whole modular rule base, and not for a single rule base.

of $\mathcal{S}$ is defined as a normal (resp. extended) interpretation of $\mathcal{S}$, M, satisfying the condition that $\mathsf{M} = \mathit{least}(\mathcal{S}, \mathsf{M})$ (resp. $\mathsf{M} = \mathit{Coh}(\mathit{least}(\mathcal{S}, \mathsf{M}))$).

As the following two examples demonstrate, it is possible that a rule base $s \in \mathcal{S}$ has a consistent normal (resp. extended) answer set at a reasoning mode $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ w.r.t. $\mathcal{S}$, even though (in the old version), (i) in every normal (resp. extended) answer set M of $\mathcal{S}$, $M_s^{\mathtt{x}}$ is inconsistent, or $\mathcal{S}$ has no normal (resp. extended) answer set.

Example 25. Consider the modular rule base $\mathcal{S} = \{s_1, s_2\}$ of Figure 4. Note that, there is a single (consistent) normal (resp. extended) answer set of s_1 in reasoning mode $\mathtt{n}$ w.r.t. $\mathcal{S}$. Thus, $s_1 \models_{\mathcal{S}}^{\mathtt{mAS}} p(c)$ and $s_1 \not\models_{\mathcal{S}}^{\mathtt{mAS}} \neg p(c)$. However, according to the old version of $\mathtt{MWeb}$ semantics, in the single normal (resp. extended) answer set of S, M, it holds that $M_{s_1}^{\mathtt{n}}$ is inconsistent. Thus, in the old version, it holds that $s_1 \models_{\mathcal{S}}^{\mathtt{SEM}} p(c)$ and $s_1 \models_{\mathcal{S}}^{\mathtt{SEM}} \neg p(c)$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. □

Rule base s_1	**Rule base** s_2
$\langle \mathit{http://example1.org} \rangle$	$\langle \mathit{http://example2.org} \rangle$
`defines local normal ex:p.` `uses normal ex:q.`	`defines global definite ex:q` `defines local normal ex:r`

`p(c) ← q(c).`	`q(c).`
	`¬ r(c).`
	`r(c).`

Fig. 4. A modular rule base $\mathcal{S}$

Example 26. Consider again the modular rule base $\mathcal{S} = \{s_1, s_2\}$ of Figure 4, with the difference that the fact "`r(c).`" in rule base s_2 is replaced by the rule "`r(c)` $\leftarrow \sim$ `q(d).`". Note that, there is a single (consistent) normal (resp. extended) answer set of s_1 in reasoning mode $\mathtt{n}$ w.r.t. $\mathcal{S}$. Thus, $s_1 \models_{\mathcal{S}}^{\mathtt{mAS}} p(c)$ and $s_1 \not\models_{\mathcal{S}}^{\mathtt{mAS}} \neg p(c)$. However, according to the old version of $\mathtt{MWeb}$ semantics, there is no normal (resp. extended) answer set of S. Thus, in the old version, it holds that $s_1 \models_{\mathcal{S}}^{\mathtt{SEM}} p(c)$ and $s_1 \models_{\mathcal{S}}^{\mathtt{SEM}} \neg p(c)$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$. Similar is the case if the fact "`r(c).`" in rule base s_2 is replaced by the rule "`r(c)` $\leftarrow \sim$ `r(c).`". □

The old version of `MWeb` semantics achieves monotonicity of reasoning in the case that a modular rule base $\mathcal{S}$ is expanded, 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. However, it does not achieve monotonicity of reasoning for `global` predicates in the more general case of modular rule base extension, as expressed in Theorem 2. This is demonstrated in the following example.

Example 27. Consider the modular rule base $\mathcal{S} = \{s\}$ of Figure 5 at times t and $t + 1$[20]. Note that, at time t, modular rule base $\mathcal{S}$ has a single (consistent) normal answer set (according to the old version of `MWebAS` semantics), M, with $M_s^{\mathtt{o}}(\mathtt{p(c)}) = 1$. Thus, at time t, it holds $s \models_{\mathcal{S}}^{\mathtt{mAS}} \mathtt{p(c)}$. However, at time $t + 1$, modular rule base

[20] Note that at time $t + 1$, an additional rule is added to rule base s.

$\mathcal{S}$ has two (consistent) normal answer sets (according to the old version of `MWebAS` semantics), M and N, with $M^{\mathsf{o}}_s(\texttt{p(c)}) = 1$ and $N^{\mathsf{o}}_s(\neg\texttt{p(c)}) = 1$. Thus, at time t, it holds $s \not\models^{\mathtt{mAS}}_{\mathcal{S}}$ `p(c)`. $\square$

Rule base s at time t	**Rule base s at time $t+1$**
⟨*http://example.org*⟩	⟨*http://example.org*⟩
`defines global open ex:p.` `defines local normal ex:r.`	`defines global open ex:p.` `defines local normal ex:r.`

`r(c)` $\leftarrow \sim$ `r(c).`
`r(c)` $\leftarrow$ `p(c).`
	`r(c)` $\leftarrow \sim$ `r(c).`
	`r(c)` $\leftarrow$ `p(c).`
	`r(c)` $\leftarrow \neg$ `p(c).`

Fig. 5. Modular rule base $\mathcal{S} = \{s\}$ at times t and $t+1$

9 Conclusions

In this paper, we presented a principled framework for modular web rule bases, called `MWeb`. According to this framework, each predicate p defined in a rule base s is characterized by its defining reasoning mode (`definite`, `open`, `positively closed`, `negatively closed`, or `normal`), scope (`global`, `local`, or `internal`), and exporting rule base list. Each predicate p used in a rule base s is characterized by its requesting reasoning mode (`definite`, `open`, `closed`, or `normal`), and importing rule base list. For legal `MWeb` modular rule bases $\mathcal{S}$, the `MWebAS` and `MWebWFS` semantics of each $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ are defined model-theoretically. These semantics extend the `AS` [31, 32] and `WFSX` [53, 3, 1] semantics on ELPs, respectively, keeping all of their semantical and computational characteristics. Additionally, the `MWebAS` and `MWebWFS` semantics of a rule base $s \in \mathcal{S}$ w.r.t. $\mathcal{S}$ can be defined (equivalently), through the `AS` and `WFXS` semantics of a set of ELPs, one for each $s \in \mathcal{S}$ and reasoning mode `d`, `o`, `c`, and `n`.

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

- reasoning in four modes, `definite`, `open`, `closed`, and `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 `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 `MWebWFS` semantics, declaring a predicate p as `open` or requesting a predicate p in open reasoning mode, we do not obtain more entailments for p than declaring p as `definite` or requesting p in definite reasoning mode. This is because `MWebWFS`, similarly to `WFS`, cannot reason by cases and take advantage of the OWA rules. The reasoning mode `open` makes sense for `MWebAS`, only. We have included it in the `MWebWFS` semantics, for the sake of uniform presentation.

The `MWeb` framework has been implemented in the XSB system [64] 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 [45] which is called at query time by the `MWeb` top-level 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 [5].

More and more knowledge is becoming available in the Semantic Web [40], both in the form of RDFS and OWL ontologies. The Resource Description Framework Schema (RDFS) [57, 58] 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) [48] provides declarative constructs to express complex concepts and statements about properties with the semantics provided by the description logic language SROIQ [37]. 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 [24]). The OWL-2 RL profile [49] 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 [39, 49, 50], 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[21]. Another interesting extension of our framework is the support of equality and expressive built-in predicates.

[21] 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.

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 [28].

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 Group is defining a dialect for the language and semantics of production rules (RIF-PRD [61]) respecting the Object Management Group's adopted specification "Production Rule Representation" [47], 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 [1, 20, 65] to our `MWeb` framework. Additionally, we plan to define a notion of `m`-equivalent `MWeb` rule bases such that, for any modular rule base $\mathcal{S}$ and $s \in \mathcal{S}$, if s is replaced in $\mathcal{S}$ by an `m`-equivalent rule base s' then the `WMeb` semantics of the other rule bases in $\mathcal{S}$ will remain unaffected. This problem is related to the work in [46]. Another direction of research is the definition of modular Extended RDF (ERDF) ontologies, extending the Extended RDF framework, presented in [5], 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 [59]. The current proposal is a step in this direction.

Acknowledgments: This work has been benefited from discussions with Luís Moniz Pereira and José Júlio Alferes. Additionally, the authors would like to thank the reviewers for their valuable comments. This research has been partially funded by European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`)

APPENDIX

In this Appendix, we prove the Propositions, Theorems, and Corollaries presented in the main paper. First, we provide a few definitions from the `AS` semantics [31, 32] and the `WFSX` semantics [53, 3], adjusted properly to fit our `MWeb` framework definitions.

Let P be an ELP. A 2-valued interpretation of P is a set I, where $I \subseteq \mathtt{HB}_P$ s.t. either: $I \cap \neg I = \emptyset$ (*consistency*), or $I = \mathtt{HB}_P$. A 3-valued interpretation of P is a set

$I = T \cup {\sim}F$, where $T, F \subseteq \mathtt{HB}_P$ s.t. either: (i) $T \cap \neg T = \emptyset$ and $T \cap F = \emptyset$ (*consistency*), or (ii) $T = F = \mathtt{HB}_P$ [22].

Let P be an ELP. We denote by $[P]$ the ground version of P, that is P instantiated over all constants appearing in P. Let I be a 2-valued (resp. 3-valued) interpretation of P. We define: $I \models P$ iff for all $r \in [P]$, it holds that $I(Head_r) \geq I(Body_r)$. Let I, J be two 2-valued (resp. 3-valued) interpretations of P. We define: $I \leq_{\mathtt{t}} J$ iff for each $L \in \mathtt{HB}_P$, it holds that $I(L) \leq J(L)$. Additionally, we define $I \leq_{\mathtt{k}} J$ iff $I \subseteq J$.

A *definite rule* is an ELP rule without weak negation. Additionally, a *definite logic program* is an ELP with definite rules.

Let P be a definite logic program. We define $least_v^2(P)$ to be the minimal (w.r.t. $\leq_{\mathtt{t}}$) 2-valued interpretation I s.t. $I \models [P]$. Similarly, we define $least_v^3(P)$ to be the minimal (w.r.t. $\leq_{\mathtt{t}}$) 3-valued interpretation J s.t. $J \models [P]$.

Let P be an ELP and let I be a 2-valued interpretation of P. We denote by $P/^{\mathtt{AS}}I$ the *Gelfond-Lifschitz* P/I modulo operator [31]. Specifically, the logic program $P/^{\mathtt{AS}}I$ is obtained from $[P]$ as follows: (i) we remove from $[P]$, all rules that contain in their body a default literal ${\sim}L$ s.t. $I(L) = 1$, and (ii) we remove from the body of the remaining rules, any default literal ${\sim}L$ s.t. $I(L) = 0$. We say that I is an *answer set* of P iff $I = least^2(P/^{\mathtt{AS}}I)$ [31, 32].

