

Compositionality of normal open logic programs

Sofie Verbaeten, Marc Denecker, Danny De Schreye

Report CW254, July 1997

Department of Computing Science, K.U.Leuven

Abstract

Compositionality of programs is an important concern in knowledge representation and software development. In the context of Logic Programming, up till now, the issue has mostly been studied for *definite* programs only.

Here, we study compositionality in the context of *normal open logic programming*. This is a very expressive logic for knowledge representation of uncertainty and incomplete knowledge on concepts and on problem domain, in which the compositionality issue turns up very naturally. The semantics of the logic is a generalisation (allowing non-Herbrand interpretations) of the well-founded semantics.

We provide a number of results which offer different sufficient conditions under which the models of the composition of two theories can be related to the intersection of the models of the composing theories. In particular, under these conditions, logical consequence will be preserved under composition.

Keywords : Logic Programming, Knowledge representation.

CR Subject Classification : I.2.3

1 Introduction

The compositionality issue arises in a situation where two or more experts cooperate to axiomatise a certain domain of application; they have more or less disjunct subdomains of expertise and they represent their expert knowledges independently in distinct logical theories. Ideally, once this stage is finished the problem arises how to combine the *knowledge modules* in one united theory. In general, there may be different modes in which the modules can be composed, but certainly the most important one seems to take the *logical conjunction* of their knowledges; i.e. to construct a new theory which contains exactly the sum of the knowledges of the component modules.

By the nature of the situation, the modules designed by the experts are incomplete representations of the problem domain; they contain uncertainty about the problem domain. Typically, experts will have two sorts of ignorance in their theories: ignorance on relations, some of which are defined by the other experts; ignorance about the objects some of which are defined by other experts. A suitable logic to represent knowledge modules should allow to represent these forms of uncertainty.

Here, we investigate the compositionality issue in the logic of *normal Open Logic Programs* (OLP) and *First Order Logic* (FOL) [10]. [10] presents this logic from a Knowledge Representation perspective and illustrates its suitability for representing uncertainty of similar nature as cooperating experts have to face: incomplete knowledge on the definitions of certain concepts and on the problem domain.

A theory \mathcal{T} in this logic is a pair $(\mathcal{T}_d, \mathcal{T}_c)$ of a First Order Logic theory \mathcal{T}_c and a normal open logic program \mathcal{T}_d , i.e. a set of normal program clauses $p \leftarrow q_1, \dots, q_n, \neg r_1, \dots, \neg r_m$. Normal open logic programs will further on also be called logic programs briefly. \mathcal{T}_d represents a set of *definitions*. Predicates occurring in the head of a clause of \mathcal{T}_d are called *defined*. The other predicates, occurring at the most in the body of program clauses, are called *open*. Intuitively, they represent concepts for which no definitions are given. Partial knowledge about these predicates can be expressed in the set of FOL axioms \mathcal{T}_c .

The model semantics of OLP-FOL is an extension of the well-founded semantics [21] and of the extended well-founded semantics [19] and was defined in [11]. This logic has a *possible state semantics*, that is, a model correspond to a state in which the problem domain might occur according to the (incomplete) expert knowledge (and not a *belief set*, a set of believed atoms, as in answer set semantics of Extended Logic Programming). At the level of the semantics, uncertainty on the definition of a concept is modeled by allowing models which give to the open predicates an arbitrary interpretation which satisfies the set of FOL axioms \mathcal{T}_c (and not e.g. by having truth value *unknown* for these open predicates as in a belief set semantics). Uncertainty on the level of the domain of discourse (no Domain Closure) is modeled by allowing general, non-Herbrand models.

Compositionality of logic programs has been investigated by a number of researchers. We refer to the discussion section for explicit references. In the context of OLP-FOL, the problem of correct composition of different independently designed modules has a natural formulation which differs from the formalisation as presented in much of the existing research. In the context of a logic with *possible state semantics*, the *compositionality criterion* that a logic theory \mathcal{T} contains precisely the sum of the knowledges in the modules $\mathcal{T}_1, \mathcal{T}_2$ has a natural formalisation: that the class of models of \mathcal{T} is precisely the intersection of the classes of models of \mathcal{T}_1 and of \mathcal{T}_2 . Note that this criterion is the one expressed by the semantics of classical logic conjunction: models of the conjunction $F \wedge G$ of arbitrary FOL formulas F, G are precisely the models of F and of G .

Here, we investigate conditions under which the simple union of two OLP-FOL theories $\mathcal{T}_1, \mathcal{T}_2$ yields a theory which satisfies the natural compositionality criterion. For a more formal description of the problem investigated, we need the following notions. The composition of two OLP-FOL theories $\mathcal{T}_1 = (\mathcal{T}_{1d}, \mathcal{T}_{1c})$ and $\mathcal{T}_2 = (\mathcal{T}_{2d}, \mathcal{T}_{2c})$ is defined as the theory $\mathcal{T}_1 \cup \mathcal{T}_2 = (\mathcal{T}_{1d} \cup \mathcal{T}_{2d}, \mathcal{T}_{1c} \cup \mathcal{T}_{2c})$. Also, given a class of interpretations \mathcal{J} , the class of members of \mathcal{J} which are models of the OLP-FOL theory \mathcal{T} is denoted $\text{Mod}^{\mathcal{J}}(\mathcal{T})$. Using these notions, the compositionality problem considered here is formalised as follows.

- Given is a class \mathcal{J} of interpretations, representing a priori knowledge shared by the experts. In general, this class may be the class of models of a logical theory representing the a priori knowledge. E.g. this theory can describe knowledge on the domain of discourse (a Domain Closure Axiom) or simple programming concepts such as definitions of membership of lists, appending of lists, etc.,
- given also is a pair of OLP-FOL theories $\mathcal{T}_1, \mathcal{T}_2$ representing the modules of the experts with non-intersecting sets of defined predicates,

we investigate conditions on $\mathcal{T}_1, \mathcal{T}_2$ such that:

$$\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2).$$

After section 2, which recalls the semantics of OLP-FOL from [11], section 3 gives us a first result, stating that for *correct* theories, the class of models of the composition is contained in the intersection of the classes of models of the two separate theories. In section 4, by using the notion of justification, we give a very general condition, the justification condition, on \mathcal{T}_1 and \mathcal{T}_2 to obtain the equality $\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2)$. In the next two sections (5, 6), some less general, but more syntactical conditions are given; in section 5 for the propositional case, in section 6 for the predicate case. In 6.2, for instance, we study conservative extensions. We conclude in section 7 with a discussion on some related works.

2 OLP-FOL

We assume familiarity with basic concepts of logic and logic programming such as logical languages \mathcal{L} , atoms, literals, (normal) program clauses or rules based on \mathcal{L} , ground instances of rules w.r.t. a language \mathcal{L} , 2-valued and 3-valued interpretations, Herbrand interpretations of \mathcal{L} . We refer to [17]. We assume some familiarity with the well-founded semantics [21] as well. We introduce some auxiliary concepts. Each language \mathcal{L} is assumed to contain propositional predicates \top and \perp ; in each interpretation I of \mathcal{L} , \top is true and \perp is false. $\mathcal{H}_{\mathcal{L}}$ (or \mathcal{H} if \mathcal{L} is clear from the context) denotes the class of all Herbrand interpretations of the language \mathcal{L} . Atomic rules are denoted $A \leftarrow \top$. Given a language \mathcal{L} and an interpretation I with domain D , define the language \mathcal{L}_D as the extension of \mathcal{L} by adding all elements of D as constants to \mathcal{L} . A literal of the form $p(d_1, \dots, d_n)$ or $\neg p(d_1, \dots, d_n)$, where $d_1, \dots, d_n \in D$ is called a *fact*. For a ground literal $F = p(t_1, \dots, t_n)$ in \mathcal{L}_D , $\tilde{I}(F)$ denotes the fact $p(\tilde{I}(t_1), \dots, \tilde{I}(t_n))$, where \tilde{I} is the mapping which assigns to each ground term in \mathcal{L}_D the corresponding domain element of the interpretation I . The truth function of I (i.e. the function which maps positive facts to $\{\mathbf{f}, \mathbf{u}, \mathbf{t}\}$) is denoted by \mathcal{H}_I . We describe a truth function as a set of tuples of facts with truth value (e.g. $\{p^{\mathbf{f}}, q^{\mathbf{u}}, r^{\mathbf{t}}\}$, meaning that $\mathcal{H}_I(p) = \mathbf{f}$, $\mathcal{H}_I(q) = \mathbf{u}$ and $\mathcal{H}_I(r) = \mathbf{t}$). 2-valued Herbrand interpretations are denoted in the conventional notation, as a subset of the Herbrand base.

A theory \mathcal{T} in the OLP-FOL logic is a pair $(\mathcal{T}_d, \mathcal{T}_c)$ with \mathcal{T}_d a set of normal program clauses $p \leftarrow q_1, \dots, q_n, \neg r_1, \dots, \neg r_m$ (with $p, q_1, \dots, q_n, r_1, \dots, r_m$ atoms) and \mathcal{T}_c a set of FOL formulas. A predicate p is *defined* in \mathcal{T}_d or in \mathcal{T} iff p occurs in the head of a rule of \mathcal{T}_d (it is possible that this rule is of the form $p(t_1, \dots, t_n) \leftarrow \perp$). *Open* predicates are predicates of \mathcal{L} which are not defined. An open logic program \mathcal{T}_d (or a theory \mathcal{T}) is *complete* if each predicate symbol of \mathcal{L} except equality $=$, \top and \perp is defined, otherwise it is *incomplete*.

The grounding of an open logic program \mathcal{T}_d w.r.t. a given 3-valued interpretation I is denoted as $Ground_I(\mathcal{T}_d)$ and is defined as the following set of program clauses:

$$\begin{aligned} & \{ \tilde{I}(F) \leftarrow \tilde{I}(F_1), \dots, \tilde{I}(F_n) \mid F \leftarrow F_1, \dots, F_n \text{ ground instance of a rule in } \mathcal{T}_d \text{ in } \mathcal{L}_D \} \\ & \cup \{ F \leftarrow \top \mid F \text{ is a positive fact of an open predicate and } F \text{ is true in } I \} \end{aligned}$$

The semantics of OLP-FOL is based on the concept of *justification*. A justification can be seen as a mathematical object justifying the truth value of facts in terms of truth values of other facts. The basic theorem of this paper, which gives weakest conditions under which two OLP-FOL theories can be composed, uses the concept of justification. For a detailed discussion we refer to [11]. The rest of this section is structured as follows. We define the concepts of justification and justification semantics. We show how the justification semantics is an extension of the well-founded semantics [21] based on general interpretations.

