From Logic Programming Semantics to the Consistency of Syntactical Treatments of Knowledge and Belief

Thomas Bolander
Informatics and Mathematical Modelling
Technical University of Denmark
tb@imm.dtu.dk

Abstract

This paper concerns formal theories for reasoning about the knowledge and belief of agents. It has seemed attractive to researchers in artificial intelligence to formalise these propositional attitudes as predicates of first-order predicate logic. This allows the agents to express stronger introspective beliefs and engage in stronger meta-reasoning than in the classical modal operator approach. Results by Montague [1963] and Thomason [1980] show, however, that the predicate approach is prone to inconsistency. More recent results by des Rivières & Levesque [1988] and Morreau & Kraus [1998] show that we can maintain the predicate approach if we make suitable restrictions to our set of epistemic axioms. Their results are proved by careful translations from corresponding modal formalisms. In the present paper we show that their results fit nicely into the framework of logic programming semantics, in that we show their results to be corollaries of well-known results in this field. This does not only allow us to demonstrate a close connection between consistency problems in the syntactic treatment of propositional attitudes and problems in semantics for logic programs, but it also allows us to strengthen the results of des Rivières & Levesque [1988] and Morreau & Kraus [1998].

1 Introduction

The approach most often used in constructing formal theories for reasoning about multiagent systems is to formalise the agents' beliefs and knowledge through modal operators. An alternative approach is to formalise these propositional attitudes as predicates of a first-order predicate logic. This has several advantages, which have been widely discussed in the literature [Davies, 1990; Attardi and Simi, 1995; Carlucci Aiello *et al.*, 1995; McCarthy, 1997; Grant *et al.*, 2000]. Most importantly, it allows us to quantify over the propositional objects of knowledge and belief as for instance in "agent 1 believes that *everything* known by agent 2 is also known by agent 1", formalised by

$$B_1(\ulcorner\forall x(K_2(x) \rightarrow K_1(x))\urcorner).$$

This formula has no counterpart in the classical modal operator approach, since if K_1 and K_2 were modal operators, we would not be able to apply them directly to the variable x (modal operators only apply to well-formed formulas). Thus the predicate approach gives us more expressive power and the ability of agents to refer to the totality of their own and others beliefs, which is important in meta-reasoning.

Unfortunately, the predicate approach easily becomes inconsistent, since the added expressive power allows the agents to express self-referential beliefs that in some cases turn out to be paradoxical. This was proved by Montague [1963] and Thomason [1980]. They prove that certain axiom schemes describing natural properties of knowledge and belief are inconsistent with formal arithmetic. Their results are reviewed in Section 3. Des Rivières & Levesque [1988] and Morreau & Kraus [1998] have shown there to be a way out of these inconsistency results: to suitably restrict the set of sentences that we instantiate our axiom schemes of knowledge and belief with. These results are reviewed in Section 3 as well.

In this paper we will show that the results of des Rivières, Levesque, Morreau and Kraus can be reduced to well-known results in logic programming semantics. This is carried out in Section 4. In Section 5 we give a strengthening of their results, using again the connection to logic programming semantics.

2 Terminology and Notation

We will be using theories of first-order predicate logic to formalise propositional attitudes of agents. To prove the consistency of these first-order theories, we use results from logic programming semantics. Below we introduce the kinds of logic programs and first-order languages we will be considering.

2.1 Logic Programs

All logic programs considered in this paper will be propositional. Thus, an **atom** is simply a propositional letter, and a **literal** is either a propositional letter p or its negation $\neg p$. We take the symbols true and false to be among our propositional letters with the obvious intended interpretation. A **clause** is a formula of propositional logic on the form

$$H \leftarrow L_1 \wedge L_2 \wedge \cdots \wedge L_n$$

where H is an atom, $n \geq 1$ and all L_i are literals. A **propositional program** (or simply a **program**) is a (possibly infinite) set of clauses. **Herbrand models** of programs are defined in the usual way. We require that all models assign the truth-value true to the propositional letter $true$ and false to $false$.