Let P be an ELP and let I be a 3-valued interpretation of P. We define $Coh(I) = I \cup \{{\sim}L \mid L \in \mathtt{HB}_P \text{ and } \neg L \in I\}$[23]. Additionally, we denote by $P/^{\mathtt{WFSX}}I$ the P/I modulo operator of the WFSX semantics [53, 3]. Specifically, the logic program $P/^{\mathtt{WFSX}}I$ is obtained from $[P]$ as follows: (i) we remove from $[P]$, all rules that contain in their body an objective literal L s.t. $I(\neg L) = 1$ or a default literal ${\sim}L$ s.t. $I(L) = 1$, (ii) we remove from the body of the remaining rules, any default literal ${\sim}L$ s.t. $I(L) = 0$, and (iii) we replace all remaining default literals ${\sim}L$ with u. We say that I is a *partial stable model*[24] of P iff $I = Coh(least^3(P/^{\mathtt{WFSX}}I))$ [53, 3].

Finally, we note that many of the proofs use Theorem 1 and Corollary 2, provided in Section 6. Additionally, they use the notation N_{M}, defined above Theorem 1.

Proposition 1 Let N be a normal (resp. extended) interpretation of s in reasoning mode m w.r.t. $\mathcal{S}$. It always exists the minimal model of s in reasoning mode m w.r.t. $\mathcal{S}$ and N.

Proof: Let N be a normal interpretation of s in reasoning mode m w.r.t. $\mathcal{S}$. First, we will show that there exists a normal interpretation of s in reasoning mode m w.r.t. $\mathcal{S}$, M, that satisfies the conditions (1-3) of Definition 19. Let $N = least_v^2(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}}N_{\mathsf{N}})$. Additionally, let M be the normal interpretation of s in reasoning mode m w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{M}} = N$. Then, M satisfies the conditions (1-3) of Definition 19. In particular, M satisfies conditions (1,2) of Definition 19 due to the rules generated (in Section 6) from the translation of the `defines` declarations of the rule bases s', where $s' \in \mathcal{S}$ s.t. there exists $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$. In addition, M satisfies condition (3) of Definition 19 due to the rules generated (in Section 6) from the translation of the `uses` declarations of the rule bases s', where $s' \in \mathcal{S}$ s.t. there exists $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$, $\langle s', \mathtt{x}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$.

Assume now that there exists a normal interpretation of s in reasoning mode m w.r.t. $\mathcal{S}$, M', s.t. M' satisfies the conditions (1-3) of Definition 19 and $\mathsf{M}' <_{\mathtt{t}} \mathsf{M}$. Then, $N_{\mathsf{M}'} \models \Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}}N_{\mathsf{N}}$ and $N_{\mathsf{M}'} <_{\mathtt{t}} N_{\mathsf{M}}$. However, this is impossible. Thus, N_{M} is the minimal model of s in reasoning mode m w.r.t. $\mathcal{S}$ and N.

---

[22] Note that in contrast to [53, 3], our definition of a 3-valued interpretation of P includes also inconsistent and incoherent interpretations. We prefer this definition because it is more suitable for our proofs.

[23] Note that if $L \in \mathtt{HB}_P$ then $\neg(\neg L) = L$.

[24] Note that we extend the definition of a partial stable model, that is provided in [53, 3], to also include inconsistent partial stable models.

The case that N is an extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ is proved similarly to the previous case. □

Proposition 2 Let M be a consistent normal or extended interpretation of rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Let $s' \in \mathcal{S}$ and let $\mathtt{x} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\langle s', \mathtt{x} \rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}$. Additionally, let $\mathsf{M}' = \{M^{\mathtt{y}}_{s''} \in \mathsf{M} \mid \langle s'', \mathtt{y} \rangle \in D^{\mathcal{S}}_{s',\mathtt{x}}\}$. It holds that:

1. If $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{x},\mathcal{S}}(s')$.
2. If $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ then $\mathsf{M}' \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{x},\mathcal{S}}(s')$.

Proof:
1) We will denote $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ by Π and $\Pi^{\mathtt{x}}_{s',\mathcal{S}}$ by Π'. Since $\Pi^{\mathtt{m}}_{s,\mathcal{S}} = \cup\{P^{\mathtt{x}}_{s',\mathcal{S}} \mid \langle s', \mathtt{x} \rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$ and $D^{\mathcal{S}}_{s',\mathtt{x}} \subseteq D^{\mathcal{S}}_{s,\mathtt{m}}$, it follows that $\Pi' \subseteq \Pi$. We define $D = \mathtt{HB}_{\Pi'}$.

Assume that $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$. Then, based on Theorem 1, it holds that $N_{\mathsf{M}} = least^2_v(\Pi/^{\mathtt{AS}} N_{\mathsf{M}})$. Further, it holds that if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body^+_r \cup Body^-_r \subseteq D$. It follows, from these facts, that $N_{\mathsf{M}} \cap D = least^2_v(\Pi'/^{\mathtt{AS}}(N_{\mathsf{M}} \cap D))$. Therefore, $N_{\mathsf{M}} \cap D$ is an answer set of Π'. Since $N_{\mathsf{M}'} = N_{\mathsf{M}} \cap D$, it follows that $N_{\mathsf{M}'}$ is an answer set of Π'. It follows now from this and Theorem 1 that $\mathsf{M}' \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{x},\mathcal{S}}(s')$.

2) Let Π, Π', and D be defined as in 1). Additionally, let $D' = D \cup \sim D$. Note that $\Pi' \subseteq \Pi$. Assume that $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$. Then, based on Theorem 1, it holds that $N_{\mathsf{M}} = Coh(least^3_v(\Pi/^{\mathtt{WFSX}} N_{\mathsf{M}}))$. Further, it holds that (i) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body^+_r \cup Body^-_r \subseteq D$, and (ii) if $L \in D$ then any rule that defines L and $\neg L$ in $[\Pi]$ is found in $[\Pi']$. It follows, from these facts, that $N_{\mathsf{M}} \cap D' = Coh(least^3_v(\Pi'/^{\mathtt{WFSX}}(N_{\mathsf{M}} \cap D')))$. Since $N_{\mathsf{M}'} = N_{\mathsf{M}} \cap D'$, it follows that $N_{\mathsf{M}'}$ is a partial stable model of Π'. It follows now from this and Theorem 1 that $\mathsf{M}' \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{x},\mathcal{S}}(s')$. □

Proposition 3 If there exists $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent then:

1. $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$, and
2. $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}'\}$, where M' is inconsistent, or $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\}$.

Proof:
1) We will denote $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ by Π. Let $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent. It follows from Theorem 1 that N_{M} is an inconsistent answer set of Π. In [32], it is shown that if an extended logic program P has an inconsistent answer set then this is the only answer set of P. Thus, N_{M} is the only answer set of Π. Assume now that there exists $\mathsf{M}' \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ such that M' is consistent. Then, it follows from Theorem 1 that $N_{\mathsf{M}'}$ is a consistent answer set of Π, which is impossible. Thus, $\mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$.

2) Let $\mathsf{M} \in \mathcal{M}^{\mathtt{AS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent. Assume that $\mathsf{N} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ such that N is consistent. It follows from Theorem 1 that $N_{\mathsf{N}} = Coh(least^3_v(\Pi/^{\mathtt{WFSX}} N_{\mathsf{N}}))$. Note that $\Pi/^{\mathtt{AS}} N_{\mathsf{M}}$ contains all the definite rules of Π. We will denote $\Pi/^{\mathtt{AS}} N_{\mathsf{M}}$ by P and the definite rules of Π that appear in $\Pi/^{\mathtt{WFSX}} N_{\mathsf{N}}$ by P'. Let λ be the least integer such that for an $L \in \mathtt{HB}_P$, it holds that $L \in T^{\uparrow\lambda}_P(\emptyset)$ and $\neg L \in N_{\mathsf{N}}$. Assume that there exists such λ. Then, all the rules in P, applied in the derivation of L, while computing $T^{\uparrow\lambda}_P(\emptyset)$, appear in P'. Thus, $L \in \Psi^{*\uparrow\omega}_{P'}(\sim\mathtt{HB}_{P'})$[25]. Therefore, $L \in least^3_v(\Pi/^{\mathtt{WFSX}} N_{\mathsf{N}})$. Since $least^3_v(\Pi/^{\mathtt{WFSX}} N_{\mathsf{N}}) \subseteq N_{\mathsf{N}}$, it follows that $L \in N_{\mathsf{N}}$. Therefore, N_{N} is inconsistent. Thus, N is inconsistent, which is impossible. Assume now that there is no $L \in \mathtt{HB}_P$ s.t. $L \in T^{\uparrow\omega}_P(\emptyset)$ and $\neg L \in N_{\mathsf{N}}$. Since $T^{\uparrow\omega}_P(\emptyset)$ contains a pair of complementary literals, it follows that $\Psi^{*\uparrow\omega}_{P'}(\sim\mathtt{HB}_{P'})$ contains also this pair of complementary literals. Thus, $least^3_v(\Pi/^{\mathtt{WFSX}} N_{\mathsf{N}})$ is inconsistent. Therefore, N_{N} is inconsistent. Thus, N is inconsistent, which is impossible. Therefore, it holds that: (i) $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}'\}$, where M' is inconsistent, or (ii) $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\}$. □

[25] The Ψ^*_P operator is a generalization of the Van Emden-Kowalski least model operator T_P for definite logic programs P, and is defined in [3].

Proposition 4 If there exists $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent then (i) rule base s is contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, and (ii) $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$.

Proof:

i) Let $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent. Then, it follows from Theorem 1 that N_{M} is an inconsistent partial stable model of Π. We will denote $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ by P and $P/^{\mathtt{WFSX}}N_{\mathsf{M}}$ by P'. It holds that $\Psi^{*\uparrow\omega}_{P'}(\sim\mathtt{HB}_{P'})$ contains a pair of complementary literals L and $\neg L$. Let $N = \mathtt{HB}_P$ and $P'' = P/^{\mathtt{AS}}N$. Then, $T^{\uparrow\omega}_{P''}(\emptyset)$ contains also this pair of complementary literals. Thus, $N = least^2_v(P/^{\mathtt{AS}}N)$. Therefore, N is an inconsistent answer set of P. It now follows from Theorem 1 that rule base s is contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$.

ii) Let $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. M is inconsistent. It follows from i) that s is contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Therefore, it follows from Proposition 3 that $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) = \{\mathsf{M}\}$. □

Proposition 5 Assume that rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ is contradictory. It holds that:

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

Proof:

1) Let $\mathtt{m} \in \{\mathtt{o}, \mathtt{c}\}$ and assume that rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ is contradictory. We will show that rule base s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$ is also contradictory. Let M be the inconsistent normal answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Then, it follows from Theorem 1 that N_{M} is the inconsistent answer set of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Thus, $least^2_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}}N_{\mathsf{M}})$ is inconsistent. Note that $\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}}N_{\mathsf{M}}$ contains exactly all the definite rules in $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. It follows from the way $\Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ are defined that $least^2_v(\Pi^{\mathtt{d}}_{s,\mathcal{S}})$ is also inconsistent. Therefore, $\Pi^{\mathtt{d}}_{s,\mathcal{S}}$ has an inconsistent answer set. It now follows from Theorem 1 that rule base s has an inconsistent answer set in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$. Thus, rule base s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$ is contradictory.