Below, we denote the complement of a fact F by $\sim F$; i.e. if F is a positive fact, then $\sim F$ denotes $\neg F$; vice versa $\sim \neg F$ denotes F . We define $\sim \top = \perp$ and vice versa.

We now define the concepts of elementary justification and justification given an open logic program \mathcal{T}_d based on \mathcal{L} and an interpretation I of \mathcal{L} .

Definition 2.1 *Given is a defined positive fact F in I .*

For any rule $F \leftarrow F_1, \dots, F_n \in Ground_I(\mathcal{T}_d)$ we call $\{F_1, \dots, F_n\}$ an elementary justification for F . If no such ground instance exists for F , then we call $\{\perp\}$ an elementary justification for F .

Each positive defined fact has an elementary justification. Also, an elementary justification is never empty (recall atomic rules are denoted as $A \leftarrow \top$). An elementary justification is always finite. The concept of an elementary justification can also be defined for negative facts.

Definition 2.2 *A set J is called an elementary justification for a negative fact $\neg F$ of a defined predicate iff each elementary justification J^+ of F contains a fact F' such that $\sim F' \in J$.*

Analogously as for positive facts, each negative fact $\neg F$ has an elementary justification and an elementary justification is never empty. It can be infinite.

Definition 2.3 *A justification J for F (given \mathcal{T}_d and I) is a (possibly infinite) tree of facts with F in the root. Each non-leaf node contains a defined fact F' such that the set of direct descendants of the node is an elementary justification for F' and no leaf contains a defined fact.*

The leaves of a justification are \top , \perp or positive or negative open facts. A justification of a fact is always defined w.r.t. a logic program \mathcal{T}_d . Therefore, when necessary, we explicitly talk about a justification in \mathcal{T}_d rather than a justification.

A *branch* in a justification J is a maximal sequence of facts (F_0, F_1, \dots) with F_0 in the root of J , and each F_i a direct descendant of F_{i-1} in J . A *positive* (resp. *negative*) *loop* is a branch with an infinite number of positive (resp. negative) facts and a finite number of negative (resp. positive) facts. A *loop over negation* is a branch with an infinite number of positive and negative facts. Next we define the value of a justification as a measure for its success.

Definition 2.4 *Let I be an interpretation.*

Let B be a branch in a justification. If B is finite and has F as leaf then the value of B under I is the truth value of F under I . With respect to infinite branches, we define the value of a positive loop as \mathbf{f} , the value of a loop over negation as \mathbf{u} and the value of a negative loop as \mathbf{t} . We denote the value of B under I by $\text{val}_I(B)$.

Let J be a justification. The value of J under I is $\min\{\text{val}_I(B) \mid B \text{ is a branch of } J\}$. We denote J 's value by $\text{val}_I(J)$. J is false, weak, strong under I if $\text{val}_I(J)$ is \mathbf{f} , \mathbf{u} , \mathbf{t} respectively.

The essential idea in the justification semantics is that an interpretation is a justified model of a logic program \mathcal{T}_d iff for each defined positive fact F , its truth value is equal to the value of its most successful justification. We call this value the *supported value* of F and denote it by $SV^{\mathcal{T}_d}(I, F)$. Formally, $SV^{\mathcal{T}_d}(I, F) = \max\{\text{val}_I(J) \mid J \text{ is a justification of } F \text{ in } \mathcal{T}_d\}$.

Definition 2.5 *A justified model of a logic program \mathcal{T}_d is an interpretation I of \mathcal{L} such that for every positive defined fact F :*

$$\mathcal{H}_I(F) = SV^{\mathcal{T}_d}(I, F).$$

Moreover, the interpretation of open predicates is 2-valued.

A justified model of a theory \mathcal{T} consisting of a logic program \mathcal{T}_d and FOL axioms \mathcal{T}_c is a justified model of \mathcal{T}_d and a model of \mathcal{T}_c in the normal FOL sense.

We extend the notion of supported value to open facts and negative facts. The supported value under I of an open fact F is defined as $\mathcal{H}_I(F)$. The supported value under I of a negative defined fact F can be defined analogously as for positive facts: it is equal to the value of the best justification of F .

The following theorem asserts that a fact and its negation have inverse supported values.

Theorem 2.1 *Let \mathcal{T}_d be a logic program, I an interpretation and F a fact. Then $SV^{\mathcal{T}_d}(I, F) = SV^{\mathcal{T}_d}(I, \sim F)^{-1}$.*

A direct consequence of this theorem is that in a model the truth value of negative facts is also equal to their supported value. This restores the asymmetry between positive and negative facts in the definition of model, which requires only that positive facts have truth value equal to their supported value.

The next theorem asserts that each logic program is consistent w.r.t. the justification semantics. An interpretation I is called an incomplete interpretation for some subset of the predicates of \mathcal{L} if it is a partial interpretation of which the truth function is defined only for the specified predicates.

Theorem 2.2 *Given is a logic program \mathcal{T}_d and an incomplete 2-valued interpretation I for the open predicates of \mathcal{L} only. There exists a unique justified model of \mathcal{T}_d extending I .*

The semantics of the OLP-FOL logic is an extension of well-founded semantics [21] based on general interpretations. Well-founded semantics can be easily *lifted* to general interpretations by using the notion of grounding w.r.t. a given 3-valued interpretation. We will now give an alternative characterisation of a model of a theory. Theorem 2.3 is a trivial extension of results proven in [11].

Definition 2.6 *An interpretation M is a model of \mathcal{T}_d iff M is a well-founded model of the grounding of \mathcal{T}_d w.r.t. M . M is a model of $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ iff M is a model of \mathcal{T}_d and M is a model of \mathcal{T}_c in the normal FOL sense.*

Theorem 2.3 *An interpretation is a justified model of \mathcal{T} iff it is a model of \mathcal{T} in the sense of definition 2.6.*

Note that the grounding of an open logic program \mathcal{T}_d w.r.t. to a Herbrand interpretation corresponds to the conventional notion of the grounding of a logic program. It trivially follows that the Herbrand model of a complete logic program is the well-founded model. Also, a Herbrand model of an incomplete logic program is an extended well-founded model [19].

An open logic program \mathcal{T}_d is interpreted as a definition of the defined predicates in terms of open predicates. The occurrence of an undefined fact in a model of \mathcal{T}_d reveals an ambiguity or a local inconsistency in the definition. This motivates the following definition.

Definition 2.7 *Given is an open logic program \mathcal{T}_d and a class of interpretations \mathcal{J} . Then \mathcal{T}_d is called a correct definition (or correct) w.r.t. \mathcal{J} iff each model of \mathcal{T}_d which belongs to \mathcal{J} is 2-valued. A theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ is correct w.r.t. \mathcal{J} iff \mathcal{T}_d is a correct definition w.r.t. \mathcal{J} .*

In case \mathcal{J} is the class of all interpretations, we simply call \mathcal{T} correct.

With $\text{Mod}(\mathcal{T})$ we will denote the class of models of a theory \mathcal{T} . If \mathcal{J} is a class of interpretations, then $\text{Mod}^{\mathcal{J}}(\mathcal{T})$ denotes the class of models of \mathcal{T} which belong to \mathcal{J} (i.e. $\text{Mod}^{\mathcal{J}}(\mathcal{T}) = \text{Mod}(\mathcal{T}) \cap \mathcal{J}$).

Recall the following definition of consequence of a theory.

Definition 2.8 *A formula is a consequence of a theory \mathcal{T} if it is true in every model of \mathcal{T} .*

3 No loss of information

Given is a first order language \mathcal{L} and two theories $\mathcal{T}_1 = (\mathcal{T}_{1d}, \mathcal{T}_{1c})$ and $\mathcal{T}_2 = (\mathcal{T}_{2d}, \mathcal{T}_{2c})$ based on \mathcal{L} . We will suppose, for the rest of the paper, that \mathcal{T}_1 and \mathcal{T}_2 define disjoint sets of predicates. Then all the predicate symbols of \mathcal{L} defined in \mathcal{T}_1 (resp. \mathcal{T}_2) are open in \mathcal{T}_2 (resp. \mathcal{T}_1). Recall that the predicate symbols of \mathcal{L} not defined in \mathcal{T}_1 or \mathcal{T}_2 are open in \mathcal{T}_1 and \mathcal{T}_2 . Denote the union of \mathcal{T}_1 and \mathcal{T}_2 by $\mathcal{T}_1 \cup \mathcal{T}_2 = (\mathcal{T}_{1d} \cup \mathcal{T}_{2d}, \mathcal{T}_{1c} \cup \mathcal{T}_{2c})$ (where \cup is just the normal union of sets; \mathcal{T}_{id} a set of clauses, \mathcal{T}_{ic} a set of FOL axioms, $i \in \{1, 2\}$).

The crucial question in this paper is the following: 'When are the consequences of $\mathcal{T}_1 \cup \mathcal{T}_2$ exactly those formulas which are true in all the interpretations which are model of \mathcal{T}_1 and of \mathcal{T}_2 ?'. Since a formula is a consequence of a theory iff it is true in every model of the theory, we can reformulate this question in terms of models: 'When is $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$?'. The equality doesn't always hold, as is shown by the following example.

Example 3.1

$$\begin{aligned} \mathcal{T}_1 & : \text{alive} \leftarrow \text{not_dead} \\ \mathcal{T}_2 & : \text{not_dead} \leftarrow \text{alive} \end{aligned}$$

Because *not_dead* is open in \mathcal{T}_1 , \mathcal{T}_1 has two models, $I_1 = \{\text{not_dead}^t, \text{alive}^t\}$ and $I_2 = \{\text{not_dead}^f, \text{alive}^f\}$. \mathcal{T}_2 has the same two models (*alive* is open in \mathcal{T}_2). But only the interpretation I_2 is a model of $\mathcal{T}_1 \cup \mathcal{T}_2$. We see that $\neg \text{alive}$ and $\neg \text{not_dead}$ are consequences of the union $\mathcal{T}_1 \cup \mathcal{T}_2$, whereas it is not the case that they are true in every interpretation which is a model of \mathcal{T}_1 and of \mathcal{T}_2 .