Given a program P, $\mathrm{comp}(P)$ denotes its Clark completion. Since we work only with propositional programs, the Clark completion is particularly simple. The **Clark completion** of P is the following set of equivalences in infinitary propositional logic: for each atom A in P,

- if A does not appear as head of any clause in P, then $A \leftrightarrow \mathit{false} \in \mathrm{comp}(P)$.
- otherwise we have $A \leftrightarrow \bigvee_{i \in I} B_i \in \mathrm{comp}(P)$, where $\{A \leftarrow B_i \mid i \in I\}$ is the set of clauses in P with head A.

Let P be a propositional program. The **dependency graph** of P is the directed graph with signed edges defined as follows. The nodes of the graph are the atoms (propositional letters) occurring in P excluding the special atoms $true$ and $false$. There is a positive edge from A to B, denoted by $\langle A, B, + \rangle$, if and only if there is a clause $A \leftarrow L_1 \wedge \cdots \wedge L_n$ in P such that $L_i = B$ for some $1 \leq i \leq n$. If $L_i = \neg B$ then there is a negative edge from A to B, denoted $\langle A, B, - \rangle$. We say that A **depends** on B, denoted by $A < B$, if there is a proper path from A to B in the graph. We say that A **depends negatively** on B, denoted by $A <_1 B$, if there is a path from A to B containing at least one negative edge. A program P is called **locally stratified** if the relation $<_1$ in the dependency graph of P is well-founded.

2.2 First-Order Languages

We use L to range over languages of first-order predicate logic. We take the connectives of first-order logic to be $\neg, \wedge$ and $\exists$. When using $\vee, \rightarrow, \leftrightarrow$ and $\forall$ in formulas, these formulas are simply abbreviations of formulas containing only $\neg, \wedge$ and $\exists$. We require all languages L to contain the one-place predicate symbols T and $\mathcal{P}$. $\mathcal{P}$ will be used as a predicate that picks out a set of (codes of) formulas in L. T will, depending on the context, be used to express one of our syntactic attitudes *belief* or *knowledge*. By $L - \{T\}$ we denote the language L with the predicate symbol T removed. We will assume that all considered languages contain a parametrised coding. By a **parametrised coding** in L we understand an injective map $\ulcorner \cdot \urcorner$ from the formulas of L into the terms of L satisfying:

(i) For any formula φ in L, the term $\ulcorner \varphi \urcorner$ has the same free variables as φ (but $\ulcorner \varphi \urcorner$ is not itself a variable).

(ii) For any formula $\varphi(x)$ in L and any term τ which is free for x in $\varphi(x)$, $\ulcorner \varphi(\tau) \urcorner$ is the term obtained by substituting τ for all free occurrences of x in $\ulcorner \varphi(x) \urcorner$.

(iii) The coding is *well-founded*, that is, there is no infinite sequence of formulas $\varphi_0, \varphi_1, \varphi_2 \ldots$ such that $\ulcorner \varphi_{i+1} \urcorner$ is a term in φ_i for all $i \in \mathbb{N}$.

We refer to [Feferman, 1984] for the construction of a parametrised coding. Feferman's coding does not satisfy (ii), but a simple variant of it will. $\ulcorner \varphi \urcorner$ is called the **code** of φ. The intended interpretation of a formula $T(\ulcorner \varphi \urcorner)$ is that "φ is known" or "φ is believed". We assume all first-order languages L to contain the language of Peano arithmetic. Throughout the paper, by *formal arithmetic* we mean Robinson's arithmetic, though any other standard formalisation of arithmetic could have been used in its place. We identify first-order languages L with their sets of sentences. By a *sentence* in L we understand a closed formula, that is, a formula without any occurrences of free variables. The set of ground terms of L is denoted $\mathrm{Terms}(L)$.

To avoid confusion between formulas of propositional programs and formulas of first-order languages we will use Latin letters for the former and Greek letters for the latter.