Let $\mathtt{x} \in \{\mathtt{o}, \mathtt{c}, \mathtt{n}\}$ and assume that rule base s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$ is contradictory. It follows from Theorem 1 that $least^2_v(\Pi^{\mathtt{d}}_{s,\mathcal{S}})$ is inconsistent. Let N be the inconsistent 2-valued interpretation of $\Pi^{\mathtt{x}}_{s,\mathcal{S}}$ (i.e. $N = \mathtt{HB}_P$, where $P = \Pi^{\mathtt{x}}_{s,\mathcal{S}}$). As $\Pi^{\mathtt{x}}_{s,\mathcal{S}}/^{\mathtt{AS}}N$ contains all the definite rules in $\Pi^{\mathtt{x}}_{s,\mathcal{S}}$ and from the way $\Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and $\Pi^{\mathtt{x}}_{s,\mathcal{S}}$ are defined, it follows that $least^2_v(\Pi^{\mathtt{x}}_{s,\mathcal{S}}/^{\mathtt{AS}}N)$ is inconsistent. Thus, N is an answer set of $\Pi^{\mathtt{x}}_{s,\mathcal{S}}$. It now follows from Theorem 1 that rule base s has an inconsistent answer set in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$. Thus, rule base s in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$ is contradictory. Statement 1) now follows.

2) Assume that rule base s' in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$ is not contradictory. Let M be a consistent normal answer set of s' in reasoning mode $\mathtt{x}$ w.r.t. $\mathcal{S}$. Let $\mathsf{M}' = \{M^{\mathtt{y}}_{s''} \in \mathsf{M} \mid \langle s'', \mathtt{y}\rangle \in D^{\mathcal{S}}_{s,\mathtt{m}}\}$. Obviously, M' is consistent. It follows from Proposition 2 that M' is an answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Since M' is consistent and rule base s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ is contradictory, this is impossible. □

Proposition 6 It holds that: $|minimal_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))| \leq 1$.

Proof: Assume that rule base s has two different minimal (w.r.t. $\leq_{\mathtt{k}}$) extended answer sets in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, M and N. It follows from Theorem 1 that N_{M} and N_{N} are two different partial stable models of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Assume that N_{M} is not a minimal (w.r.t. $\leq_{\mathtt{k}}$) partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Thus, there exists a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$, $N_{\mathsf{M}'}$, s.t. $N_{\mathsf{M}'} <_{\mathtt{k}} N_{\mathsf{M}}$. But then, from Theorem 1, M' is an extended answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ and it holds that $\mathsf{M}' <_{\mathtt{k}} \mathsf{M}$, which is impossible. Similarly, it is shown that N_{N} is a minimal (w.r.t. $\leq_{\mathtt{k}}$) partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. However, it holds that if $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ has a minimal (w.r.t. $\leq_{\mathtt{k}}$) partial stable model, this is unique and is the well-founded model (according to WFSX semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Thus, $|minimal_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))| \leq 1$. □

Theorem 1 Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$. It holds that: M is a normal (resp. extended) answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ iff N_{M} is an answer set (resp. partial stable model) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$.

Proof:

$\Rightarrow$) Let M be a normal answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Assume that $N_{\mathsf{M}} \neq \mathit{least}^2_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}} N_{\mathsf{M}})$ and let $N = \mathit{least}^2_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}} N_{\mathsf{M}})$. Let N be the normal interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Note that N satisfies conditions (1-3) of Definition 19 (in Def. 19, replace M by N and N by M) and $\mathsf{N} <_{\mathtt{t}} \mathsf{M}$. Thus, $\mathsf{M} \neq \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$, which is impossible. Therefore, $N_{\mathsf{M}} = \mathit{least}^2_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}} N_{\mathsf{M}})$. Thus, N_{M} is an answer set of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$.

Let M be an extended answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ and let $\mathsf{M}' = \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. Assume that $N_{\mathsf{M}'} \neq \mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}})$ and let $N = \mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}})$. Let N be the extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Note that N satisfies conditions (1-3) of Definition 19 (in Def. 19, replace M by N and N by M) and $\mathsf{N} <_{\mathtt{t}} \mathsf{M}'$. Thus, $\mathsf{M}' \neq \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$, which is impossible. Therefore, $N_{\mathsf{M}'} = \mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}})$. Since $\mathsf{M} = \mathit{Coh}(\mathsf{M}')$, it follows that $N_{\mathsf{M}} = \mathit{Coh}(\mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}}))$. Thus, N_{M} is a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$.

$\Leftarrow$) Let N be an answer set of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Additionally, let M be the normal interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{M}} = N$. Then, $N_{\mathsf{M}} = \mathit{least}^2_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}} N_{\mathsf{M}})$. It follows that M satisfies conditions (1-3) of Definition 19 (in Def. 19, replace N by M). Assume that $\mathsf{M} \neq \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. Then, there is a normal interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, N, s.t. $\mathsf{N} = \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. But then, $N_{\mathsf{N}} <_{\mathtt{t}} N_{\mathsf{M}}$ and $N_{\mathsf{N}} \models \Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{AS}} N_{\mathsf{M}}$, which is impossible. Thus, $\mathsf{M} = \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. Therefore, M is a normal answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$.

Let N be a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Additionally, let M be the extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{M}} = N$. Then, $N_{\mathsf{M}} = \mathit{Coh}(\mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}}))$. Let $N_{\mathsf{M}'} = \mathit{least}^3_v(\Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}})$. It follows that M' satisfies conditions (1-3) of Definition 19 (in Def. 19, replace M by M' and N by M). Assume that $\mathsf{M}' \neq \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. Then, there is an extended interpretation of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, N, s.t. $\mathsf{N} = \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. But then $N_{\mathsf{N}} <_{\mathtt{t}} N_{\mathsf{M}'}$ and $N_{\mathsf{N}} \models \Pi^{\mathtt{m}}_{s,\mathcal{S}}/^{\mathtt{WFSX}} N_{\mathsf{M}}$, which is impossible. Thus, $\mathsf{M}' = \mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M})$. Since $N_{\mathsf{M}} = \mathit{Coh}(N_{\mathsf{M}'})$, it follows that $\mathsf{M} = \mathit{Coh}(\mathit{least}^{\mathtt{m}}_{\mathcal{S}}(s, \mathsf{M}))$. Thus, M is an extended answer set of s in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. $\square$

Proposition 7 Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $\mathtt{m} \in \{\mathtt{d}, \mathtt{o}, \mathtt{c}, \mathtt{n}\}$ s.t. $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) \neq \emptyset$. It holds that: $\mathsf{M} = \mathtt{mWF}^{\mathcal{S}}_{s,\mathtt{m}}$ iff N_{M} is the well-founded model (according to $\mathtt{WFSX}$ semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$.

Proof:

$\Rightarrow$) Since $\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s) \neq \emptyset$, it follows from Definition 23 that $\mathsf{M} = \mathit{least}_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))$. It follows from Theorem 1 that N_{M} is a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. It holds that the well-founded model (according to $\mathtt{WFSX}$ semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ is the least w.r.t. $\leq_{\mathtt{k}}$ partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ [3]. Assume that N_{M} is not the least w.r.t. $\leq_{\mathtt{k}}$ partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ and that there exists a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$, N_{N}, s.t. $N_{\mathsf{M}} \not\leq_{\mathtt{k}} N_{\mathsf{N}}$. Then, it follows from Theorem 1 that $\mathsf{M} \not\leq_{\mathtt{k}} \mathsf{N}$ and $\mathsf{N} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$. However, this is impossible. Thus, N_{M} is the well-founded model (according to $\mathtt{WFSX}$ semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$.

$\Leftarrow$) Let N_{M} be the well-founded model (according to $\mathtt{WFSX}$ semantics) of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$. Then, N_{M} is the least w.r.t. $\leq_{\mathtt{k}}$ partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ [3]. It follows from Theorem 1 that $\mathsf{M} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$. Assume that there exists $\mathsf{N} \in \mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s)$ s.t. $\mathsf{M} \not\leq_{\mathtt{k}} \mathsf{N}$. Then, it follows from Theorem 1 that N_{N} is a partial stable model of $\Pi^{\mathtt{m}}_{s,\mathcal{S}}$ and $N_{\mathsf{M}} \not\leq_{\mathtt{k}} N_{\mathsf{N}}$, which is impossible. Thus, $\mathsf{M} = \mathit{least}_{\leq_{\mathtt{k}}}(\mathcal{M}^{\mathtt{EAS}}_{\mathtt{m},\mathcal{S}}(s))$. $\square$

Proposition 8 Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $L \in \mathtt{HB}^{\mathcal{S}}_s \cup {\sim}\mathtt{HB}^{\mathcal{S}}_s$. It holds that: if $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$ then $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$.

Proof: Let $pred(L) = p$. If $p \in Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$. If $p \in Pred_s^{\mathtt{U}} - Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$. If $p \in Pred_s^{\mathtt{U}}$ and $qual(L)$ is defined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$.

In the case that rule base s is contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$ then it holds that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. Assume that rule base s is not contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$. Note that if N is a consistent answer set of an ELP P then $N \cup {\sim}(\mathtt{HB}_P - N)$ is a partial stable model of P. Since rule base s is not contradictory in reasoning mode $\mathtt{m}$ w.r.t. $\mathcal{S}$, it follows from Theorem 1 that $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$ has only consistent answer sets.

Assume now that $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. Then, it follows from Theorem 1 that for every partial stable model M of $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$, it holds that $M(\tau_s^{\mathtt{m}}(L)) = 1$. Thus, it follows that for every answer set M' of $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$, it holds that $M'(\tau_s^{\mathtt{m}}(L)) = 1$. Therefore, it follows from Theorem 1 that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. □

Proposition 9 Let $\mathcal{S}$ be a modular rule base, let $s \in \mathcal{S}$, and let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup {\sim}\mathtt{HB}_s^{\mathcal{S}}$. It holds that: (i) the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ is data complete for co-NP and program complete for co-NEXPTIME and (ii) the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ is data complete for P and program complete for EXPTIME.

Proof: Let $pred(L) = p$. If $p \in Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$. If $p \in Pred_s^{\mathtt{U}} - Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$. If $p \in Pred_s^{\mathtt{U}}$ and $qual(L)$ is defined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$.

(i) It follows from Corollary 2 that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $\tau_s^{\mathtt{m}}(L) \in \mathcal{C}_{\mathtt{AS}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{m}}\right)$. Note that there is a polynomial time transformation from $s, \mathcal{S}$ to $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$. Let P be an ELP and let $L' \in \mathtt{HB}_P \cup {\sim}\mathtt{HB}_P$. It is stated in [21] that the problem of establishing if $L' \in \mathcal{C}_{\mathtt{AS}}(P)$ is data complete for co-NP and program complete for co-NEXPTIME. Thus, the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ has data complexity co-NP and program complexity co-NEXPTIME. Now due to Proposition 11, it follows that the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ is data complete for co-NP and program complete for co-NEXPTIME.

(ii) It follows from Corollary 2 that $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $\tau_s^{\mathtt{m}}(L) \in \mathcal{C}_{\mathtt{WFSX}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{m}}\right)$. Note that there is a polynomial time transformation from $s, \mathcal{S}$ to $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$. Let P be an ELP and let $L' \in \mathtt{HB}_P \cup {\sim}\mathtt{HB}_P$. It is shown in [3] that the problem of establishing if $L' \in \mathcal{C}_{\mathtt{WFSX}}(P)$ has data complexity P. Thus, the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ has data complexity P. Additionally, it is shown in [3] that on normal logic programs, WFS and WFSX have the same conclusions. Let P^{nr} be a normal logic program and $L^{\mathrm{nr}} \in \mathtt{HB}_{P^{\mathrm{nr}}} \cup {\sim}\mathtt{HB}_{P^{\mathrm{nr}}}$. It is stated in [21] that the problem of establishing if $L^{\mathrm{nr}} \in \mathcal{C}_{\mathtt{WFS}}(P^{\mathrm{nr}})$ is data complete for P. Therefore, the problem of establishing if $L' \in \mathcal{C}_{\mathtt{WFSX}}(P)$ is data complete for P. Now due to Proposition 11, it follows that the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ is data complete for P.