In the next sections we give some conditions on the theories \mathcal{T}_1 and \mathcal{T}_2 , so that the equality $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$ holds. But generally, for *correct* theories, $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_1 \cup \mathcal{T}_2$, the inclusion $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$ already holds. This means that formulas which are true in every interpretation which is a model of \mathcal{T}_1 and \mathcal{T}_2 are consequences of $\mathcal{T}_1 \cup \mathcal{T}_2$ (but it is possible that $\mathcal{T}_1 \cup \mathcal{T}_2$ has more consequences, see example 3.1). Before giving this result, we show in the next example that it is not always the case that the union of two correct theories is correct.

Example 3.2

$$\begin{aligned}\mathcal{T}_1 & : \text{alive} \leftarrow \neg \text{dead} \\ \mathcal{T}_2 & : \text{dead} \leftarrow \neg \text{alive}\end{aligned}$$

It is easy to see that $\text{Mod}(\mathcal{T}_1) = \text{Mod}(\mathcal{T}_2) = \{\{\text{dead}^{\mathbf{f}}, \text{alive}^{\mathbf{t}}\}, \{\text{dead}^{\mathbf{t}}, \text{alive}^{\mathbf{f}}\}\}$. So, \mathcal{T}_1 and \mathcal{T}_2 have only 2-valued models. But $\mathcal{T}_1 \cup \mathcal{T}_2$ has a unique model which is 3-valued, namely $\{\text{dead}^{\mathbf{u}}, \text{alive}^{\mathbf{u}}\}$.

Also the converse does not hold. If we know that $\mathcal{T}_1 \cup \mathcal{T}_2$ is a correct theory, we cannot conclude that \mathcal{T}_1 and \mathcal{T}_2 are correct.

Example 3.3

$$\begin{aligned}\mathcal{T}_1 & : \begin{cases} \text{dead} \leftarrow \neg \text{alive}, \neg \text{dancing} \\ \text{alive} \leftarrow \neg \text{dead} \end{cases} \\ \mathcal{T}_2 & : \begin{cases} \text{dancing} \leftarrow \top \end{cases}\end{aligned}$$

$\mathcal{T}_1 \cup \mathcal{T}_2$ has only one model (which is 2-valued), stating that dancing and alive are true and dead is false. The predicate dancing is open in \mathcal{T}_1 , so if we assume that dancing is false, we get the 3-valued model of \mathcal{T}_1 in which dead and alive have truth value \mathbf{u} .

Theorem 3.1 (no loss of information)

Given two theories \mathcal{T}_1 and \mathcal{T}_2 such that $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are correct. Then $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$.

Proof Suppose M is a model of $\mathcal{T}_1 \cup \mathcal{T}_2$. We prove that M is a model of \mathcal{T}_1 . Since M is a model of the FOL axioms $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$, M is a model of \mathcal{T}_{1c} .

We prove that M is a justified model of \mathcal{T}_{1d} .

Let F be a fact. Suppose that F is open in \mathcal{T}_1 . Then $\mathcal{H}_M(F) = SV^{\mathcal{T}_{1d}}(M, F)$ by definition and M is 2-valued.

Suppose that F is defined in \mathcal{T}_1 . We first assume that $\mathcal{H}_M(F) = \mathbf{t}$. Because M is a justified model of $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$, $SV^{\mathcal{T}_{1d} \cup \mathcal{T}_{2d}}(M, F) = \mathcal{H}_M(F)$. So there exists a justification J for F in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ such that $\text{val}_M(J) = \mathbf{t}$. This means that every branch of J has value \mathbf{t} . Let J_1 be the justification for F in \mathcal{T}_{1d} obtained from J by cutting off every branch after its first fact defined in \mathcal{T}_2 . Then it is easy to see that $\text{val}_M(J_1) = \mathbf{t}$ and $SV^{\mathcal{T}_{1d}}(M, F) = \mathbf{t} = \mathcal{H}_M(F)$.

If $\mathcal{H}_M(F) = \mathbf{f}$, then $\mathcal{H}_M(\sim F) = \mathbf{t}$ en by the previous we know that $SV^{\mathcal{T}_{1d}}(M, \sim F) = \mathbf{t}$.

By the consistency theorem (2.1) $SV^{\mathcal{T}_{1d}}(M, F) = \mathbf{f}$, hence $SV^{\mathcal{T}_{1d}}(M, F) = \mathcal{H}_M(F)$. \square

Example 3.2 shows that if for instance $\mathcal{T}_1 \cup \mathcal{T}_2$ is not correct, the inclusion doesn't hold anymore.

Recall that we made the assumption that each predicate symbol is defined in at most one theory \mathcal{T}_1 or \mathcal{T}_2 . This assumption is necessary, as is made clear by the following example.

Example 3.4

$$\begin{aligned} \mathcal{T}_1 & : \begin{cases} \text{not_dead} \leftarrow \text{alive} \\ \text{alive} \leftarrow \text{not_dead} \end{cases} \\ \mathcal{T}_2 & : \begin{cases} \text{alive} \leftarrow \top \end{cases} \end{aligned}$$

\mathcal{T}_1 has only one model in which both *alive* and *not_dead* are false. When adding a second rule for *alive* (stating that *alive* is always true), we get another unique model in which *alive* and *not_dead* are true.

For some theories we are only interested in certain classes of interpretations, like for instance Herbrand interpretations. In this case, we have the following result, which is a generalisation of theorem 3.1.

Theorem 3.2 *Given a class of interpretations \mathcal{J} and two theories \mathcal{T}_1 and \mathcal{T}_2 such that $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are correct w.r.t. \mathcal{J} . Then $\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2)$.*

Proof The proof is analogous to the proof of theorem 3.1. We now start with a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ which belongs to \mathcal{J} . □

Example 3.5

$$\begin{aligned} \mathcal{T}_1 & : \begin{cases} \text{even}(0) \leftarrow \top \\ \text{even}(s(s(X))) \leftarrow \neg \text{odd}(X) \end{cases} \\ \mathcal{T}_2 & : \begin{cases} \text{odd}(s(0)) \leftarrow \top \\ \text{odd}(s(s(X))) \leftarrow \neg \text{even}(X) \end{cases} \end{aligned}$$

It is easily seen that \mathcal{T}_1 and \mathcal{T}_2 are correct. But $\mathcal{T}_1 \cup \mathcal{T}_2$ has a 3-valued (non-Herbrand) model. Indeed, let J_0 be the pre-interpretation with domain the disjoint union $\mathbb{N} \cup \mathbb{Z}$, the interpretation of 0 (constant) $0 \in \mathbb{N}$ and the interpretation of $s/1$ the union of the successor functions on the natural numbers and on the integers. Define the interpretation I with pre-interpretation J_0 as follows: the interpretation of *even*/ 1 is

$$\{\text{even}(2 \times n)^{\mathbf{t}}, \text{even}(2 \times n + 1)^{\mathbf{f}} \mid n \in \mathbb{N}\} \cup \{\text{even}(z)^{\mathbf{u}} \mid z \in \mathbb{Z}\},$$

and the interpretation of *odd*/ 1 is

$$\{\text{odd}(2 \times n)^{\mathbf{f}}, \text{odd}(2 \times n + 1)^{\mathbf{t}} \mid n \in \mathbb{N}\} \cup \{\text{odd}(z)^{\mathbf{u}} \mid z \in \mathbb{Z}\}.$$

Then I is a 3-valued model of $\mathcal{T}_1 \cup \mathcal{T}_2$. It is obvious though, that only Herbrand interpretations are relevant w.r.t. this theory, and it is clear that $\mathcal{T}_1 \cup \mathcal{T}_2$ has only 2-valued Herbrand models. In fact, there is only one, namely

$$\{\text{even}(s^{2n}(0)), \text{odd}(s^{2n+1}(0)) \mid n \in \mathbb{N}\}.$$

Hence, $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_1 \cup \mathcal{T}_2$ are correct w.r.t. \mathcal{H} . Applying theorem 3.2, with $\mathcal{J} = \mathcal{H}$, gives us the inclusion

$$\text{Mod}^{\mathcal{H}}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \text{Mod}^{\mathcal{H}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{H}}(\mathcal{T}_2).$$

As we will see later (theorem 4.2), there's actually an equality here.

4 The justification condition

The aim of this section is to provide a general condition on \mathcal{T}_1 and \mathcal{T}_2 , so that the equality

$$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2) \quad (*)$$

holds, or equivalently, so that the consequences of $\mathcal{T}_1 \cup \mathcal{T}_2$ are exactly those formulas which are true in every interpretation which is a model of \mathcal{T}_1 and of \mathcal{T}_2 . We first give some motivating examples.

4.1 Motivating examples

In the first example the predicates defined in \mathcal{T}_2 are all 'new', that is, they don't occur in \mathcal{T}_{1d} .

Example 4.1

$$\begin{array}{l} \mathcal{T}_1 : \left\{ \begin{array}{l} \text{parent}(X, Y) \leftarrow \text{father}(X, Y) \\ \text{parent}(X, Y) \leftarrow \text{mother}(X, Y) \\ \text{father}(a, b) \leftarrow \top \\ \text{mother}(b, c) \leftarrow \top \end{array} \right. \\ \mathcal{T}_2 : \left\{ \begin{array}{l} \text{anc}(X, Y) \leftarrow \text{parent}(X, Y) \\ \text{anc}(X, Y) \leftarrow \text{anc}(X, Z), \text{parent}(Z, Y) \end{array} \right. \end{array}$$

The first theory \mathcal{T}_1 defines the predicate *parent/2* in terms of the predicate *father/2* and *mother/2*, for which there are given some facts. The second theory \mathcal{T}_2 defines the predicate *anc/2* in terms of itself and *parent/2*.

In this case the equality (*) holds. Looking for instance at Herbrand interpretations, we see that the only Herbrand interpretation which is a model of \mathcal{T}_1 and of \mathcal{T}_2 is the interpretation $I = \{\text{father}(a, b), \text{mother}(b, c), \text{parent}(a, b), \text{parent}(b, c), \text{anc}(a, b), \text{anc}(b, c), \text{anc}(a, c)\}$. It is easily seen that I is the unique Herbrand model of $\mathcal{T}_1 \cup \mathcal{T}_2$.

Next we give an example in which clauses of \mathcal{T}_2 have predicates in their body which are defined in \mathcal{T}_1 , and vice versa (hence, there is a kind of mutual dependency between \mathcal{T}_1 and \mathcal{T}_2). In this example the equality (*) doesn't hold, and we have a strict inclusion.