2.3 Regular Formulas and RPQ Formulas

We now define the sets of first-order sentences which we intend to prove that our axiom schemes of knowledge and belief can consistently be instantiated with.

Definition 1. *Let L be a first-order language. The set of **regular formulas** of L is the least set satisfying:*

(i) Any atomic formula of $L - \{T\}$ is a regular formula.

(ii) If φ and ψ are regular formulas and x is a variable, then $\varphi \wedge \psi$, $\neg\varphi$ and $\exists x \varphi$ are regular formulas.

(iii) If φ is a regular formula then $T(\ulcorner \varphi \urcorner)$ is a regular formula.

Our definition differs slightly from the one given by Morreau and Kraus [1998]. Instead of using a parametrised coding, they have an $(n+1)$-place predicate symbol T^n for each n, such that instead of writing $T(\ulcorner \varphi(x_1, \ldots, x_n) \urcorner)$, where $x_1, \ldots, x_n$ are the free variables of φ, they would be writing $T^n(\ulcorner \varphi \urcorner, x_1, \ldots, x_n)$ where $\ulcorner \cdot \urcorner$ is then a standard (non-parametrised) Gödel coding. To simplify matters, we have chosen to take care of the free variables by using a parametrised coding rather than by introducing infinitely many predicate symbols of different arities.

As an example of a regular formula we have, for a suitable choice of L,

$$\exists x T(\ulcorner \mathit{telephone}(\mathit{Mike}) = x \urcorner),$$

expressing that the agent knows Mike's telephone number. If we have more than one agent, we can of course introduce a predicate symbol T_i for each agent i. In that case the following sentence also becomes regular:

$$T_1(\ulcorner \forall x (T_2(\ulcorner \mathit{departure\text{-}time}(\mathit{train}, x) \urcorner) \rightarrow \mathit{departure\text{-}time}(\mathit{train}, x)) \urcorner),$$

expressing that agent 1 believes agent 2 to have correct beliefs about the departure time of the train. As an example of a non-regular formula we have

$$T_{\text{user}}(\ulcorner \forall x (\mathit{utter}(\mathit{system}, x) \rightarrow T_{\text{system}}(x)) \urcorner),$$

expressing that the user believes that the system only utters what it believes to be the case. It is non-regular because T_{system} is applied directly to a variable and not to the code of a formula. To allow expressing beliefs such as this one, Morreau and Kraus [1998] extended the set of regular formulas to a more inclusive class called the RPQ formulas.

Definition 2. *Let L be a first-order language. The set of **RPQ formulas** of L is the least set satisfying:*

(i) Any atomic formula of $L - \{T\}$ is an RPQ formula.

(ii) If φ and ψ are RPQ formulas and x is a variable, then $\varphi \wedge \psi$, $\neg\varphi$ and $\exists x\varphi$ are RPQ formulas.

(iii) If φ is an RPQ formula then $T(\ulcorner\varphi\urcorner)$ is an RPQ formula.

*(iv) If φ is **any formula** in L, then $\exists x\,(\mathcal{P}(x) \wedge \varphi)$ is an RPQ formula.*

This definition also differs from the one given by Morreau and Kraus [1998]. It defines a slightly more inclusive set of formulas, and at the same time it is simpler, since it avoids Morreau and Kraus' use of two distinct collections of variables.

By **regular sentence** we understand a closed regular formula, and by **RPQ sentence** a closed RPQ formula.

3 Review of Previous Results

Consider the following axiom schemes in a first-order language L:

A1. $T(\ulcorner\varphi\urcorner) \rightarrow \varphi$

A2. $T(\ulcorner T(\ulcorner\varphi\urcorner) \rightarrow \varphi\urcorner)$

A3. $T(\ulcorner\varphi \rightarrow \psi\urcorner) \rightarrow (T(\ulcorner\varphi\urcorner) \rightarrow T(\ulcorner\psi\urcorner))$

A4. $T(\ulcorner\varphi\urcorner)$, if φ is a theorem in formal arithmetic.