Let P' be the instantiated version of P. Note that P' is exponentially larger than P w.r.t. the program size. It holds that $L' \in \mathcal{C}_{\mathtt{WFSX}}(P)$ iff $L' \in \mathcal{C}_{\mathtt{WFSX}}(P')$. Therefore, the problem of establishing if $L' \in \mathcal{C}_{\mathtt{WFSX}}(P)$ has program complexity EXPTIME. Thus, the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ has program complexity EXPTIME. Additionally, it is stated in [21] that the problem of establishing if $L^{\mathrm{nr}} \in \mathcal{C}_{\mathtt{WFS}}(P^{\mathrm{nr}})$ is program complete for EXPTIME. Therefore, the problem of establishing if $L' \in \mathcal{C}_{\mathtt{WFSX}}(P)$ is program complete for EXPTIME. Now due to Proposition 11, it follows that the problem of establishing if $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ is program complete for EXPTIME. □

Proposition 10 Let $\mathcal{S}$ be a modular rule base. Additionally, let $p \in Pred_s^{\mathtt{D}}$ s.t. $mode_s^{\mathtt{D}}(p) = \mathtt{d}$, and let $L \in [p]_{\mathcal{S}} \cup {\sim}[p]_{\mathcal{S}}$. It holds that: $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$.

Proof: Note that $\Pi_{s,\mathcal{S}}^{\mathtt{d}}$ is a definite logic program. Thus, it holds that $\mathcal{C}_{\mathtt{AS}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{d}}\right) = \mathcal{C}_{\mathtt{WFSX}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{d}}\right)$. Based on Corollary 2, it holds that: (i) $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $\tau_s^{\mathtt{d}}(L) \in \mathcal{C}_{\mathtt{AS}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{d}}\right)$, and (ii) $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $\tau_s^{\mathtt{d}}(L) \in \mathcal{C}_{\mathtt{WFSX}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{d}}\right)$. Therefore, it follows that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. □

Proposition 11 Let s be a rule base s.t. $Pred_s^{\mathtt{U}} = \emptyset$ and for all $p \in Pred_s^{\mathtt{D}}$, $mode_s^{\mathtt{D}}(p) = \mathtt{n}$. Let $\mathcal{S} = \{s\}$, let $p \in Pred_s^{\mathtt{D}}$, and let $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$. It holds that: (i) $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $L \in \mathcal{C}_{\mathtt{AS}}(P_s)$, and (ii) $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $L \in \mathcal{C}_{\mathtt{WFSX}}(P_s)$.

Proof:

(i) Note that $\Pi_{s,\mathcal{S}}^{\mathtt{n}}$ results from P_s, if each $L' \in \mathtt{HB}_{P_s}$ is replaced by $\tau_s^{\mathtt{n}}(L')$. Based on this and Corollary 2, it follows that: $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $\tau_s^{\mathtt{n}}(L) \in \mathcal{C}_{\mathtt{AS}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{n}}\right)$ iff $L \in \mathcal{C}_{\mathtt{AS}}(P_s)$.

(ii) Similarly to (i), it follows that: $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$ iff $\tau_s^{\mathtt{n}}(L) \in \mathcal{C}_{\mathtt{WFSX}}\left(\Pi_{s,\mathcal{S}}^{\mathtt{n}}\right)$ iff $L \in \mathcal{C}_{\mathtt{WFSX}}(P_s)$. $\square$

Proposition 12 Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases s.t. $\mathcal{S} \subseteq \mathcal{S}'$ and for all $s \in \mathcal{S}$ and $p \in Pred_s^{\mathtt{U}}$, $Import_s^{\mathcal{S}}(p) = Import_s^{\mathcal{S}'}(p)$. Let $L \in \mathtt{HB}_s^{\mathcal{S}} \cup \sim\mathtt{HB}_s^{\mathcal{S}}$. It holds that: $s \models_{\mathcal{S}}^{\mathtt{SEM}} L$ iff $s \models_{\mathcal{S}'}^{\mathtt{SEM}} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.

Proof:

Case $\mathtt{SEM} = \mathtt{mAS}$: Let $pred(L) = p$. If $p \in Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = |mode_s^{\mathtt{D}}(p)|$. If $p \in Pred_s^{\mathtt{U}} - Pred_s^{\mathtt{D}}$ and $qual(L)$ is undefined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$. If $p \in Pred_s^{\mathtt{U}}$ and $qual(L)$ is defined then let $\mathtt{m} = mode_s^{\mathtt{U}}(p)$.

Note that $\Pi_{s,\mathcal{S}}^{\mathtt{m}} = \Pi_{s,\mathcal{S}'}^{\mathtt{m}}$. Now, let $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. Then, it follows from Theorem 1 that $\tau_s^{\mathtt{m}}(L)$ is true according to all answer sets of $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$. Thus, $\tau_s^{\mathtt{m}}(L)$ is true according to all answer sets of $\Pi_{s,\mathcal{S}'}^{\mathtt{m}}$. Therefore, it follows from Theorem 1 that $s \models_{\mathcal{S}'}^{\mathtt{mAS}} L$. Similarly, it follows that if $s \models_{\mathcal{S}'}^{\mathtt{mAS}} L$ then $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$.

Case $\mathtt{SEM} = \mathtt{mWFS}$: Let $\mathtt{m}$ be defined as in the previous case. Note that $\Pi_{s,\mathcal{S}}^{\mathtt{m}} = \Pi_{s,\mathcal{S}'}^{\mathtt{m}}$.

Now, let $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. Then, it follows from Theorem 1 that $\tau_s^{\mathtt{m}}(L)$ is true according to all partial stable models of $\Pi_{s,\mathcal{S}}^{\mathtt{m}}$. Thus, $\tau_s^{\mathtt{m}}(L)$ is true according to all partial stable models of $\Pi_{s,\mathcal{S}'}^{\mathtt{m}}$. Therefore, it follows from Theorem 1 that $s \models_{\mathcal{S}'}^{\mathtt{mWFS}} L$. Similarly, it follows that if $s \models_{\mathcal{S}'}^{\mathtt{mWFS}} L$ then $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. $\square$

Proposition 13 Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases such that for all $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}'$:

1. $Nam_s = Nam_{s'}$, $P_s \subseteq P_{s'}$, $Pred_s^{\mathtt{D}} \subseteq Pred_{s'}^{\mathtt{D}}$, $Pred_s^{\mathtt{U}} \subseteq Pred_{s'}^{\mathtt{U}}$,
2. For all $p \in Pred_s^{\mathtt{D}}$:
   $scope_s(p) \leq scope_{s'}(p)$, $mode_s^{\mathtt{D}}(p) = mode_{s'}^{\mathtt{D}}(p)$, $context_s(p) = context_{s'}(p)$, $Export_s^{\mathcal{S}}(p) \subseteq Export_{s'}^{\mathcal{S}'}(p)$, and
3. For all $p \in Pred_s^{\mathtt{U}}$:
   $mode_s^{\mathtt{U}}(p) = mode_{s'}^{\mathtt{U}}(p)$ and $Import_s^{\mathcal{S}}(p) \subseteq Import_{s'}^{\mathcal{S}'}(p)$.

Let $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$ s.t. $Nam_s = Nam_{s'}$. Let $p \in Pred_s^{\mathtt{D}}$ s.t. $mode_s^{\mathtt{D}}(p) \in \{\mathtt{d}, \mathtt{o}\}$ and let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models_{\mathcal{S}}^{\mathtt{SEM}} L$ then $s' \models_{\mathcal{S}'}^{\mathtt{SEM}} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.

Proof:

Case $mode_s^{\mathtt{D}}(p) = \mathtt{d}$ *and* $\mathtt{SEM} = \mathtt{mAS}$: Let $\Pi = \Pi_{s,\mathcal{S}}^{\mathtt{d}}$ and let $\Pi' = \Pi_{s',\mathcal{S}'}^{\mathtt{d}}$. Note that both Π and Π' are definite logic programs and that it holds $\Pi \subseteq \Pi'$. Additionally, note that the $\mathtt{AS}$ semantics of a definite logic program P coincide with $least_v^2(P)$. Assume that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. Then, for every normal answer set of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$, M, it holds that $M_s^{\mathtt{d}}(L) = 1$. Then, according to Theorem 1, for every answer set M of Π, it holds that $\tau_s^{\mathtt{d}}(L) \in M$. Thus, $\tau_s^{\mathtt{d}}(L) \in least_v^2(\Pi)$. It now follows that $\tau_{s'}^{\mathtt{d}}(L) \in least_v^2(\Pi')$. Therefore, for every answer set N of Π', it holds that $\tau_{s'}^{\mathtt{d}}(L) \in N$. Thus, from Theorem 1, for every normal answer set of s' in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}'$, N, it holds that $N_{s'}^{\mathtt{d}}(L) = 1$. Thus, $s' \models_{\mathcal{S}'}^{\mathtt{mAS}} L$.

Case $mode_s^{\mathtt{D}}(p) = \mathtt{d}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$: Let $\Pi = \Pi_{s,\mathcal{S}}^{\mathtt{d}}$ and let $\Pi' = \Pi_{s',\mathcal{S}'}^{\mathtt{d}}$. Note that both Π and Π' are definite logic programs and that it holds $\Pi \subseteq \Pi'$. Additionally, note that the $\mathtt{WFSX}$ semantics of a definite logic program P coincide with $least_v^3(P)$. Assume that $s \models_{\mathcal{S}}^{\mathtt{mWFSX}} L$. Then, for every extended answer set of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$, M, it holds that $M_s^{\mathtt{d}}(L) = 1$. Then, according to Theorem 1, for every partial stable model M of Π, it holds that $\tau_s^{\mathtt{d}}(L) \in M$. Thus, $\tau_s^{\mathtt{d}}(L) \in least_v^3(\Pi)$. It now follows that $\tau_{s'}^{\mathtt{d}}(L) \in least_v^3(\Pi')$.

Therefore, for every partial stable model N of Π', it holds that $\tau^{\mathtt{d}}_{s'}(L) \in N$. Thus, from Theorem 1, for every extended answer set of s' in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}'$, N, it holds that $N^{\mathtt{d}}_{s'}(L) = 1$. Thus, $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$.

Case $mode^{\mathtt{D}}_s(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mAS}$: Let $\Pi = \Pi^{\mathtt{o}}_{s,\mathcal{S}}$, let $\Pi' = \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$, and let P be the set of definite rules in $\Pi^{\mathtt{o}}_{s,\mathcal{S}}$. Note that it holds $P \subseteq \Pi \subseteq \Pi'$. In the case that s' is contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$, it holds that $s' \models^{\mathtt{mAS}}_{\mathcal{S}'} L$. Assume now that s' is not contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Additionally, assume that $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$ and let M be a normal answer set of s' in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Note that M is consistent. It follows from Theorem 1 that N_{M} is a consistent answer set of Π'. Thus, $N_{\mathsf{M}} = least^2_v(\Pi'/^{\mathtt{AS}}N_{\mathsf{M}})$. We will show that $M^{\mathtt{o}}_{s'}(L) = 1$.