Example 4.2

$$\begin{array}{l} \mathcal{T}_1 : \left\{ \begin{array}{l} \text{parent}(X) \leftarrow \text{father}(X) \\ \text{parent}(X) \leftarrow \text{mother}(X) \end{array} \right. \\ \mathcal{T}_2 : \left\{ \begin{array}{l} \text{father}(X) \leftarrow \text{parent}(X), \text{male}(X) \\ \text{mother}(X) \leftarrow \text{parent}(X), \text{female}(X) \\ \text{male}(a) \leftarrow \top \end{array} \right. \end{array}$$

In \mathcal{T}_1 *parent/1* is defined in terms of *father/1* and *mother/1*, while in \mathcal{T}_2 *father/1* and *mother/1* are defined in terms of *parent/1*.

The Herbrand interpretation $I = \{\text{male}(a), \text{parent}(a), \text{father}(a)\}$ is a model of both \mathcal{T}_1 and \mathcal{T}_2 , but it is not a model of $\mathcal{T}_1 \cup \mathcal{T}_2$. The only Herbrand model of $\mathcal{T}_1 \cup \mathcal{T}_2$ is $I' = \{\text{male}(a)\}$.

Example 4.2 suggests that the condition to be put on \mathcal{T}_1 and \mathcal{T}_2 to obtain the equality (*) is that only 'new' predicates (i.e. predicates not occurring in \mathcal{T}_{1d}) can be defined in \mathcal{T}_2 . This condition is surely sufficient (see theorem 6.3 in section 6), but is not necessary, as is shown by the following example.

Example 4.3

$$\begin{array}{l} \mathcal{T}_1 : \left\{ \begin{array}{l} \text{grpar}(X, Y) \leftarrow \text{parent}(X, Z), \text{parent}(Z, Y) \\ \text{gr_grpar}(X, Y) \leftarrow \text{grpar}(X, Z), \text{parent}(Z, Y) \end{array} \right. \\ \mathcal{T}_2 : \left\{ \begin{array}{l} \text{parent}(a, b) \leftarrow \top \\ \text{parent}(b, c) \leftarrow \top \\ \text{parent}(c, d) \leftarrow \top \end{array} \right. \end{array}$$

\mathcal{T}_2 defines $\text{gr_grpar}/2$ in terms of $\text{grpar}/2$ and $\text{parent}/2$, and $\text{grpar}/2$ is defined in \mathcal{T}_1 in terms of $\text{parent}/2$, which is defined in \mathcal{T}_2 .

The equality (*) holds. For instance, the only Herbrand interpretation which is a model of \mathcal{T}_1 and of \mathcal{T}_2 is $\{\text{parent}(a, b), \text{parent}(b, c), \text{parent}(c, d), \text{grpar}(a, c), \text{grpar}(b, d), \text{gr_grpar}(a, d)\}$, and this is the unique Herbrand model of $\mathcal{T}_1 \cup \mathcal{T}_2$.

Comparing the last two examples, we see that in example 4.2 the dependency between the defined predicates in \mathcal{T}_1 and the defined predicates in \mathcal{T}_2 is an 'infinite' one:

$\text{parent}/1(\mathcal{T}_1)$ is defined in terms of $\text{father}/1(\mathcal{T}_2)$,
 $\text{father}/1(\mathcal{T}_2)$ is defined in terms of $\text{parent}/1(\mathcal{T}_1)$,
 $\text{parent}/1(\mathcal{T}_1)$ is defined in terms of $\text{father}/1(\mathcal{T}_2)$, ...

Whereas in example 4.3, the dependency is a 'finite' one:

$\text{gr_grpar}/2(\mathcal{T}_2)$ is defined in terms of $\text{grpar}/2(\mathcal{T}_1)$,
 $\text{grpar}/2(\mathcal{T}_1)$ is defined in terms of $\text{parent}/2(\mathcal{T}_2)$,
 $\text{parent}/2(\mathcal{T}_2)$ is defined totally in \mathcal{T}_2 .

A condition in terms of dependency relations is discussed in the next sections. We now give an example which requires a more general condition. The equality (*) holds, although the dependency between the defined predicates in \mathcal{T}_1 and the defined predicates in \mathcal{T}_2 is an infinite one.

Example 4.4

$$\begin{array}{l} \mathcal{T}_1 : \left\{ \begin{array}{l} \text{dead} \leftarrow \neg\text{alive}, \neg\text{dancing} \\ \text{dancing} \quad (\text{FOL axiom}) \end{array} \right. \\ \mathcal{T}_2 : \left\{ \begin{array}{l} \text{alive} \leftarrow \neg\text{dead} \end{array} \right. \end{array}$$

The predicates $\text{dead}(\mathcal{T}_1)$ and $\text{alive}(\mathcal{T}_2)$ depend on each other.

But the equality $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$ holds. Indeed, the intersection of $\text{Mod}(\mathcal{T}_1)$ and $\text{Mod}(\mathcal{T}_2)$ consists of one interpretation $\{\text{dancing}^t, \text{alive}^t, \text{dead}^f\}$, which is the only model of $\mathcal{T}_1 \cup \mathcal{T}_2$.

In Event Calculus, there is a broad class of examples for which, like example 4.4, the equality (*) holds although there is an infinite kind of dependency between the defined predicates in \mathcal{T}_1 and the defined predicates in \mathcal{T}_2 . These examples require a more general condition. In the following subsection we define the justification condition, which subsumes all the following conditions given in sections 5 and 6. We will also give an example in a sort of Event Calculus to illustrate the justification condition.

4.2 The justification condition

Definition 4.1 Two theories \mathcal{T}_1 and \mathcal{T}_2 satisfy the justification condition if for each interpretation which is a model of the FOL axioms $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$ it holds that for any fact F , if there

is a justification of F in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ with only true leaves (or no leaves) and an infinite branch with an infinite number of facts defined in \mathcal{T}_1 and an infinite number of facts defined in \mathcal{T}_2 , then there is a strong justification of F in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$.

Now we are able to give the main theorem.

Theorem 4.1 *Given two correct theories \mathcal{T}_1 and \mathcal{T}_2 . If \mathcal{T}_1 and \mathcal{T}_2 satisfy the justification condition, then*

$$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2).$$

For the proof of theorem 4.1 we first need the following lemma.

Lemma 4.1 *Given two correct theories \mathcal{T}_1 and \mathcal{T}_2 and a model M of $\mathcal{T}_1 \cup \mathcal{T}_2$. Let F be a fact of a predicate defined in \mathcal{T}_1 (resp. \mathcal{T}_2) with $\mathcal{H}_M(F) = \mathbf{u}$. Then there is a weak justification of F in \mathcal{T}_1 (resp. \mathcal{T}_2) with a leaf G defined in \mathcal{T}_2 (resp. \mathcal{T}_1) and $\mathcal{H}_M(G) = \mathbf{u}$.*

Proof Suppose that, given the conditions, the conclusion of the lemma doesn't hold. Then for every justification J of F in \mathcal{T}_1 with value \mathbf{u} , the facts defined in \mathcal{T}_2 occurring as leaf of J have value \mathbf{t} . Consider the incomplete interpretation M'_0 on the facts open in \mathcal{T}_1 , with the same pre-interpretation as M and truth function $\mathcal{H}_{M'_0}$ on the open facts in \mathcal{T}_1 defined as follows: $\mathcal{H}_{M'_0} = \mathcal{H}_M$ on the facts open in $\mathcal{T}_1 \cup \mathcal{T}_2$ and on the facts G defined in \mathcal{T}_2 with $\mathcal{H}_M(G) \neq \mathbf{u}$, and $\mathcal{H}_{M'_0}(G) = \mathbf{t}$ on the other facts G defined in \mathcal{T}_2 . Hence M'_0 is 2-valued on the facts open in \mathcal{T}_1 and $\mathcal{H}_{M'_0} = \mathcal{H}_M$ on the facts defined in \mathcal{T}_2 occurring as leaf in the justifications of F in \mathcal{T}_1 with value \mathbf{u} . Then there is a unique interpretation M' extending M'_0 to all the facts such that M' is a justified model of \mathcal{T}_{1d} (theorem 2.2). By the construction of M'_0 and M' it is clear that $SV^{\mathcal{T}_1}(M', F) = SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathbf{u}$, hence $\mathcal{H}_{M'}(F) = \mathbf{u}$.

It is sufficient to prove that M' is a model of the FOL axioms \mathcal{T}_{1c} . Indeed, this gives a contradiction with the assumption that \mathcal{T}_1 has only 2-valued models. We prove that M is F -weaker than M' ($M \leq_F M'$, i.e. for each positive fact G which is true or false according to M , $\mathcal{H}_M(G) = \mathcal{H}_{M'}(G)$). Then, because M is a model of \mathcal{T}_{1c} , M' is a model of \mathcal{T}_{1c} (we can prove by induction on the structure of an axiom φ that if φ is \mathbf{t} or \mathbf{f} according to M , $\mathcal{H}_M(\varphi) = \mathcal{H}_{M'}(\varphi)$).

Given a fact G . If G is open in \mathcal{T}_1 and $\mathcal{H}_M(G)$ is \mathbf{t} or \mathbf{f} , then $\mathcal{H}_M(G) = \mathcal{H}_{M'}(G)$ by construction of M' . Suppose G is defined in \mathcal{T}_1 and $\mathcal{H}_M(G) = \mathbf{t}$. Then there is a justification J of G in $\mathcal{T}_1 \cup \mathcal{T}_2$ with all branches true under M . Let J_1 be the justification of G in \mathcal{T}_1 obtained from J by cutting off every branch after his first fact defined in \mathcal{T}_2 . Then every branch of J_1 has value \mathbf{t} under M . Because, concerning the facts open in \mathcal{T}_1 , we already have that M is F -weaker than M' , every branch of J_1 is true under M' . Hence, $SV^{\mathcal{T}_1}(M', G) = \mathbf{t}$ and $\mathcal{H}_{M'}(G) = \mathbf{t} = \mathcal{H}_M(G)$. If $\mathcal{H}_M(G) = \mathbf{f}$, then $\mathcal{H}_M(\sim G) = \mathbf{t}$ and by the previous $SV^{\mathcal{T}_1}(M', \sim G) = \mathbf{t}$. By the consistency theorem (2.1) $SV^{\mathcal{T}_1}(M', G) = \mathbf{f}$ and $\mathcal{H}_{M'}(G) = \mathbf{f} = \mathcal{H}_M(G)$. This concludes the proof of the lemma. \square

Proof (of theorem 4.1)

To prove the inclusion $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) \subseteq \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$, it is sufficient to prove, by theorem 3.1, that each model of $\mathcal{T}_1 \cup \mathcal{T}_2$ is 2-valued. Let M be a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ and assume there is a fact F (defined in \mathcal{T}_1) with $\mathcal{H}_M(F) = \mathbf{u}$. Applying the previous lemma, there is a justification J_1 of F in \mathcal{T}_1 with $\text{val}_M(J_1) = \mathbf{u}$ and there is a leaf G

of J_1 , defined in \mathcal{T}_2 , with $\mathcal{H}_M(G) = \mathbf{u}$. We can extend J_1 by adhering to each true leaf, defined in \mathcal{T}_2 , a strong justification in $\mathcal{T}_1 \cup \mathcal{T}_2$ (there exists one because M is a model of $\mathcal{T}_1 \cup \mathcal{T}_2$) and by adhering to each unknown leaf (like G) a weak justification J_2 in \mathcal{T}_2 (there exists one according to the previous lemma). We can repeat this process (on G and J_2 instead of F and J_1) and because of the lemma it will not stop. Hence, there exists an interpretation (M), model of $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$, and there is a fact (F) which has a justification with only true leaves and an infinite branch with an infinite number of facts defined in \mathcal{T}_1 and an infinite number of facts defined in \mathcal{T}_2 and F has no strong justification. This contradicts the justification condition, so $\mathcal{H}_M(F) = \mathbf{u}$ is impossible and the first inclusion is proved.