A5. $T(\ulcorner\varphi\urcorner) \rightarrow T(\ulcorner T(\ulcorner\varphi\urcorner)\urcorner)$

A6. $\neg T(\ulcorner\varphi \wedge \neg\varphi\urcorner)$

As already mentioned, $T(\ulcorner\varphi\urcorner)$ is intended to denote either "φ is known" or "φ is believed". Thus, for instance, the first axiom scheme expresses that everything known (believed) is true. It seems reasonable to characterise knowledge by the axiom schemes A1–A4 and belief by A2–A6. But the following theorem shows that this is not always possible.

Theorem 3 (Montague [1963], Thomason [1980]). *Let L be a first-order language. Formal arithmetic extended with any of the following sets of axioms is **inconsistent**.*

(a) The axiom schemes of knowledge, A1–A4, instantiated over the sentences of L.

(b) The axiom schemes of belief, A2–A6, instantiated over the sentences of L.

The inconsistency of (a) is Montague's result, and the inconsistency of (b) is Thomason's result. A way out of these inconsistency results is to restrict the set of sentences that we instantiate A1–A6 with. This strategy gives us the following positive results.

Theorem 4 (des Rivières and Levesque [1988]). *Let L be a first-order language. Formal arithmetic extended with any of the following sets of axioms is **consistent**.*

(a) The axiom schemes of knowledge, A1–A4, instantiated over the regular sentences of L.

(b) The axiom schemes of belief, A2–A6, instantiated over the regular sentences of L.

Theorem 5 (Morreau and Kraus [1998]). *Theorem 4 still holds when we replace "regular sentences" with "RPQ sentences".*

Theorem 4 is proved in [des Rivières and Levesque, 1988] by a careful translation from a corresponding first-order modal logic. Theorem 5 is proved in [Morreau and Kraus, 1998] by a similar translation from a corresponding second-order modal logic. In the following section we give proofs of their results taking a completely different route. We show that the problems can be reduced to problems of consistency of particular logic programs.

Instead of working directly with the axiom schemes A1–A6, we will most of the time be working with the *truth scheme* which is the following axiom scheme:

$$T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi.$$

This is often sufficient since, as the following lemma shows, instances of axiom schemes A1–A6 are logical consequences of corresponding instances of the truth scheme. To prove that the axiom schemes A1–A6 instantiated over a set of sentences S are consistent, it thus suffices to prove the consistency of the truth schema instantiated over that same set.

Lemma 6. *Let L be a first-order language, and let S be a set of sentences in L satisfying:*

if φ and ψ are in S then $T(\ulcorner\varphi\urcorner), \neg\varphi$ and $\varphi \wedge \psi$ are in S.

Let $\mathcal{M}$ be a model of L in which $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ holds for all φ in S. Then all of A1–A6 hold in $\mathcal{M}$ for all φ, ψ in S.

Proof. That A1 holds in $\mathcal{M}$ when φ is in S is a trivial consequence of the fact that $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ holds in $\mathcal{M}$. To see that A2 holds in $\mathcal{M}$, we first note that if φ is in S then $\neg\,(T(\ulcorner\varphi\urcorner) \wedge \neg\varphi)$ is in S as well, by assumption on S. This sentence is an abbreviation of $T(\ulcorner\varphi\urcorner) \rightarrow \varphi$, so we get that the following instance of the truth schema holds in $\mathcal{M}$:

$$T(\ulcorner T(\ulcorner\varphi\urcorner) \rightarrow \varphi\urcorner) \leftrightarrow (T(\ulcorner\varphi\urcorner) \rightarrow \varphi).$$

Using this together with the fact that A1 holds in $\mathcal{M}$, we get that $T(\ulcorner T(\ulcorner\varphi\urcorner) \rightarrow \varphi\urcorner)$ holds in $\mathcal{M}$. That is, A2 holds in $\mathcal{M}$. A3–A6 are proved to hold in $\mathcal{M}$ in a similar manner. □

4 From LP Semantics to Consistent Treatments of Knowledge and Belief

The results of this paper are based on the following lemma.