Let $I = \{\tau^{\mathtt{o}}_t(p'(c_1, ..., c_k)) \mid \langle t, \mathtt{o}\rangle \in D^{\mathtt{o}}_{s,\mathcal{S}},\ p' \in Pred^{\mathtt{D}}_t,\ mode^{\mathtt{D}}_t(p') \in \{\mathtt{o}, \mathtt{c}\}$, and (i) $\tau^{\mathtt{o}}_t(cxt_t(p')(c_1, ..., c_k)) \in least^2_v(P)$, if $cxt_t(p')$ is defined, or (ii) $c_1, ..., c_k \in \mathtt{HU}_{\mathcal{S}}$, otherwise$\}$. Additionally, let $I' = (I \cap N_{\mathsf{M}}) \cup (\neg I \cap N_{\mathsf{M}})$ and let $N = least^2_v(\Pi/^{\mathtt{AS}}I')$. It holds that N is consistent and $N = least^2_v(\Pi/^{\mathtt{AS}}N)$. Thus, N is an answer set of Π. Let N be a normal interpretation of s in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Then, it follows from Theorem 1 that N is a normal answer set of s in reasoning mode $\mathtt{o}$ w.r.t $\mathcal{S}$. Since $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$, it holds that $N^{\mathtt{o}}_s(L) = 1$. Thus, $N(\tau^{\mathtt{o}}_s(L)) = 1$. It holds that $least^2_v(\Pi/^{\mathtt{AS}}N) \subseteq least^2_v(\Pi'/^{\mathtt{AS}}N_{\mathsf{M}})$. Thus, $N \subseteq N_{\mathsf{M}}$. It follows from this that $N_{\mathsf{M}}(\tau^{\mathtt{o}}_{s'}(L)) = 1$. Therefore, $M^{\mathtt{o}}_{s'}(L) = 1$. Thus, for every normal answer set of s' in reasoning mode $\mathtt{o}$ w.r.t $\mathcal{S}'$, M, it holds that $M^{\mathtt{o}}_{s'}(L) = 1$. Therefore, $s' \models^{\mathtt{mAS}}_{\mathcal{S}'} L$.

Case $mode^{\mathtt{D}}_s(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$: Let $\Pi = \Pi^{\mathtt{o}}_{s,\mathcal{S}}$, let $\Pi' = \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$, and let P be the set of definite rules in $\Pi^{\mathtt{o}}_{s,\mathcal{S}}$. Note that it holds $P \subseteq \Pi \subseteq \Pi'$. In the case that s' is contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$, it follows from Proposition 3 that $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$. Assume now that s' is not contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Assume that $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$ and let M be an extended answer set of s' in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. It follows from Proposition 4 that M is consistent. Thus, it follows from Theorem 1 that N_{M} is a consistent partial stable model of Π'. Thus, $N_{\mathsf{M}} = Coh(least^3_v(\Pi'/^{\mathtt{WFSX}}N_{\mathsf{M}}))$. We will show that $M^{\mathtt{o}}_{s'}(L) = 1$.

Let $I = \{\tau^{\mathtt{o}}_t(p'(c_1, ..., c_k)) \mid \langle t, \mathtt{o}\rangle \in D^{\mathtt{o}}_{s,\mathcal{S}},\ p' \in Pred^{\mathtt{D}}_t,\ mode^{\mathtt{D}}_t(p') \in \{\mathtt{o}, \mathtt{c}\}$, and (i) $\tau^{\mathtt{o}}_t(cxt_t(p')(c_1, ..., c_k)) \in least^3_v(P)$, if $cxt_t(p')$ is defined, or (ii) $c_1, ..., c_k \in \mathtt{HU}_{\mathcal{S}}$, otherwise$\}$. Additionally, let $I' = (I \cap N_{\mathsf{M}}) \cup (\neg I \cap N_{\mathsf{M}}) \cup ({\sim} I \cap N_{\mathsf{M}}) \cup ({\sim}(\neg I) \cap N_{\mathsf{M}})$ and let $N = Coh(least^3_v(\Pi/^{\mathtt{WFSX}}I'))$. It holds that N is consistent and $N = Coh(least^3_v(\Pi/^{\mathtt{WFSX}}N))$. Thus, N is a partial stable model of Π. Let N be an extended interpretation of s in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Then, it follows from Theorem 1 that N is an extended answer set of s in reasoning mode $\mathtt{o}$ w.r.t $\mathcal{S}$. Since $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$, it holds that $N^{\mathtt{o}}_s(L) = 1$. Thus, $N(\tau^{\mathtt{o}}_s(L)) = 1$. It holds that $Coh(least^3_v(\Pi/^{\mathtt{WFSX}}N)) \subseteq Coh(least^3_v(\Pi'/^{\mathtt{WFSX}}N_{\mathsf{M}}))$. Thus, $N \subseteq N_{\mathsf{M}}$. It follows from this that $N_{\mathsf{M}}(\tau^{\mathtt{o}}_{s'}(L)) = 1$. Therefore, $M^{\mathtt{o}}_{s'}(L) = 1$. Thus, for every extended answer set of s' in reasoning mode $\mathtt{o}$ w.r.t $\mathcal{S}'$, M, it holds that $M^{\mathtt{o}}_{s'}(L) = 1$. Therefore, $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$. □

Proposition 14 Let $\mathcal{S}$ and $\mathcal{S}'$ be modular rule bases such that for all $s \in \mathcal{S}$, there exists $s' \in \mathcal{S}'$:

1. $Nam_s = Nam_{s'}$, $P_s = P_{s'}$, $Pred^{\mathtt{D}}_s = Pred^{\mathtt{D}}_{s'}$, $Pred^{\mathtt{U}}_s = Pred^{\mathtt{U}}_{s'}$, and
2. For all $p \in Pred^{\mathtt{D}}_s$:
   (a) $scope_s(p) = scope_{s'}(p)$ and $Export^{\mathcal{S}}_s(p) = Export^{\mathcal{S}'}_{s'}(p)$,
   (b) $mode^{\mathtt{D}}_s(p) \leq |mode^{\mathtt{D}}_{s'}(p)|$,
   (d) if $|mode^{\mathtt{D}}_s(p)|$, $|mode^{\mathtt{D}}_{s'}(p)| \in \{\mathtt{o}, \mathtt{c}\}$ then $context_s(p) = context_{s'}(p)$.
3. For all $p \in Pred^{\mathtt{U}}_s$:
   $mode^{\mathtt{U}}_s(p) \leq mode^{\mathtt{U}}_{s'}(p)$ and $Import^{\mathcal{S}}_s(p) = Import^{\mathcal{S}'}_{s'}(p)$.

Let $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$ s.t. $Nam_s = Nam_{s'}$. Let $p \in Pred^{\mathtt{D}}_s$ s.t. $mode^{\mathtt{D}}_s(p), mode^{\mathtt{D}}_{s'}(p) \in \{\mathtt{d}, \mathtt{o}\}$ and let $L \in [p]_{\mathcal{S}}$. It holds that: if $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$ then $s' \models^{\mathtt{SEM}}_{\mathcal{S}'} L$, for $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.
Proof:

Case $mode^{\mathtt{D}}_s(p) = mode^{\mathtt{D}}_{s'}(p) = \mathtt{d}$ *and* $\mathtt{SEM} = \mathtt{mAS}$: Let $\Pi = \Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and let $\Pi' = \Pi^{\mathtt{d}}_{s',\mathcal{S}'}$. Note that it holds $\Pi = \Pi'$. Assume that $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$. Then, for every normal answer set of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$, M, it holds that $M^{\mathtt{d}}_s(L) = 1$. Then, according to Theorem 1, for every answer set M of Π, it holds that $\tau^{\mathtt{d}}_s(L) \in M$. Therefore, for every answer set N of Π', it holds that $\tau^{\mathtt{d}}_{s'}(L) \in N$. Thus, from Theorem 1, for every normal answer set of s' in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}'$, N, it holds that $N^{\mathtt{d}}_{s'}(L) = 1$. Thus, $s' \models^{\mathtt{mAS}}_{\mathcal{S}'} L$.

Case $mode^{\mathtt{D}}_s(p) = mode^{\mathtt{D}}_{s'}(p) = \mathtt{d}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$: Let $\Pi = \Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and let $\Pi' = \Pi^{\mathtt{d}}_{s',\mathcal{S}'}$. Note that it holds $\Pi = \Pi'$. Assume that $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$. Then, for every extended answer set of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$, M, it holds that $M^{\mathtt{d}}_s(L) = 1$. Then, according to Theorem 1, for every partial stable model M of Π, it holds that $\tau^{\mathtt{d}}_s(L) \in M$. Therefore, for every partial stable model N of Π', it holds that $\tau^{\mathtt{d}}_{s'}(L) \in N$. Thus, from Theorem 1, for every extended answer set of s' in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}'$, N, it holds that $N^{\mathtt{d}}_{s'}(L) = 1$. Thus, $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$.

Case $mode^{\mathtt{D}}_s(p) = mode^{\mathtt{D}}_{s'}(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mAS}$: It is proved similarly to *Case* $mode^{\mathtt{D}}_s(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mAS}$, in the proof of Proposition 13 (though $\Pi^{\mathtt{o}}_{s,\mathcal{S}} \not\subseteq \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$).

Case $mode^{\mathtt{D}}_s(p) = mode^{\mathtt{D}}_{s'}(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$: It is proved similarly to *Case* $mode^{\mathtt{D}}_s(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$, in the proof of Proposition 13 (though $\Pi^{\mathtt{o}}_{s,\mathcal{S}} \not\subseteq \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$).

Case $mode^{\mathtt{D}}_s(p) = \mathtt{d}, mode^{\mathtt{D}}_{s'}(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mAS}$: Let $\Pi = \Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and let $\Pi' = \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$. Note that Π is a definite logic program. Let P be the set of definite rules in Π'. In the case that s' is contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$, it holds that $s' \models^{\mathtt{mAS}}_{\mathcal{S}'} L$. Assume now that s' is not contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Additionally, assume that $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$ and let M be a normal answer set of s' in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Note that M is consistent. Thus, it follows from Theorem 1 that N_{M} is a consistent answer set of Π'. We will show that $M^{\mathtt{o}}_{s'}(L) = 1$.

Let $N = least^2_v(\Pi)$. It holds that $\{\tau^{\mathtt{o}}_{s'}(L') \mid L' \in \mathtt{HB}^{\mathcal{S}}_s,\ \tau^{\mathtt{d}}_s(L') \in N,\ s' \in \mathcal{S}'$, and $Nam_s = Nam_{s'}\} \subseteq least^2_v(P) \subseteq N_{\mathsf{M}}$. Obviously, $N = least^2_v(\Pi/^{\mathtt{AS}}N)$. Thus, N is an answer set of Π. Let N be a normal interpretation of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Then, it follows from Theorem 1 that N is a normal answer set of s in reasoning mode $\mathtt{d}$ w.r.t $\mathcal{S}$. Since $s \models^{\mathtt{mAS}}_{\mathcal{S}} L$, it holds that $N^{\mathtt{d}}_s(L) = 1$. Thus, $N(\tau^{\mathtt{d}}_s(L)) = 1$. It follows from this that $N_{\mathsf{M}}(\tau^{\mathtt{o}}_{s'}(L)) = 1$. Thus, $M^{\mathtt{o}}_{s'}(L) = 1$. Therefore, $s' \models^{\mathtt{mAS}}_{\mathcal{S}'} L$.