To prove the other inclusion, let M be a model of \mathcal{T}_1 and of \mathcal{T}_2 . By the monotonicity of FOL it is clear that M is a model of $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$. To prove that M is a justified model of $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$, we take a fact F and prove that $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathcal{H}_M(F)$. This is clear when F is open in $\mathcal{T}_1 \cup \mathcal{T}_2$ (note that M is 2-valued). Suppose F is defined in \mathcal{T}_1 (analogously for \mathcal{T}_2) and $\mathcal{H}_M(F) = \mathbf{t}$. Because $\mathcal{H}_M(F) = SV^{\mathcal{T}_1}(M, F)$, there exists a strong justification J_1 of F in \mathcal{T}_1 , i.e. with every branch value \mathbf{t} . Extending J_1 , we are going to build a justification J of F in $\mathcal{T}_1 \cup \mathcal{T}_2$. We do this by adhering to each leaf of J_1 which is a defined fact in \mathcal{T}_2 a strong justification in \mathcal{T}_2 (there exists one because the leaves have truth value \mathbf{t} and M is a model of \mathcal{T}_2). We repeat this process. If it stops, we obtain a strong justification of F in $\mathcal{T}_1 \cup \mathcal{T}_2$ and hence $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathcal{H}_M(F) = \mathbf{t}$. If it doesn't stop, there is an interpretation (M), model of $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$, a fact (F) and a justification of F in $\mathcal{T}_1 \cup \mathcal{T}_2$ with only true leaves and an infinite branch with an infinite number of facts defined in \mathcal{T}_1 and an infinite number of facts defined in \mathcal{T}_2 . Because of the justification condition F has a strong justification in $\mathcal{T}_1 \cup \mathcal{T}_2$. Hence, $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathcal{H}_M(F) = \mathbf{t}$. Suppose next that $\mathcal{H}_M(F) = \mathbf{f}$. Then $\mathcal{H}_M(\sim F) = \mathbf{t}$ and by the previous we know that $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, \sim F) = \mathbf{t}$. By the consistency theorem (2.1), $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathbf{f}$ and hence $SV^{\mathcal{T}_1 \cup \mathcal{T}_2}(M, F) = \mathcal{H}_M(F)$. This concludes the proof of the theorem. \square

Theorem 4.1 implies that if \mathcal{T}_1 and \mathcal{T}_2 are correct and if they satisfy the justification condition, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is also correct.

Example 4.5 *Reconsider example 3.5 of the previous section. \mathcal{T}_1 and \mathcal{T}_2 don't satisfy the justification condition. Indeed, for each interpretation with pre-interpretation J_0 the justification of a fact $\text{even}(z)$ (analogously $\text{odd}(z)$), $z \in \mathbb{Z}$, is an infinite branch with infinitely many facts defined in \mathcal{T}_1 (the even-facts) and infinitely many facts defined in \mathcal{T}_2 (the odd-facts), and there is no justification for $\text{even}(z)$ (analogously for $\text{odd}(z)$) with value \mathbf{t} .*

As was already shown in example 3.5 the only model of $\mathcal{T}_1 \cup \mathcal{T}_2$ with pre-interpretation J_0 is 3-valued and the equality () doesn't hold.*

But, as we mentioned before, the only interpretations that really matter here are the Herbrand interpretations. Looking at Herbrand interpretations only, we can see that every justification of a fact is finite. Hence, for Herbrand interpretations the justification condition is satisfied and it holds that $\text{Mod}^{\mathcal{H}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{H}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{H}}(\mathcal{T}_2)$.

These observations lead to the following definition and theorem.

Definition 4.2 *Given a class of interpretations \mathcal{J} . Two theories \mathcal{T}_1 and \mathcal{T}_2 satisfy the justification condition w.r.t. \mathcal{J} if for each interpretation of \mathcal{J} which is a model of the FOL axioms*

$\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$, it holds that for any fact F , if there is a justification of F in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ with only true leaves (or no leaves) and an infinite branch with an infinite number of facts defined in \mathcal{T}_1 and an infinite number of facts defined in \mathcal{T}_2 , then there is a strong justification of F in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$.

When \mathcal{J} is the class of all interpretations, we simply get the justification condition as stated in definition 4.1. The next theorem uses this concept to generalise theorem 4.1.

Theorem 4.2 *Given a class of interpretations \mathcal{J} and two theories \mathcal{T}_1 and \mathcal{T}_2 correct w.r.t. \mathcal{J} . If \mathcal{T}_1 and \mathcal{T}_2 satisfy the justification condition w.r.t. \mathcal{J} , then*

$$\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2).$$

Proof The proof is analogous to the proof of theorem 4.1. We now consider only interpretations of \mathcal{J} . □

Hence, if \mathcal{T}_1 and \mathcal{T}_2 correct w.r.t. \mathcal{J} and if they satisfy the justification condition w.r.t. \mathcal{J} , then $\mathcal{T}_1 \cup \mathcal{T}_2$ is correct w.r.t. \mathcal{J} . The theories \mathcal{T}_1 and \mathcal{T}_2 of example 4.5 satisfy the conditions of theorem 4.2 with $\mathcal{J} = \mathcal{H}$.

We will now give an example similar to Event Calculus.

Example 4.6

$$\begin{array}{l} \mathcal{T}_1 : \left\{ \begin{array}{l} \text{on}(T) \quad \leftarrow \quad \text{sw}(E) \quad , \quad E < T \quad , \quad \text{off}(E) \quad , \\ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \neg \text{swturn}(E, T) \\ \text{swturn}(E, T) \leftarrow \text{sw}(E_1) \quad , \quad E < E_1 \quad , \quad E_1 < T \end{array} \right. \\ \mathcal{T}_2 : \left\{ \begin{array}{l} \text{off}(T) \leftarrow \text{sw}(E) \quad , \quad E < T \quad , \quad \text{on}(E) \quad , \\ \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \neg \text{swturn}(E, T) \end{array} \right. \end{array}$$

Suppose we have a situation in which there is a lamp and a switch. The lamp can be on or off on a certain time. When the lamp is on and we turn the switch, the lamp will be off and vice versa. The lamp is on (resp. off) on a certain time T if there is an event E before T on which the lamp was off (resp. on) and on which there was a switch and such that there were no switches between E and T .

A first expert defines the predicate $\text{on}/1$ in terms of the predicate $\text{off}/1$, of which he has no knowledge, resulting in the theory \mathcal{T}_1 , and a second expert defines the predicate $\text{off}/1$ in terms of the predicate $\text{on}/1$, of which he has no knowledge, resulting in the theory \mathcal{T}_2 . We would like to know if the union of these two theories contains exactly the sum of the knowledges of the two experts. In particular, we would like to know if the equality

$$\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$$

holds. This is not the case. Consider for instance the class of interpretations \mathcal{F} with domain \mathbb{R} (the real numbers), the interpretation of the open predicate $<$ just the 'smaller than' relation on the real numbers and the interpretation of the open predicate $\text{sw}/1$:

$$\{ \text{sw}(z)^{\text{t}} \mid z \in \mathbb{Z} \} \cup \{ \text{sw}(r)^{\text{f}} \mid r \in \mathbb{R} - \mathbb{Z} \}.$$

Consider the interpretation I in \mathcal{F} with the interpretation of $\text{swturn}/2$:

$$\{ \text{swturn}(r, s)^{\text{t}} \mid r, s \in \mathbb{R} \quad , \quad r < s \quad , \quad \exists z \in \mathbb{Z} : z \in]r, s[\}$$

and all the other swturn-facts false, the interpretation of on/1:

$$\{on(r)^t \mid r \in \mathbb{R}, \exists z \in \mathbb{Z} : 2z < r \leq 2z + 1\}$$

and all the other on-facts false and the interpretations of off/1:

$$\{off(r)^t \mid r \in \mathbb{R}, \exists z \in \mathbb{Z} : 2z - 1 < r \leq 2z\}$$

and all the other off-facts false. Then I is a model of \mathcal{T}_1 and of \mathcal{T}_2 . But I is not a model of the union $\mathcal{T}_1 \cup \mathcal{T}_2$. Indeed, the only interpretation in \mathcal{F} which is a model of $\mathcal{T}_1 \cup \mathcal{T}_2$ assigns to all on-facts and all off-facts truth value **f** (every justification in $\mathcal{T}_1 \cup \mathcal{T}_2$ of these facts has a positive loop consisting of on- and off-facts).

Now suppose that both experts have the following a priori knowledge, represented by the class \mathcal{J} of all interpretations in which this a priori knowledge is true. The experts both know that the domain is \mathbb{R}^+ (the positive real numbers), the interpretation of the open predicate $</2$ is the 'smaller than' relation on the positive real numbers, they know that $off(0)$ is true (representing the fact that initially the light is off) and they both have complete knowledge about the open predicate $sw/1$:

$$\{sw(n)^t \mid n \in D\} \cup \{sw(r)^f \mid r \in \mathbb{R}^+ - D\}$$

where D is a fixed subset of \mathbb{N} , modelling the fact that at each natural number of D there is a switch. D can be for instance \mathbb{N} or a finite subset of the natural numbers.