Lemma 7 (Przymusinski [1987], Sato [1990]). *If a program P is locally stratified then $\mathrm{comp}(P)$ has a Herbrand model.*

Our formulation is taken from [Sato, 1990]. It should be noted that Sato is not considering infinite programs in his paper, but his proof carries over without modification to this more general framework. This is because Sato is considering the set of ground instances of non-propositional programs rather than these programs themselves. The set of ground instances of a finite non-propositional program is in general an infinite propositional program, that is, the kind of logic program we are considering in this paper.

Definition 8. *Let L be a first-order language, and let S be a set of sentences in L. We define an infinite program $P_{L,S}$ as follows. For every sentence φ in L, the program $P_{L,S}$ contains a propositional atom denoted p_φ. The clauses of $P_{L,S}$ are given by:*

$$\begin{array}{ll} p_{\varphi\wedge\psi} \leftarrow p_\varphi \wedge p_\psi, & \text{for all } \varphi,\psi \in L. \\ p_{\neg\varphi} \leftarrow \neg p_\varphi, & \text{for all } \varphi \in L. \\ p_{\exists x\varphi(x)} \leftarrow p_{\varphi(\tau)}, & \text{for all } \exists x\varphi(x) \in L \text{ and } \tau \in \mathrm{Terms}(L). \\ p_{T(\ulcorner\varphi\urcorner)} \leftarrow p_\varphi, & \text{for all } \varphi \in S. \end{array}$$

The relation between models of the program $P_{L,S}$ and models of the first-order language L is given by the following lemma.

Lemma 9. *Let L and S be as above. If $\mathrm{comp}(P_{L,S})$ has a Herbrand model $\mathcal{M}$ then L has a Herbrand model $\mathcal{N}$ satisfying:*

(i) For every sentence φ in L,

$$\mathcal{M} \models p_\varphi \;\Leftrightarrow\; \mathcal{N} \models \varphi. \tag{1}$$

(ii) $\mathcal{N} \models T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$, for all $\varphi \in S$.

Proof. Assume $\mathcal{M}$ is a model of $\mathrm{comp}(P_{L,S})$. $\mathrm{comp}(P_{L,S})$ is the following set of equivalences:

$$p_{\varphi\wedge\psi} \leftrightarrow p_\varphi \wedge p_\psi, \text{ for all } \varphi,\psi \in L. \tag{2}$$

$$p_{\neg\varphi} \leftrightarrow \neg p_\varphi, \text{ for all } \varphi \in L. \tag{3}$$

$$p_{\exists x\varphi(x)} \leftrightarrow \bigvee\nolimits_{\tau\in\mathrm{Terms}(L)} p_{\varphi(\tau)}, \text{ for all } \exists x\varphi(x) \in L. \tag{4}$$

$$p_{T(\ulcorner\varphi\urcorner)} \leftrightarrow p_\varphi, \text{ for all } \varphi \in S. \tag{5}$$

Let $\mathcal{N}$ be the following Herbrand model of L:

$$\mathcal{N} = \{\varphi \in L \mid \varphi \text{ is an atom and } \mathcal{M} \models p_\varphi\}.$$

(i) is proved by induction on the syntactic complexity of φ. If φ is an atom then (1) holds by definition of $\mathcal{N}$. To prove (1) for sentences of the form $\varphi \wedge \psi$, $\neg\varphi$ and $\exists x\varphi(x)$ we simply use (2), (3) and (4), respectively. For the case of $\neg\varphi$ the proof is:

$$\mathcal{M} \models p_{\neg\varphi} \Leftrightarrow \mathcal{M} \models \neg p_\varphi \Leftrightarrow \mathcal{M} \not\models p_\varphi \overset{\text{i.h.}}{\Leftrightarrow} \mathcal{N} \not\models \varphi \Leftrightarrow \mathcal{N} \models \neg\varphi,$$