Case $mode^{\mathtt{D}}_s(p) = \mathtt{d}, mode^{\mathtt{D}}_{s'}(p) = \mathtt{o}$ *and* $\mathtt{SEM} = \mathtt{mWFS}$: Let $\Pi = \Pi^{\mathtt{d}}_{s,\mathcal{S}}$ and let $\Pi' = \Pi^{\mathtt{o}}_{s',\mathcal{S}'}$. Let P be the set of definite rules in Π'. Note that Π is a definite logic program. In the case that s' is contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$, it holds that $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$. Assume now that s' is not contradictory in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. Additionally, assume that $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$ and let M be an extended answer set of s' in reasoning mode $\mathtt{o}$ w.r.t. $\mathcal{S}'$. It follows from Proposition 4 that M is consistent. Thus, it follows from Theorem 1 that N_{M} is a consistent partial stable model of Π'. We will show that $M^{\mathtt{o}}_{s'}(L) = 1$.

Let $N = least^3_v(\Pi)$. It holds that $\{\tau^{\mathtt{o}}_{s'}(L') \mid L' \in \mathtt{HB}^{\mathcal{S}}_s, \tau^{\mathtt{d}}_s(L') \in N,\ s' \in \mathcal{S}'$, and $Nam_s = Nam_{s'}\} \subseteq least^3_v(P) \subseteq N_{\mathsf{M}}$. Since N_{M} is consistent, N is also consistent and it holds $N = Coh(least^3_v(\Pi/^{\mathtt{WFSX}}N))$. Thus, N is a partial stable model of Π. Let N be an extended interpretation of s in reasoning mode $\mathtt{d}$ w.r.t. $\mathcal{S}$ s.t. $N_{\mathsf{N}} = N$. Then, it follows from Theorem 1 that N is an extended answer set of s in reasoning mode $\mathtt{d}$ w.r.t $\mathcal{S}$. Since $s \models^{\mathtt{mWFS}}_{\mathcal{S}} L$, it holds that $N^{\mathtt{d}}_s(L) = 1$. Thus, $N(\tau^{\mathtt{d}}_s(L)) = 1$. It follows from this that $N_{\mathsf{M}}(\tau^{\mathtt{o}}_{s'}(L)) = 1$. Thus, $M^{\mathtt{o}}_{s'}(L) = 1$. Therefore, $s' \models^{\mathtt{mWFS}}_{\mathcal{S}'} L$.

Proposition 15 Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$. Let $p \in Pred^{\mathtt{D}}_s$ s.t. p is $\mathtt{c}$-stratified in s w.r.t. $\mathcal{S}$ and let $L = p(c_1, ..., c_n)$, where $c_i \in \mathtt{HU}_{\mathcal{S}}$, for $i = 1, ..., n$. Let $\mathtt{SEM} \in \{\mathtt{mAS}, \mathtt{mWFS}\}$.

1. If p is freely (positively or negatively) closed in s then: $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$ or $s \models^{\mathtt{SEM}}_{\mathcal{S}} \neg L$,
2. If p is (positively or negatively) closed in s w.r.t. context cxt then:
   $s \models^{\mathtt{SEM}}_{\mathcal{S}} L$, or $s \models^{\mathtt{SEM}}_{\mathcal{S}} \neg L$, or $s \models^{\mathtt{SEM}}_{\mathcal{S}} \sim cxt(c_1, ..., c_n)$.

Proof:
1) *Case* $\texttt{SEM} = \texttt{mAS}$: First assume that $mode^{\texttt{D}}_s(p) = \texttt{c}^+$. Obviously, in the case that s is contradictory in reasoning mode $\texttt{c}$ w.r.t. $\mathcal{S}$ or $\mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s) = \{\}$, statement 1. holds. Assume now that s is not contradictory in reasoning mode $\texttt{c}$ w.r.t. $\mathcal{S}$ and that $\mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s) \neq \{\}$. We will denote $\Pi^{\texttt{c}}_{s,\mathcal{S}}$ by Π. Let D be the smallest set s.t. (i) $\tau^{\texttt{c}}_s(\neg L) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body^+_r \cup Body^-_r \subseteq D$. We define $\Pi_p = \{r \in [\Pi] \mid Head_r \in D\}$.

Let A be an atom in $\texttt{HB}_{\Pi_p}$, we define: (i) $\tau(A) = A$, (ii) $\tau(\sim A) = \sim A$, (iii) $\tau(\neg A) = \neg\_A$, and (iv) $\tau(\sim \neg A) = \sim \neg\_A$. Let $\Pi^{\texttt{nr}}_p$ be the normal logic program that results, if we replace each rule $L_0 \leftarrow L_1, ..., L_n$ in Π_p by $\tau(L_0) \leftarrow \tau(L_1), ..., \tau(L_n)$. It is easy to see that $\Pi^{\texttt{nr}}_p$ is a normal logic program. Since p is $\texttt{c}$-stratified in s w.r.t. $\mathcal{S}$, it follows that $\Pi^{\texttt{nr}}_p$ is stratified. Thus, $\Pi^{\texttt{nr}}_p$ has a unique stable model. It follows from this that Π_p has a unique answer set. Let $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$. Since s is not contradictory in reasoning mode $\texttt{c}$ w.r.t. $\mathcal{S}$, it follows that M is consistent. Thus, it follows from Theorem 1 that N_{M} is a consistent answer set of Π. Therefore, $N_{\mathsf{M}} = least^2_v(\Pi/^{\texttt{AS}}N_{\mathsf{M}})$. Due to (ii) in the definition of D, it holds that $N_{\mathsf{M}} \cap D = least^2_v(\Pi_p/^{\texttt{AS}}(N_{\mathsf{M}} \cap D))$. Now it follows from this that $N_{\mathsf{M}} \cap D$ is the unique answer set of Π_p. Thus, for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $N_{\mathsf{M}}(\tau^{\texttt{c}}_s(L)) = 1$ or for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $N_{\mathsf{M}}(\tau^{\texttt{c}}_s(L)) = 0$. Note that in Π_p, there exist the rule $\neg Nam_s{:}\texttt{c\_}p(c_1, ..., c_n) \leftarrow \sim Nam_s{:}\texttt{c\_}p(c_1, ..., c_n)$. It follows from this that for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $N_{\mathsf{M}}(\tau^{\texttt{c}}_s(L)) = 1$ or for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $N_{\mathsf{M}}(\tau^{\texttt{c}}_s(\neg L)) = 1$. Therefore, it follows that for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $M^{\texttt{c}}_s(L) = 1$ or for all $\mathsf{M} \in \mathcal{M}^{\texttt{AS}}_{\texttt{c},\mathcal{S}}(s)$, $M^{\texttt{c}}_s(\neg L) = 1$. Now statement 1. follows.
Now assume that $mode^{\texttt{D}}_s(p) = \texttt{c}^-$. The proof of statement 1. follows similarly to the previous case, with the difference that D is defined as the smallest set s.t. (i) $\tau^{\texttt{c}}_s(L) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body^+_r \cup Body^-_r \subseteq D$. Additionally, note that Π_p now contains the rule $Nam_s{:}\texttt{c\_}p(c_1, ..., c_n) \leftarrow \sim\neg Nam_s{:}\texttt{c\_}p(c_1, ..., c_n)$.

Case $\texttt{SEM} = \texttt{mWFS}$: First assume that $mode^{\texttt{D}}_s(p) = \texttt{c}^+$. Obviously, in the case that $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) = \{\mathsf{M}\}$, where M is inconsistent or $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) = \{\}$, statement 1. holds. Assume now that $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) \neq \{\mathsf{M}\}$, where M is inconsistent and that $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) \neq \{\}$. We will denote $\Pi^{\texttt{c}}_{s,\mathcal{S}}$ by Π. Let D be the smallest set s.t. (i) $\tau^{\texttt{c}}_s(\neg L) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body^+_r \cup Body^-_r \subseteq D$. We define $\Pi_p = \{r \in [\Pi] \mid Head_r \in D\}$.

Let A be an atom in $\texttt{HB}_{\Pi_p}$, we define: (i) $\tau(A) = A$, (ii) $\tau(\sim A) = \sim A$, (iii) $\tau(\neg A) = \neg\_A$, and (iv) $\tau(\sim \neg A) = \sim \neg\_A$. Let $\Pi^{\texttt{nr}}_p$ be the normal logic program that results if we replace each rule $L_0 \leftarrow L_1, ..., L_n$ in Π_p by $\tau(L_0) \leftarrow \tau(L_1), ..., \tau(L_n)$. It is easy to see that $\Pi^{\texttt{nr}}_p$ is a normal logic program. Since p is $\texttt{c}$-stratified in s w.r.t. $\mathcal{S}$, it follows that $\Pi^{\texttt{nr}}_p$ is stratified. Thus, $\Pi^{\texttt{nr}}_p$ has a well-founded model $W^{\texttt{nr}}_p$ s.t. for all $L' \in D$, $W^{\texttt{nr}}_p(\tau(L')) = 1$ or $W^{\texttt{nr}}_p(\tau(L')) = 0$.

We will denote $\texttt{mWF}^{\mathcal{S}}_{s,\texttt{c}}$ by W. Since $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) \neq \{\mathsf{M}\}$, where M is inconsistent, and $\mathcal{M}^{\texttt{EAS}}_{\texttt{c},\mathcal{S}}(s) \neq \{\}$, it follows from Definition 23 that W is consistent. Thus, it follows from Proposition 7 that N_{W} is the well-founded model of Π (according to $\texttt{WFSX}$ sematics) and that N_{W} is consistent.

Lemma: It holds that $W^{\texttt{nr}}_p = \{\tau(A) \mid A \in N_{\mathsf{W}} \cap (D \cup \sim D)\}$.
Proof: Consider the sequence $\{I_\alpha\}_{\alpha \in I\!N}$, which is defined recursively as follows: $I_0 = \{\}$ and $I_{\alpha+1} = least^3_v(\Pi^{\texttt{nr}}_p/^{\texttt{WFSX}}I_\alpha)$. In [3], it is shown that $I_\alpha \subseteq I_{\alpha+1}$, for $\alpha \in I\!N$. Let λ be the smallest ordinal s.t. $I_\lambda = I_{\lambda+1}$. In [3], it is shown that $I_\lambda = W^{\texttt{nr}}_p$.

Consider now the sequence $\{J_\alpha\}_{\alpha \in I\!N}$, which is defined recursively as follows: $J_0 = \{\}$ and $J_{\alpha+1} = Coh(least^3_v(\Pi/^{\texttt{WFSX}}J_\alpha))$. Let λ' be the smallest ordinal s.t. $J_{\lambda'} = J_{\lambda'+1}$. In [3], it is shown that $J_{\lambda'} = N_{\mathsf{W}}$.