Then $\mathcal{T}_1, \mathcal{T}_2$ satisfy the justification condition w.r.t. \mathcal{J} . Indeed, take an arbitrary interpretation in \mathcal{J} . It is obvious that all justifications of the swturn-facts in $\mathcal{T}_1 \cup \mathcal{T}_2$ are finite. We now take a closer look at the justifications in $\mathcal{T}_1 \cup \mathcal{T}_2$ of the on-facts (analogously for the off-facts). The justifications of an on-fact, $on(t)$ ($t \in \mathbb{R}^+$), with only true leaves are always finite. This is because, starting from a given $t \in \mathbb{R}^+$, each elementary justification of $on(t)$ is of the form $\{sw(e), e < t, off(e), \neg swturn(e, t)\}$. The fact $sw(e)$ is true only if $e \in D$. The fact $e < t$ is true only if e is taken smaller than t . Hence, in a justification of $on(t)$ with only true leaves, $e < t$ and $e \in D$. We can then repeat the same reasoning for the fact $off(e)$. Because the domain of all the interpretations in \mathcal{J} is \mathbb{R}^+ , each justification of $on(t)$ with only true leaves is finite.

By applying theorem 4.2 (note that \mathcal{T}_1 and \mathcal{T}_2 are correct w.r.t. \mathcal{J}) we get

$$Mod^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = Mod^{\mathcal{J}}(\mathcal{T}_1) \cap Mod^{\mathcal{J}}(\mathcal{T}_2).$$

Intersecting $Mod^{\mathcal{J}}(\mathcal{T}_1)$ and $Mod^{\mathcal{J}}(\mathcal{T}_2)$, we obtain only one interpretation. Let $D = \{n_1, n_2, \dots\} \subseteq \mathbb{N}$ and let $n_0 = 0$, then the unique interpretation of $Mod^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2)$ is given as follows: the true on-facts are given by

$$\{on(r) \mid n_{i+1} < r \leq n_{i+2} \text{ and } i \in \mathbb{N}\}$$

(all the other on-facts are false) and the true off-facts are given by

$$\{off(r) \mid n_i < r \leq n_{i+1} \text{ and } i \in \mathbb{N}\} \cup \{off(0)\}$$

(and all the other off-facts are false).

The justification condition is a very general condition from which many other conditions can be deduced, like for instance the next proposition, which is an easy corollary to theorem 4.2.

Proposition 4.1 *Given a class of interpretations \mathcal{J} and two theories \mathcal{T}_1 and \mathcal{T}_2 correct w.r.t. \mathcal{J} . If for each interpretation in \mathcal{J} , which is a model of the FOL axioms $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$, it holds that for every fact F , F or $\sim F$ has a strong justification of finite depth in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$, then*

$$\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2).$$

Proof Suppose that for each interpretation in \mathcal{J} and for every fact F it holds that F or $\sim F$ has a strong justification of finite depth in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$. We prove that $\mathcal{T}_{1d}, \mathcal{T}_{2d}$ satisfy the justification condition. Then the proposition is a direct consequence of theorem 4.2. Suppose there is an interpretation I in \mathcal{J} and a fact F which has a justification J_1 in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ with only true leaves (or no leaves) and with an infinite branch with an infinite number of facts defined in \mathcal{T}_{1d} and an infinite number of facts defined in \mathcal{T}_{2d} and suppose F has no strong justification in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$. We prove that this is impossible. Because F has no strong justification in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$, $\sim F$ has a strong justification J_2 of finite depth in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$. By the definition of justification, there is a branch B_1 in J_1 and a branch B_2 in J_2 such that $B_1 = \sim B_2$. But this means that B_1 has a false leaf, which gives a contradiction. \square

In the next sections we introduce some stronger, but more syntactical conditions on the theories \mathcal{T}_1 and \mathcal{T}_2 such that (*) still holds.

5 A more syntactical condition for propositional theories

In this section we restrict ourselves to propositional theories, based on a propositional language \mathcal{L} . In the next section we extend to the predicate case again.

Given a propositional language \mathcal{L} and a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ based on \mathcal{L} , let \succ denote the dependency relation on the proposition symbols of \mathcal{T} . Recall that the dependency relation is the transitive closure of the relation \succ_1 , with $p \succ_1 q$ if there's a clause in \mathcal{T}_d with head p , and q or $\neg q$ in the body. If $p \succ q$, we say that p depends on q .

We now introduce a condition on the propositional theories \mathcal{T}_1 and \mathcal{T}_2 in terms of this dependency relation. The next theorem states that if \mathcal{T}_1 and \mathcal{T}_2 satisfy this new condition, the equality (*) holds.

Theorem 5.1 *Given a propositional language \mathcal{L} and two correct theories \mathcal{T}_1 and \mathcal{T}_2 based on \mathcal{L} . Consider the dependency relation \succ on the proposition symbols of $\mathcal{T}_1 \cup \mathcal{T}_2$. If for each descending sequence K of proposition symbols, there is an $i \in \{1, 2\}$ such that the proposition symbols defined in \mathcal{T}_i appear only finitely many times in the sequence K , then $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$.*

Proof It is clear by the definition of a justification (definition 2.3) that if \mathcal{T}_1 and \mathcal{T}_2 satisfy the condition of the theorem, there are no justifications in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ with an infinite branch with an infinite number of facts defined in \mathcal{T}_1 and an infinite number of facts defined in \mathcal{T}_2 . Hence, by the definition of justification condition (definition 4.1), \mathcal{T}_1 and \mathcal{T}_2 satisfy the justification condition. The theorem follows then from theorem 4.1. \square

The condition given in theorem 5.1 is sufficient, but not necessary, as was shown in example 4.4.

6 A more syntactical condition for predicate theories

We return to the case of a first order language \mathcal{L} and two theories \mathcal{T}_1 and \mathcal{T}_2 based on \mathcal{L} .

6.1 Condition in terms of dependency relation

When we only want to consider Herbrand interpretations, we can use the result of the previous section. Let $\text{Ground}(\mathcal{T}_1 \cup \mathcal{T}_2)$ denote the grounding of $\mathcal{T}_1 \cup \mathcal{T}_2$ (i.e. of the logic program part and of the FOL axioms) w.r.t. the Herbrand universe. We assume that for each ground atom p of a defined predicate symbol such that no ground instantiation of a clause has p in the head, the grounding contains the rule $p \leftarrow \perp$.

Theorem 6.1 *Given two theories \mathcal{T}_1 and \mathcal{T}_2 which are correct w.r.t. \mathcal{H} . Consider the dependency relation \succ on the ground atoms of $\text{Ground}(\mathcal{T}_1 \cup \mathcal{T}_2)$. If for each descending sequence K of ground atoms, there is an $i \in \{1, 2\}$ such that the ground atoms defined in \mathcal{T}_i appear only finitely many times in the sequence K , then $\text{Mod}^{\mathcal{H}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{H}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{H}}(\mathcal{T}_2)$.*

Proof The grounding of a theory can be treated as a propositional program. This theorem is then a direct consequence of theorem 5.1. \square

If \mathcal{T}_1 and \mathcal{T}_2 are correct w.r.t. \mathcal{H} and satisfy the condition of theorem 6.1, then $\mathcal{T}_1 \cup \mathcal{T}_2$ is also correct w.r.t. \mathcal{H} .

Sometimes it is not enough to consider only Herbrand interpretations. For this reason, let's define in the usual way the dependency relation \succ on the predicate symbols of $\mathcal{T}_1 \cup \mathcal{T}_2$. Analogously as in the previous section (see theorem 5.1), we obtain the following result.

Theorem 6.2 *Given a class of interpretations \mathcal{J} and two theories \mathcal{T}_1 and \mathcal{T}_2 correct w.r.t. \mathcal{J} . Consider the dependency relation on the predicate symbols of $\mathcal{T}_1 \cup \mathcal{T}_2$. If for each descending sequence K of predicate symbols there is an $i \in \{1, 2\}$ such that the predicate symbols defined in \mathcal{T}_i appear only finitely many times in the sequence K , then $\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2)$.*

Hence $\mathcal{T}_1 \cup \mathcal{T}_2$ is also correct w.r.t. \mathcal{J} . Examples 4.1 and 4.3 of section 4 satisfy the conditions of theorem 6.2, with \mathcal{J} the class of all interpretations, whereas example 4.2 does not. We give another nice example.

Example 6.1

$$\begin{aligned} \mathcal{T}_1 & : \begin{cases} \text{even}(0) \leftarrow \top \\ \text{even}(s(X)) \leftarrow \text{odd}(X) \end{cases} \\ \mathcal{T}_2 & : \begin{cases} \text{odd}(s(X)) \leftarrow \text{even}(X) \end{cases} \end{aligned}$$

First note that the equality $\text{Mod}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$ does not hold. Indeed, consider the 2-valued non-Herbrand interpretation I with pre-interpretation J_0 (see example 3.5), the interpretation of $\text{even}/1$

$$\{\text{even}(2 \times n)^{\mathbf{t}}, \text{even}(2 \times n + 1)^{\mathbf{f}} \mid n \in \mathbb{N}\} \cup \{\text{even}(z)^{\mathbf{t}} \mid z \in \mathbb{Z}\},$$

and the interpretation of $\text{odd}/1$

$$\{\text{odd}(2 \times n)^{\mathbf{f}}, \text{odd}(2 \times n + 1)^{\mathbf{t}} \mid n \in \mathbb{N}\} \cup \{\text{odd}(z)^{\mathbf{t}} \mid z \in \mathbb{Z}\}.$$

Then I is a non-Herbrand model of \mathcal{T}_1 and of \mathcal{T}_2 . But I is not a model of $\mathcal{T}_1 \cup \mathcal{T}_2$, because the only model of $\mathcal{T}_1 \cup \mathcal{T}_2$ with pre-interpretation J_0 assigns truth value **f** to all facts $\text{even}(z), \text{odd}(z)$ with $z \in \mathbb{Z}$.