where the first equivalence is by (3) and the third is by induction hypothesis. The two remaining cases are proved similarly. Thus (i) holds. Furthermore, using (i) and (5), we get for all $\varphi \in S$:

$$\mathcal{N} \models T(\ulcorner\varphi\urcorner) \Leftrightarrow \mathcal{M} \models p_{T(\ulcorner\varphi\urcorner)} \Leftrightarrow \mathcal{M} \models p_\varphi \Leftrightarrow \mathcal{N} \models \varphi,$$

and thus $\mathcal{N} \models T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$, proving (ii). □

Lemma 10. *Let L be a first-order language and let R be the set of regular sentences in L. The propositional program $P_{L,R}$ is locally stratified.*

Proof. To simplify matters we will throughout this proof be identifying every propositional letter p_φ with the corresponding first-order sentence φ in L. It should always be clear from the context whether φ is used to denote the first-order sentence or the corresponding propositional letter. Thus, by the identification, the nodes of the dependency graph of $P_{L,R}$ are all sentences in L. The edges are:

- $\langle \varphi\wedge\psi, \varphi, + \rangle$ and $\langle \varphi\wedge\psi, \psi, + \rangle$, for all $\varphi,\psi \in L$.
- $\langle \neg\varphi, \varphi, - \rangle$, for all $\varphi \in L$.
- $\langle \exists x\alpha(x), \alpha(\tau), + \rangle$, for $\exists x\alpha(x) \in L$ and $\tau \in \mathrm{Terms}(L)$.
- $\langle T(\ulcorner\varphi\urcorner), \varphi, + \rangle$, for all $\varphi \in R$.

Edges of the first type are called *$\wedge$-edges*, edges of the second type are called *$\neg$-edges*, edges of the third type are called *$\exists$-edges* and edges of the last type *T-edges*.

We have to prove that $P_{L,R}$ is locally stratified. Actually, we will be proving something slightly stronger. We will prove that the relation $<$ in the dependency graph of $P_{L,R}$ is well-founded. That is, we will prove that there does not exist any path of infinite length in the graph. Assume the opposite, that is, assume the existence of an infinite path σ.

Claim. σ contains infinitely many T-edges.

Proof of claim. Assume the opposite. Then there will be an infinite subpath σ' of σ containing no T-edges. Thus all edges on σ' must be $\wedge$-, $\neg$- or $\exists$-edges. But note that for any such edge, the start node will have higher syntactic complexity than the end node. Thus, along σ' the syntactic complexity will be strictly decreasing, which contradicts σ' being infinite. This proves the claim. ◊

With every formula φ in L we now associate a natural number $d(\varphi)$, called the *T-degree* of φ. The T-degree is defined recursively by

- $d(\varphi) = 1 + d(\psi)$, if $\varphi = T(\ulcorner\psi\urcorner)$ for some ψ.
- $d(\varphi) = 0$, if φ is any other atomic formula.
- $d(\varphi) = \max\{d(\psi) \mid \psi \text{ is a subformula of } \varphi\}$, otherwise.

The well-foundedness of the parametrised coding ensures that d is well-defined. By the above claim, σ contains an infinite number of T-edges. Let φ be the end node of such an edge. Then φ is regular. Let σ' be the infinite subpath of σ having φ as its start node. Then every node on σ' must be a regular formula (c.f. the definition of a regular formula). This implies that every edge on σ' is

(i) either a $\wedge$-, $\neg$- or T-edge,

(ii) or of type $\langle \exists x\alpha(x), \alpha(\tau), + \rangle$, where $\alpha(x)$ does not contain $T(x)$ as a subformula.