To prove the *Lemma*, it is enough to show that $I_\alpha = \{\tau(A) \mid A \in J_\alpha \cap (D \cup \sim D)\}$, for all $\alpha \in I\!N$. We will prove the statement by induction. For $\alpha = 0$, the statement holds. Assume now that the statement holds for $\alpha = k$, where $k \in I\!N$. We will show that it also holds for $\alpha = k + 1$. Let $\neg L' \in J_k \cap D$. Due to the Coh operator in the definition of J_k, it follows that $\sim L' \in J_k \cap \sim D$. Thus, $\tau(\sim L') \in I_k$. Therefore, $I_k(\tau(L')) = 0$.

Since $I_k \subseteq I_{k+1}$, in follows that $I_{k+1}(\tau(L')) = 0$. Now, due to (ii) in the definition of D and since $I_k = \{\tau(A) \mid A \in J_k \cap (D \cup \sim D)\}$, it follows that $least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k) = \{\tau(A) \mid A \in least_v^3(\Pi/^{\mathtt{WFSX}} J_k) \cap (D \cup \sim D)\}$. Obviously, $least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k) \subseteq \{\tau(A) \mid A \in Coh(least_v^3(\Pi/^{\mathtt{WFSX}} J_k)) \cap (D \cup \sim D)\}$.

Let $L' \in \{\tau(A) \mid A \in Coh(least_v^3(\Pi/^{\mathtt{WFSX}} J_k)) \cap D\}$. Then, $L' \in \{\tau(A) \mid A \in least_v^3(\Pi/^{\mathtt{WFSX}} J_k) \cap D\}$. Thus, $L' \in least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k)$. Now, let $\sim L' \in \{\tau(A) \mid A \in Coh(least_v^3(\Pi/^{\mathtt{WFSX}} J_k)) \cap \sim D\}$. It holds that $L' \notin \{\tau(A) \mid A \in Coh(least_v^3(\Pi/^{\mathtt{WFSX}} J_k) \cap D)\}$, because otherwise N_{W} is inconsistent. Therefore, $L' \notin least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k)$. Since for all $L'' \in D$, $W_p^{\mathtt{nr}}(\tau(L'')) = 1$ or $W_p^{\mathtt{nr}}(\tau(L'')) = 0$, it follows that $\sim L' \in least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k)$. Therefore, it holds $\{\tau(A) \mid A \in Coh(least_v^3(\Pi/^{\mathtt{WFSX}} J_k)) \cap (D \cup \sim D)\} \subseteq least_v^3(\Pi_p^{\mathtt{nr}}/^{\mathtt{WFSX}} I_k)$. Now it follows that $I_{k+1} = J_{k+1}$. $\square$

End of Lemma

Based on the fact that for all $L' \in D$, $W_p^{\mathtt{nr}}(\tau(L')) = 1$ or $W_p^{\mathtt{nr}}(\tau(L')) = 0$ and since Π_p contains the rule $\neg Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n) \leftarrow \sim n_s{:}\mathtt{c}\_p(c_1, ..., c_n)$, it follows that $W_p^{\mathtt{nr}}(\tau(L')) = 1$ or $W_p^{\mathtt{nr}}(\tau(\neg L')) = 1$. Now based on the *Lemma*, it follows that $N_{\mathsf{W}}(\tau_s^{\mathtt{c}}(L)) = 1$ or $N_{\mathsf{W}}(\tau_s^{\mathtt{c}}(\neg L)) = 1$. It follows from this that $W_s^{\mathtt{c}}(L) = 1$ or $W_s^{\mathtt{c}}(\neg L) = 1$. Therefore, from Corollary 1, it follows statement 1.

Now assume that $mode_s^{\mathtt{D}}(p) = \mathtt{c}^-$. The proof of statement 1. follows similarly to the previous case, with the difference that D is defined as the smallest set s.t. (i) $\tau_{s'}^{\mathtt{c}}(L) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body_r^+ \cup Body_r^- \subseteq D$. Additionally, note that Π_p now contains the rule $Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n) \leftarrow \sim\neg Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n)$.

2) *Case* $\mathtt{SEM} = \mathtt{mAS}$ *and Case* $\mathtt{SEM} = \mathtt{mWFS}$: These cases are proved as the corresponding cases in 1) but now Π_p contains (i) the rule $\neg Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n) \leftarrow Nam_s{:}\mathtt{c}\_cxt(c_1, ..., c_n), \sim Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n)$, if $mode_s^{\mathtt{D}}(p) = \mathtt{c}^+$, and (ii) the rule $Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n) \leftarrow Nam_s{:}\mathtt{c}\_cxt(c_1, ..., c_n), \sim\neg Nam_s{:}\mathtt{c}\_p(c_1, ..., c_n)$, if $mode_s^{\mathtt{D}}(p) = \mathtt{c}^-$. $\square$

Proposition 16 Let $\mathcal{S}$ be a modular rule base and let $s \in \mathcal{S}$ s.t. $\mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{AS}}(s) \neq \{\}$. Additionally, let $p \in Pred_s^{\mathtt{D}}$ s.t. p is $\mathtt{c}$-stratified in s w.r.t. $\mathcal{S}$, and let $L \in [p]_{\mathcal{S}} \cup \sim[p]_{\mathcal{S}}$. It holds that: $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ iff $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$.

Proof:

Case $mode_s^{\mathtt{D}}(p) = \mathtt{c}^+$: Let $L = p(c_1, ..., c_n)$ or $L = \sim p(c_1, ..., c_n)$ or $L = \neg p(c_1, ..., c_n)$ or $L = \sim\neg p(c_1, ..., c_n)$. In the case that s is contradictory in reasoning mode $\mathtt{c}$ w.r.t. $\mathcal{S}$, it follows from Proposition 3 that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$ and $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. Assume now that s is not contradictory in reasoning mode $\mathtt{c}$ w.r.t. $\mathcal{S}$. We will denote $\Pi_{s,\mathcal{S}}^{\mathtt{c}}$ by Π. Let D be the smallest set s.t. (i) $\tau_s^{\mathtt{c}}(\neg p(c_1, ..., c_n)) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body_r^+ \cup Body_r^- \subseteq D$. We define $\Pi_p = \{r \in [\Pi] \mid Head_r \in D\}$.

Let A be an atom in $\mathtt{HB}_{\Pi_p}$, we define: (i) $\tau(A) = A$, (ii) $\tau(\sim A) = \sim A$, (iii) $\tau(\neg A) = \neg\_A$, and (iv) $\tau(\sim \neg A) = \sim \neg\_A$. Let $\Pi_p^{\mathtt{nr}}$ be the normal logic program that results, if we replace each rule $L_0 \leftarrow L_1, ..., L_n$ in Π_p by $\tau(L_0) \leftarrow \tau(L_1), ..., \tau(L_n)$. It is easy to see that $\Pi_p^{\mathtt{nr}}$ is a normal logic program. Since p is $\mathtt{c}$-stratified in s w.r.t. $\mathcal{S}$, it follows that $\Pi_p^{\mathtt{nr}}$ is stratified. Thus, $\Pi_p^{\mathtt{nr}}$ has a unique stable model. It follows from this that Π_p has a unique answer set. Let $\mathsf{M} \in \mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{AS}}(s)$. Since s is not contradictory in reasoning mode $\mathtt{c}$ w.r.t. $\mathcal{S}$, it follows that M is consistent. Thus, it follows, from Theorem 1, that N_{M} is a consistent answer set of Π. Therefore, $N_{\mathsf{M}} = least_v^2(\Pi/^{\mathtt{AS}} N_{\mathsf{M}})$. It follows, from the statement (ii) in the Definition of D, that $N_{\mathsf{M}} \cap D = least_v^2(\Pi_p/^{\mathtt{AS}}(N_{\mathsf{M}} \cap D))$. It follows from this that $N_{\mathsf{M}} \cap D$ is the unique answer set of Π_p. Additionally, $\Pi_p^{\mathtt{nr}}$ has a well-founded model $W_p^{\mathtt{nr}}$ and since $\Pi_p^{\mathtt{nr}}$ is stratified, it holds (i) for all $L' \in D$, $W_p^{\mathtt{nr}}(\tau(L')) = 1$ or $W_p^{\mathtt{nr}}(\tau(L')) = 0$, and (ii) $W_p^{\mathtt{nr}} = \{\tau(A) \mid A \in (N_{\mathsf{M}} \cap D) \cup \sim(D - N_{\mathsf{M}})\}$.

We will denote $\mathtt{mWF}_{s,\mathtt{c}}^{\mathcal{S}}$ by W. Since $\mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{AS}}(s) \neq \{\}$, it follows from Theorem 1 that Π has an answer set. Thus, Π has a partial stable model. Now it follows from Theorem 1 that $\mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{EAS}}(s) \neq \{\}$. Thus, $\mathsf{W} \in \mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{EAS}}(s)$. Since s is not contradictory in reasoning mode $\mathtt{c}$ w.r.t. $\mathcal{S}$, it follows from Proposition 4 that W is consistent. Thus, it follows from Proposition 7 that N_{W} is the well-founded model of Π (according to $\mathtt{WFSX}$ semantics) and that N_{W} is consistent. It follows from the *Lemma* in the proof of Proposition 15 that it holds $W_p^{\mathtt{nr}} = \{\tau(A) \mid A \in N_{\mathsf{W}} \cap (D \cup {\sim}D)\}$. Thus, $(N_{\mathsf{M}} \cap D) \cup {\sim}(D - N_{\mathsf{M}}) = N_{\mathsf{W}} \cap (D \cup {\sim}D)$.

Assume now that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$. Then, $M_s^{\mathtt{c}}(L) = 1$. Thus, $\tau_s^{\mathtt{c}}(L) \in (N_{\mathsf{M}} \cap D) \cup {\sim}(D - N_{\mathsf{M}})$. Therefore, $\tau_s^{\mathtt{c}}(L) \in N_{\mathsf{W}} \cap (D \cup {\sim}D)$. Thus, $W_s^{\mathtt{c}}(L) = 1$. Since $\mathtt{mWF}_{s,\mathtt{c}}^{\mathcal{S}} = \mathsf{W}$, it follows from Corollary 1 that $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$.

Reversely, assume that $s \models_{\mathcal{S}}^{\mathtt{mWFS}} L$. Since $\mathtt{mWF}_{s,\mathtt{c}}^{\mathcal{S}} = \mathsf{W}$, it follows from Corollary 1 that $W_s^{\mathtt{c}}(L) = 1$. Therefore, $\tau_s^{\mathtt{c}}(L) \in N_{\mathsf{W}} \cap (D \cup {\sim}D)$. Thus, $\tau_s^{\mathtt{c}}(L) \in (N_{\mathsf{M}} \cap D) \cup {\sim}(D - N_{\mathsf{M}})$. Therefore, $M_s^{\mathtt{c}}(L) = 1$. From this, it follows that for all $\mathsf{M} \in \mathcal{M}_{\mathtt{c},\mathcal{S}}^{\mathtt{AS}}(s)$, it holds that $M_s^{\mathtt{c}}(L) = 1$. It now follows that $s \models_{\mathcal{S}}^{\mathtt{mAS}} L$.