But again, the only interpretations which are important w.r.t. these theories are the Herbrand interpretations. And for these interpretations the equality holds, i.e.

$$\text{Mod}^{\mathcal{H}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{H}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{H}}(\mathcal{T}_2).$$

To prove this, we take the grounding of $\mathcal{T}_1 \cup \mathcal{T}_2$:

$$\begin{array}{ll} \text{even}(0) & \leftarrow \top \\ \text{even}(s(0)) & \leftarrow \text{odd}(0) \\ \dots & \\ \text{even}(s^{n+1}(0)) & \leftarrow \text{odd}(s^n(0)) \\ \dots & \\ \text{odd}(0) & \leftarrow \perp \\ \text{odd}(s(0)) & \leftarrow \text{even}(0) \\ \dots & \\ \text{odd}(s^{n+1}(0)) & \leftarrow \text{even}(s^n(0)) \\ \dots & \end{array}$$

Then it is easy to check that \mathcal{T}_1 and \mathcal{T}_2 satisfy the conditions of theorem 6.1 (every descending sequence of ground atoms of $\mathcal{T}_1 \cup \mathcal{T}_2$ is finite). It is also easy to see that $\mathcal{T}_1 \cup \mathcal{T}_2$ has a unique 2-valued Herbrand model given by

$$\left\{ \text{even}(s^{2n}(0)), \text{odd}(s^{2n+1}(0)) \mid n \in \mathbb{N} \right\}.$$

Note that we can not apply theorem 6.2 (with $\mathcal{J} = \mathcal{H}$) to obtain the same result, because \mathcal{T}_1 and \mathcal{T}_2 don't satisfy the conditions of that theorem: there is an infinite descending sequence

$$\text{even}/1 \succ \text{odd}/1 \succ \text{even}/1 \succ \dots,$$

with $\text{even}/1$ defined in \mathcal{T}_1 and $\text{odd}/1$ in \mathcal{T}_2 .

In the last two subsections of this section we give some even stronger, syntactical conditions on \mathcal{T}_1 and \mathcal{T}_2 to obtain (*).

6.2 Conservative extensions

We first need some notation. Given a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$. Let $\text{Head}(\mathcal{T}_d)$ denote the set of all predicate symbols occurring in the head of a clause of \mathcal{T}_d . Let $\text{Pred}(\mathcal{T}_d)$ denote the set of all predicate symbols occurring in \mathcal{T}_d . The following theorem is a direct consequence of theorem 6.2.

Theorem 6.3 *Given a class of interpretations \mathcal{J} and two theories \mathcal{T}_1 and \mathcal{T}_2 correct w.r.t. \mathcal{J} . If $\text{Head}(\mathcal{T}_{2d}) \cap \text{Pred}(\mathcal{T}_{1d}) = \emptyset$, then $\text{Mod}^{\mathcal{J}}(\mathcal{T}_1 \cup \mathcal{T}_2) = \text{Mod}^{\mathcal{J}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{J}}(\mathcal{T}_2)$.*

The condition $\text{Head}(\mathcal{T}_{2d}) \cap \text{Pred}(\mathcal{T}_{1d}) = \emptyset$ means that only predicate symbols not occurring in \mathcal{T}_{1d} can be defined in \mathcal{T}_{2d} . Hence, predicate symbols defined in \mathcal{T}_{1d} can not depend on predicate symbols defined in \mathcal{T}_{2d} (the converse is possible though). An example was given in section 4, example 4.1.

Under the conditions of theorem 6.3, we are given a way to construct every model of $\mathcal{T}_1 \cup \mathcal{T}_2$ by successively finding a model of \mathcal{T}_1 and \mathcal{T}_2 . Let us be more precise. Suppose two correct logic programs \mathcal{T}_{1d} and \mathcal{T}_{2d} are given and \mathcal{T}_{1d} does not refer to the predicate symbols defined in \mathcal{T}_{2d} . Suppose $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ is complete. Because \mathcal{T}_{1d} does not refer to predicates defined in \mathcal{T}_{2d} , every predicate occurring in \mathcal{T}_{1d} is also defined in \mathcal{T}_{1d} . Given a pre-interpretation J , denote the set of models of \mathcal{T}_{1d} with pre-interpretation J by \mathcal{M}_J . It is clear that each model in \mathcal{M}_J has the same truth function on predicate symbols defined in \mathcal{T}_{1d} . In the set \mathcal{M}_J there is exactly one interpretation M which is also a model of \mathcal{T}_{2d} . Because of the equality, $\text{Mod}(\mathcal{T}_{1d} \cup \mathcal{T}_{2d}) = \text{Mod}(\mathcal{T}_{1d}) \cap \text{Mod}(\mathcal{T}_{2d})$, M is a model of $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ and every model of $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ can be obtained in this way. If $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ is incomplete, then, instead of starting with a given pre-interpretation only, we also fix a 2-valued truth function on the predicate symbols open in $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ and then repeat the same reasoning. The reasoning remains valid when adding FOL axioms \mathcal{T}_{1c} and \mathcal{T}_{2c} to the logic programs. This is because of the monotonicity of FOL; not satisfying \mathcal{T}_{1c} or \mathcal{T}_{2c} is equivalent with not satisfying $\mathcal{T}_{1c} \cup \mathcal{T}_{2c}$.

In a way, the logic program $\mathcal{T}_d = \mathcal{T}_{1d} \cup \mathcal{T}_{2d}$ is split into two parts. In [16] Lifschitz and Turner discuss this idea of splitting a logic program in the context of answer set semantics for disjunctive logic programs with classical negation. They call \mathcal{T}_{1d} the bottom of \mathcal{T}_d and \mathcal{T}_{2d} the top of \mathcal{T}_d .

A corollary to theorem 6.3 is a property which is in literature often called the *conservative extension property*. If we extend an initial correct logic program \mathcal{T}_{1d} by a correct logic program \mathcal{T}_{2d} , which gives only definitions for 'new' predicate symbols (i.e. not occurring in \mathcal{T}_{1d}), then for every formula φ , consisting only of predicate symbols defined in \mathcal{T}_{1d} , φ is a consequence of \mathcal{T}_{1d} if and only if φ is a consequence of $\mathcal{T}_{1d} \cup \mathcal{T}_{2d}$. This can be proven by induction on the length of the formula φ , using theorem 6.3.

Conservative extensions were studied by Lifschitz and Turner in the context of disjunctive logic programming with classical negation [16], by Gelfond and Przymusinska in the context of extended logic programming [13] and in the context of epistemic specifications [14].

6.3 Hierarchical and acyclic programs

The last results concern hierarchical and acyclic programs. For more details about this kind of programs, see [17] and [1]. We just give their definitions.

Definition 6.1 *A logic program \mathcal{T}_d is hierarchical, if there exists a mapping $||$ from $\text{Pred}(\mathcal{T}_d)$ to the natural numbers such that for every clause $p(t_1, \dots, t_n) \leftarrow L_1, \dots, L_m$ in \mathcal{T}_d , $|p|$ is greater than the value under $||$ of each predicate symbol occurring (positively or negatively) in the body.*

It is obvious that hierarchical programs cannot have 3-valued models and that each subset of clauses of a hierarchical program is itself hierarchical.

Proposition 6.1 *Given a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ with \mathcal{T}_d a hierarchical logic program. Then \mathcal{T} is correct.*

Proposition 6.2 *Given a hierarchical logic program \mathcal{T}_d . Then every $\mathcal{T}_d' \subseteq \mathcal{T}_d$ is hierarchical.*

Combining these propositions together with theorem 6.2, with \mathcal{J} the class of all interpretations, gives us:

Theorem 6.4 *Given a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ with \mathcal{T}_d a hierarchical logic program. Then for every $\mathcal{T}_1 = (\mathcal{T}_{1d}, \mathcal{T}_{1c})$ and $\mathcal{T}_2 = (\mathcal{T}_{2d}, \mathcal{T}_{2c})$ such that $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ (and such that every*

predicate is defined in at most one theory, either \mathcal{T}_1 or \mathcal{T}_2), the equality $\text{Mod}(\mathcal{T}) = \text{Mod}(\mathcal{T}_1) \cap \text{Mod}(\mathcal{T}_2)$ holds.

Using induction, we can extend this theorem to split a theory into a finite number of theories. We refer to the very similar case of acyclic programs below for an example of how this is done. When only Herbrand interpretations are relevant w.r.t. the theory, we can put a more general, syntactical restriction on the theory.

Definition 6.2 A logic program \mathcal{T}_d is acyclic, if there exists a mapping $||$ from the Herbrand base to the natural numbers such that for every clause $A \leftarrow B_1, \dots, B_m, \neg B_{m+1}, \dots, \neg B_n$ in $\text{Ground}(\mathcal{T}_d)$, $|A| > |B_i|$ for every $1 \leq i \leq n$.

Analogously to the case of hierarchical programs, the following properties hold.

Proposition 6.3 Given a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ with \mathcal{T}_d an acyclic logic program. Then \mathcal{T} is correct w.r.t. \mathcal{H} .

Proposition 6.4 Given an acyclic logic program \mathcal{T}_d . Then every $\mathcal{T}_d' \subseteq \mathcal{T}_d$ is acyclic.

And as a result of these two propositions and theorem 6.1 we get:

Theorem 6.5 Given a theory $\mathcal{T} = (\mathcal{T}_d, \mathcal{T}_c)$ with \mathcal{T}_d an acyclic logic program. Then for every $\mathcal{T}_1 = (\mathcal{T}_{1d}, \mathcal{T}_{1c})$ and $\mathcal{T}_2 = (\mathcal{T}_{2d}, \mathcal{T}_{2c})$ such that $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ (and such that every predicate is defined in at most one theory, either \mathcal{T}_1 or \mathcal{T}_2), the equality $\text{Mod}^{\mathcal{H}}(\mathcal{T}) = \text{Mod}^{\mathcal{H}}(\mathcal{T}_1) \cap \text{Mod}^{\mathcal{H}}(\mathcal{T}_2)$ holds.

Again, we can extend this theorem to split \mathcal{T} into a finite number of theories.

Example 6.2

$$\begin{aligned} \mathcal{T}_1 & : \begin{cases} \text{even}(0) \leftarrow \top \\ \text{even}(s(X)) \leftarrow \text{odd}(X) \end{cases} \\ \mathcal{T}_2 & : \begin{cases} \text{odd}(s(X)) \leftarrow \text{even}(X) \end{cases} \\ \mathcal{T}_3 & : \begin{cases} \text{nat}(X) \leftarrow \text{even}(X) \\ \text{nat}(X) \leftarrow \text{odd}(X) \end{cases} \end{aligned}$$

It is clear that $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3$ is acyclic. Indeed, consider the mapping $||$,

$$\begin{aligned} |\text{even}(s^n(0))| & = n \quad , \\ |\text{odd}(s^n(0))| & = n \quad , \\ |\text{nat}(s^n(0))| & = n + 1 \quad , \end{aligned}$$

with $n \in \mathbb{N}$ and $s^0(0) = 0$. Then for every clause in $\text{Ground}(\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3)$ the value under $||$ of the head is greater than the value of the atom in the body.

The intersection of $\text{Mod}^{\mathcal{H}}(\mathcal{T}_1)$, $\text{Mod}^{\mathcal{H}}(\mathcal{T}_2)$ and $\text{Mod}^{\mathcal{H}}(\mathcal{T}_3)$ gives us the unique Herbrand model of $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3$:

$$\left\{ \text{even}(s^{2n}(0)), \text{odd}(s^{2n+1}(0)), \text{nat}(s^n(0)) \mid n \in \mathbb{N} \right\}.$$

7 Related works

Prior to composing the logical modules of cooperating K.R. experts, the individual experts face the problem of representing their *incomplete* knowledge as logical modules. Therefore, we believe that the suitability of a logic for modular knowledge representation depends in the first place on its expressivity for representing *incomplete* knowledge. One point on which this study differs from most others is that it investigates the compositionality problem in the context of a knowledge representation logic, OLP-FOL, designed as an extension of logic programming for nonmonotonic knowledge representation in the context of uncertainty [10]. Most related studies modify the logic programming semantics to make it suitable for modular programming but do not investigate the knowledge representation qualities of the resulting logic. An exception is [16], which studies the compositionality problem in the context of Disjunctive Logic Programming.

In some studies in modular logic programming e.g. [7], [3], compositionality of the logic is seen as a first rank requirement; the semantics of the (syntactical) composition of modules is required to be the result of a semantical composition operator on the semantics of the modules. The semantic composition operator under study here is the *logical conjunction* operator. In a logic with a possible state semantics (as for OLP-FOL), this composition operator can be naturally defined by the intersection on the classes of models. As was observed first in [20], compositionality (w.r.t. logical conjunction) and nonmonotonicity are contradictory requirements. More precisely, when a syntactic composition operator is compositional w.r.t. the logical conjunction operator, this syntactic composition operator is monotonic. However, a monotonic logic is not very suitable for knowledge representation. The syntactic composition operator in OLP-FOL is not always compositional and it should not be.

As opposed to most of the works about compositionality, this paper considers logic programs which may contain negation in the body of their clauses. In section 6.2 we mentioned three works [16], [13] and [14] which also allow negation. Other exceptions which also consider negation are [12], [20], [8]. However, either they consider a weaker compositional semantics based on completion semantics [20], or the results are restricted either to hierarchical dependencies between modules, as in [12], or to the case of one module representing a conservative extension of another module [8]. We discuss these three works in more detail. In [12], they consider programs with import predicates (called units); import predicates correspond to open predicates in our approach. An abstract semantics for units is a function taking in input a set of imported literals and producing another set of literals. They present conservative extensions of the well-founded and Fitting's semantics. These semantics are proved to be compositional w.r.t. union of units, i.e. the semantics of the union can be deduced from the semantics of the components. In case of the extension of the well-founded semantics this is only stated if the system of units is hierarchical, that is to say that there is no circular dependency between the units. Although they use a very different terminology and their set-up is different, their approach leans closest to ours. In [20] a compositional semantics (which can be seen as a compositional counterpart of Kunen's semantics) for normal programs based on a first order completion of the program is defined. Definitions which are not explicitly given in the program are not closed and hence those predicates remain open, which is needed in a modular context. Their semantics is compositional while remaining nonmonotonic to a certain extent. In essence, the semantics is compositional and monotonic on the level of composition of modules, while addition of clauses to modules remains a nonmonotonic operation. In [8] a compositional semantics for logic programs is defined which handles inconsistencies locally instead of globally. This semantics fulfills the following composition requirement stating that the meaning of a

program P is not modified if P is extended with a program Q such that none of the atoms defined in P are also defined in Q and no definitions in P depend on atoms defined in Q (the program Q is in fact a conservative extension of the program P).

By allowing predicates to be open in a logic program, we deal with incomplete knowledge on predicates. In [12] import predicates are introduced to represent this kind of incomplete knowledge. In [7], [6] they work with a different notion of open predicates to capture the possible composition with other programs. Their semantics of admissible Herbrand models for definite programs is suited for compositionality problems, but seems less suitable for knowledge representation. In [5] this semantics is extended to normal logic programs by transforming them to open positive programs. Though in this semantics the compositionality problem can be handled elegantly, the semantics is much weaker than ours and seems not suited for knowledge representation and nonmonotonic reasoning.

A special place in the compositionality research is taken by [3], where each definite program is denoted by its immediate consequence operator T_P and not by its set of models. The union of two definite programs can be proven to correspond to a certain operation on the corresponding T_P operators. The result is a highly abstract sort of semantics of a program, which is not really suitable for studying knowledge representation problems, but which allows [3] to define many different composition operators, all in terms of different ways of composing the T_P operators of the distinct modules. In comparison, we investigate only one operator, namely the operator which joins the knowledge of the modules. Recently, the algebraic approach of [3] was extended for normal logic programs [4] using Fittings 3-valued completion operator.

In [2] the problem of modelling the composition by union of definite programs is studied, by considering computed answer substitution as observable behaviour of programs (instead of the more standard notion of success set). To capture this notion of behaviour, programs are denoted by programs (obtained through unfolding) rather than by Herbrand models. The OR-compositional (i.e. compositional w.r.t. program union) semantics of open programs they define, corresponds to a program equivalence notion, according to which two programs P_1 and P_2 are equivalent iff for any program Q , $P_1 \cup Q$ and $P_2 \cup Q$ give the same computed answer substitutions.

Besides uncertainty on predicates, one can also have incomplete knowledge on the domain of discourse. By considering general interpretations, like in [20], we take into account this kind of incomplete knowledge. Other approaches either do not allow to model this kind of incomplete knowledge or model it by allowing Herbrand interpretations of arbitrary extensions of the module language.

In the previous discussed approaches and also in our approach, module composition is seen as a metalinguistic mechanism. Another main direction in the research of compositionality of logic programming formalisms is of a linguistic nature and is seen for instance in [18], [15]. They extend the formalism of Horn clause logic with modal operators in order to provide a richer support for modular programming. In [15], they show that (multi)modal logics are well-suited for supporting the notion of module. Each module is given a name and for each module a modal operator is introduced. They give a modal characterisation of the situation in which the meaning of modules does not depend on the external environment (and on other modules) except for explicit importations. However, they give some hints on how this modal characterisation can also be used to capture the notions of inheritance between modules, of operations on modules such as union or how it can be used to define programs with several modules each one with its own internal language. All this offers lots of possibilities, which are worth investigating further.

For a survey of different kinds of approaches to modularity, we refer to [9].

Note that we did not consider problems such as: parametrised modules, several experts designing definitions for the same concepts, several experts overloading the same predicate symbol to represent different concepts, These are topics for future work.

References

- [1] K.R. Apt and M. Bezem. Acyclic programs. In *Proc. of the International Conference on Logic Programming*, pages 579–597. MIT press, 1990.
- [2] A. Bossi, M. Gabbrielli, G. Levi, and M.C. Meo. Contributions to the Semantics of Open Logic Programs. In ICOT, editor, *Proc. of the International Conference on Fifth Generating Computer Systems*, 1992.
- [3] A. Brogi. *Program Construction in Computational Logic*. PhD thesis, Dipartimento di Informatica Universita di Pisa, 1993.
- [4] A. Brogi, S. Contiero, and F. Turini. Composing General Logic Programs. *to appear in Proc. of LPNMR*, 1997.
- [5] A. Brogi, E. Lamma, P. Mancarella, and P. Mello. Normal Logic Programs as Open Positive Programs. In K. Apt, editor, *Proc. of JICSLP*, pages 783–797, 1992.
- [6] A. Brogi, E. Lamma, and P. Mello. Open Logic Theories. In L.-H. Eriksson, L. Hallnas, and P. Schroeder-Heister, editors, *Proc. of the Second International Workshop on Extensions of Logic Programming*, pages 73–88. Springer-Verlag, 1991.
- [7] A. Brogi, E. Lamma, and P. Mello. Compositional Model-Theoretic Semantics for Logic Programs. *New Generation Computing*, 11(1):1–21, 1992.
- [8] F. Bry. A Compositional Semantics for Logic Programs and Deductive Databases. In M. Maher, editor, *Proc. of JICSLP*, pages 453–467, 1996.
- [9] M. Bugliesi, E. Lamma, and P. Mello. Modularity in logic programming. *Journal of Logic Programming*, 20:443–502, 1994.
- [10] M. Denecker. A Terminological Interpretation of (Abductive) Logic Programming. In V.W. Marek, A. Nerode, and M. Truszczynski, editors, *International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 15–29. Springer, Lecture notes in Artificial Intelligence 928, 1995.
- [11] M. Denecker and D. De Schreye. Justification semantics: a unifying framework for the semantics of logic programs. In *Proc. of the Logic Programming and Nonmonotonic Reasoning Workshop*, pages 365–379, 1993.
- [12] A. Lallouet G. Ferrand. A Compositional Proof Method of Partial Correctness for Normal Logic Programs. In J. Lloyd, editor, *Proc. of ISLP*, pages 210–223, 1995.
- [13] M. Gelfond and H. Przymusinska. Towards a Theory of Elaboration Tolerance: Logic Programming Approach. *to appear in Journal on Software and Knowledge Engineering*.
- [14] M. Gelfond and H. Przymusinska. Definitions in Epistemic Specifications. In A. Nerode, W. Marek, and V. S. Subrahmanian, editors, *Proc. of the First International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 245–259, 1991.

- [15] L. Giordano and A. Martelli. A modal reconstruction of blocks and modules in logic programming. In *Proc. of the International Logic Programming Symposium*, pages 239–253. The MIT Press, 1991.
- [16] V. Lifschitz and H. Turner. Splitting a Logic Program. In P. Van Hentenrijck, editor, *Proc. of the Eleventh International Conference on Logic Programming*, pages 23–38, 1994.
- [17] J.W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, 1987.
- [18] L. Monteiro and A. Porto. Contextual Logic Programming. In G. Levi and M. Martelli, editors, *Proc. of the Sixth International Conference on Logic Programming*, 1989.
- [19] L. M. Pereira, J.N. Aparicio, and J.J. Alferes. Hypothetical reasoning with well founded semantics. Technical report, AI Centre, Uninova, Portugal, 1990.
- [20] F. Teusink S. Etalle. A Compositional Semantics for Normal Open Programs. In M. Maher, editor, *Proc. of JICSLP*, pages 468–482, 1996.
- [21] A. Van 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.