Case $mode_s^{\mathtt{D}}(p) = \mathtt{c}^-$: The proof of this case follows similarly to the previous case with the difference that D is defined as the smallest set s.t. (i) $\tau_{s'}^{\mathtt{c}}(p(c_1, ..., c_n)) \in D$ and (ii) if there exists $r \in [\Pi]$ with $Head_r \in D$ then $Body_r^+ \cup Body_r^- \subseteq D$. □

References

1. J. J. Alferes, C. V. Damásio, and L. M. Pereira. A Logic Programming System for Non-monotonic Reasoning. *Journal of Automated Reasoning*, 14(1):93–147, 1995.
2. J. J. Alferes and L. M. Pereira. On Logic Program Semantics with Two Kinds of Negation. In *5th International Joint Conference and Symposium on Logic Programming (JICSLP-1992)*, pages 574–588, 1992.
3. J. J. Alferes and L. M. Pereira. *Reasoning with Logic Programming.* Springer-Verlag New York, Inc., 1996.
4. A. Analyti, G. Antoniou, and C. V. Damásio. A Principled Framework for Modular Web Rule Bases and Its Semantics. In *11th International Conference on Principles of Knowledge Representation and Reasoning (KR-2008)*, pages 390–400. AAAI press, 2008.
5. A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner. Extended RDF as a Semantic Foundation of Rule Markup Languages. *Journal of Artificial Intelligence Research (JAIR)*, 32:37–94, 2008.
6. K. Apt, H. Blair, and A. Walker. Towards a Theory of Declarative Knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–142. Morgan Kaufmann, 1988.
7. U. Aßmann, S. Berger, F. Bry, T. Furche, J. Henriksson, and P.-L. Patranjan. A Generic Module System for Web Rule Languages: Divide and Rule. In *International Symposium on Advances in Rule Interchange and Applications, (RuleML'07)*, pages 63–77, 2007.
8. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
9. J. Bao, D. Caragea, and V. Honavar. On the Semantics of Linking and Importing in Modular Ontologies. In *5th International Semantic Web Conference (ISWC'06)*, pages 72–86, 2006.
10. C. Baral and M. Gelfond. Logic Programming and Knowledge Representation. *Journal of Logic Programming*, 19/20:73–148, 1994.
11. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.
12. A. Bochman. *Explanatory Nonmonotonic Reasoning.* Advances in Logic. World Scientific, 2005.
13. A. Borgida and L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal Data Semantics*, 1:153–184, 2003.
14. G. Brewka and T. Eiter. Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In *22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, pages 385–390, 2007.

15. G. Brewka, F. Roelofsen, and L. Serafini. Contextual Default Reasoning. In *20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 268–273, 2007.
16. M. Bugliesi, E. Lamma, and P. Mello. Modularity in Logic Programming. *Journal of Logic Programming*, 19/20:443–502, 1994.
17. P. Cabalar, S. P. Odintsov, and D. Pearce. Strong Negation in Well-Founded and Partial Stable Semantics for Logic Programs. In *2nd International Joint Conference, 10th Ibero-American Conference on AI, 18th Brazilian AI Symposium (IBERAMIA-SBIA 2006)*, pages 592–601, 2006.
18. A. Cortés-Calabuig, M. Denecker, O. Arieli, B. V. Nuffelen, and M. Bruynooghe. On the Local Closed-World Assumption of Data-Sources. In *17th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'05)*, pages 333–334, 2005.
19. C. V. Damásio, A. Analyti, G. Antoniou, and G. Wagner. Supporting Open and Closed World Reasoning on the Web. In *4th International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'06)*, pages 149–163, 2006.
20. C. V. Damásio and L. M. Pereira. A Survey of Paraconsistent Semantics for Logic Programs. In D. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, Reasoning with Actual and Potential Contradictions. Coordinated by P. Besnard and A. Hunter, pages 241–320. Kluwer Academic Publishers, 1998.
21. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.
22. Duerst and Suignard. Internationalized Resource Identifiers. RFC 3987, Jan. 2005.
23. T. Eiter, G. Gottlob, and H. Veith. Modular Logic Programming and Generalized Quantifiers. In *4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'97)*, pages 290–309, 1997.
24. T. Eiter, G. Ianni, T. Krennwallner, and A. Polleres. Rules and Ontologies for the Semantic Web. In *Reasoning Web*, Lecture Notes in Computer Science, pages 1–53, 2008.
25. T. Eiter, G. Ianni, R. Schindlauer, and H. Tompits. A Uniform Integration of Higher-Order Reasoning and External Evaluations in Answer-Set Programming. In *19th Intern. Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 90–96, 2005.
26. O. Etzioni, K. Golden, and D. S. Weld. Sound and Efficient Closed-World Reasoning for Planning. *Artificial Intelligence*, 89(1-2):113–148, 1997.
27. M. Fitting. Bilattices and the Semantics of Logic Programming. *Journal of Logic Programming*, 11(1&2):91–116, 1991.
28. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett. No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web. In *1st European Semantic Web Symposium (ESWS 2004)*, pages 342–356, 2004.
29. A. V. Gelder, K. A. Ross, and J. S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
30. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. A. Bowen, editors, *5th International Conference on Logic Programming (ICLP'88)*, pages 1070–1080. MIT Press, 1988.
31. M. Gelfond and V. Lifschitz. Logic programs with Classical Negation. In *7th International Conference on Logic Programming (ICLP'90)*, pages 579–597, 1990.
32. M. Gelfond and V. Lifschitz. Classical Negation in Logic programs and Disjunctive Databases. *New Generation Computing*, 9:365–385, 1991.
33. B. C. Grau and O. Kutz. Modular Ontology Languages Revisited. In *IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa'07)*, 2007.

34. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *12th International World Wide Web Conference (WWW2003)*, pages 48–57, 2003.
35. J. Heflin, J. A. Hendler, and S. Luke. SHOE: A Blueprint for the Semantic Web. In *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, pages 29–63. MIT Press, 2003.
36. J. Heflin and H. Munoz-Avila. LCW-Based Agent Planning for the Semantic Web. In *AAAI Workshop on Ontologies and the Semantic Web*, pages 63–70, 2002.
37. I. Horrocks, O. Kutz, and U. Sattler. The Even More Irresistible SROIQ. In *10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 57–67, 2006.
38. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic Web. In *18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*, pages 792–797, 2002.
39. G. Ianni, A. Martello, C. Panetta, and G. Terracina. Efficiently Querying RDF(S) Ontologies with Answer Set Programming. *Journal of Logic and Computation*, 19:671 – 695, 2009.
40. Linked Data - Connect Distributed Data across the Web. Available at `http://linkeddata.org/`.
41. M. Kifer, J. de Bruijn, H. Boley, and D. Fensel. A Realistic Architecture for the Semantic Web. In *1st International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'05)*, pages 17–29, 2005.
42. M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843, 1995.
43. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of Abstract Description Systems. *Artificial Intelligence*, 156(1):1–73, 2004.
44. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006.
45. I. Niemelä and P. Simons. Smodels - An Implementation of the Stable Model and Well-Founded Semantics for Normal LP. In *4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'97)*, pages 421–430, 1997. Available at `http://www.tcs.hut.fi/Software/smodels/`.
46. E. Oikarinen and T. Janhunen. Modular Equivalence for Normal Logic Programs. In *7th European Conference on Artificial Intelligence (ECAI-2006)*, pages 412–416, 2006.
47. Production Rule Representation (PRR), OMG specification, version 1.0, 2007.
48. OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009, Edited by W3C OWL Working Group, Available at `http://www.w3.org/TR/owl2-overview/`.
49. OWL 2 Web Ontology Language Profiles. W3C Recommendation 27 October 2009, Edited by Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz, Available at `http://www.w3.org/TR/owl2-profiles/`.
50. OWL 2 RL in RIF. W3C Working Draft 1 October 2009, Edited by Dave Reynolds, Available at `http://www.w3.org/TR/rif-owl-rl/`.
51. A. Paschke and H. Boley. Rule Markup Languages and Semantic Web Rule Languages. In *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, chapter 1, pages 1–24. IGI Global, May 2009.
52. D. Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47(1-2):3–41, 2006.
53. L. M. Pereira and J. J. Alferes. Well Founded Semantics for Logic Programs with Explicit Negation. In *10th European Conference on Artificial Intelligence (ECAI-1992)*, pages 102–106, 1992.

54. A. Polleres, C. Feier, and A. Harth. Rules with Contextually Scoped Negation. In *3rd European Semantic Web Conference (ESWC-2006)*, pages 332–347, 2006.
55. E. Pontelli, T. C. Son, and C. Baral. A Framework for Composition and Inter-operation of Rules in the Semantic Web. In *2nd International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML'06)*, pages 39–50, 2006.
56. T. C. Przymusinski. Stable Semantics for Disjunctive Programs. *New Generation Comput.*, 9(3/4):401–424, 1991.
57. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004. Edited by G. Klyne and J. J. Carroll, Available at `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210`.
58. RDF Semantics. W3C Recommendation, 10 February 2004. Edited by Patrick Hayes, Available at `http://www.w3.org/TR/2004/ REC-rdf-mt-20040210/`.
59. RIF: The Rule Interchange Working Group Charter. Available at `http://www.w3.org/2005/rules/wg/charter`.
60. RIF Framework for Logic Dialects. W3C Candidate Recommendation 1 October 2009, Edited by Harold Boley and Michael Kifer, Available at `http://www.w3.org/TR/rif-fld/`.
61. RIF Production Rule Dialect. W3C Candidate Recommendation 1 October 2009, Edited by Christian de Sainte Marie, Adrian Paschke, and Gary Hallmark, Available at `http://www.w3.org/TR/rif-prd/`. Work in progress.
62. RIF Use Cases and Requirements. W3C Working Draft 18 December 2008, Edited by Adrian Paschke, David Hirtle, Allen Ginsberg, Paula-Lavinia Patranjan, and Frank McCabe, Available at `http://www.w3.org/TR/rif-ucr/`.
63. The rule markup initiative (ruleml). Available at `http://www.ruleml.org`.
64. K. F. Sagonas, T. Swift, and D. S. Warren. XSB as an Efficient Deductive Database Engine. In R. T. Snodgrass and M. Winslett, editors, *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 442–453. ACM Press, 1994. Available at `http://xsb.sourceforge.net/`.
65. C. Sakama and K. Inoue. Paraconsistent Stable Semantics for extended disjunctive programs. *Journal of Logic and Computation*, 5(3):265–285, 1995.
66. L. Serafini, A. Borgida, and A. Tamilin. Aspects of Distributed and Modular Ontology Reasoning. In *19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 570–575, 2005.
67. M. Sintek and S. Decker. TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In *1st International Semantic Web Conference (ISWC'02)*, pages 364–378, 2002.
68. G. Wagner. A Database Needs Two Kinds of Negation. In *3rd Symposium on Mathematical Fundamentals of Database and Knowledge Base Systems (MFDBS'91)*, pages 357–371. Springer-Verlag, 1991.
69. G. Wagner, A. Giurca, I.-M. Diaconescu, G. Antoniou, A. Analyti, and C. V. Damasio. Reasoning on the Web with Open and Closed Predicates. In *3rd International Workshop on Applications of Logic Programming to the (Semantic) Web and Web Services (ALPSWS 2008), in conjunction with ICLP'08*, pages 71–84, 2008.
70. G. Yang, M. Kifer, and C. Zhao. Flora-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In *2nd Int. Conf. on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE'03)*, pages 671–688, 2003.