Item (ii) follows from that fact that when x is a variable then $T(x)$ is not a regular formula, and therefore no formula having $T(x)$ as a subformula can be regular either. Now note that on any edge of type (i) or (ii), the T-degree of the end node will be less than or equal to the T-degree of the start node. Thus the T-degree will be monotonically decreasing along σ' and must therefore be constant from some point. But then from this point it can not contain any T-edges, since the T-degree of the end node of such an edge is always one less than the T-degree of the start node. This contradicts the claim above. □

Lemma 11. *Let L be a first-order language and let S be a set of sentences in L. If $P_{L,S}$ is locally stratified then any Herbrand model of $L-\{T\}$ can be expanded into a Herbrand model of L in which $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ holds for all φ in S.*

Proof. Let $\mathcal{M}$ denote a Herbrand model of $L - \{T\}$. Let P be the program $P_{L,S}$ extended with the following clauses:

$p_\varphi \leftarrow \mathit{true}$, if φ is an atom in $L - \{T\}$ and $\mathcal{M} \models \varphi$.

$p_\varphi \leftarrow \mathit{false}$, if φ is an atom in $L - \{T\}$ and $\mathcal{M} \models \neg\varphi$.

$P_{L,S}$ is assumed to be locally stratified, and since P has the same dependency graph as $P_{L,S}$, then P must be locally stratified as well. Therefore $\mathrm{comp}(P)$ has a Herbrand model $\mathcal{M}'$, by Lemma 7. Finally, Lemma 9 gives us the existence of a Herbrand model $\mathcal{N}$ of L in which the equivalences $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ hold for all φ in S. To see that $\mathcal{N}$ expands $\mathcal{M}$ we just have to note that if φ is an atom in $L - \{T\}$ then

$$\mathcal{M} \models \varphi \Rightarrow p_\varphi \leftarrow \mathit{true} \in P \Rightarrow \mathcal{M}' \models p_\varphi \Rightarrow \mathcal{N} \models \varphi$$

and

$$\mathcal{M} \models \neg\varphi \Rightarrow p_\varphi \leftarrow \mathit{false} \in P \Rightarrow \mathcal{M}' \models \neg p_\varphi \Rightarrow \mathcal{N} \models \neg\varphi,$$

where the last implications are by (i) in Lemma 9. □

Theorem 12. *Let L be a first-order language and let U be a theory in $L - \{T\}$ containing formal arithmetic. If U has a Herbrand model then U extended with any of the following sets of axioms has a Herbrand model.*

(i) The axiom scheme $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ instantiated over the regular sentences of L.

(ii) The axiom schemes of knowledge, A1–A4, instantiated over the regular sentences of L.

(iii) The axiom schemes of belief, A2–A6, instantiated over the regular sentences of L.

Proof. Assume U has a Herbrand model $\mathcal{M}$. Let R denote the set of regular sentences in L. By Lemma 10, $P_{L,R}$ is locally stratified. So by Lemma 11, $\mathcal{M}$ can be expanded into a Herbrand model $\mathcal{N}$ in which $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ holds for all regular φ. This proves (i). (ii) and (iii) then immediately follow, using Lemma 6. □

Theorem 4 is an immediate consequence of Theorem 12, when taking U to be formal arithmetic. The machinery we have introduced can also be applied to prove Theorem 5. It is an immediate consequence of the following corollary to Theorem 12.

Corollary 13. *Theorem 12 still holds when we replace "regular sentences" with "RPQ sentences". Furthermore, the extension of $\mathcal{P}$ in the Herbrand model constructed will be the set of codes of regular sentences.*

Proof. Let S denote the set of RPQ sentences of L. Modify the program $P_{L,S}$ by removing every clause of the form

$$p_{\exists x(\mathcal{P}(x)\wedge\varphi(x))} \leftarrow p_{\mathcal{P}(\tau)\wedge\varphi(\tau)},$$

where τ is not the code of any regular sentence. Call the new program $Q_{L,S}$. It is easy to see that $Q_{L,S}$ is locally stratified, using the argument given in the proof of Lemma 10. Lemma 11 still holds when we use $Q_{L,S}$ instead of $P_{L,S}$, so any Herbrand model of U can be expanded into a model of L in which $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ holds for all RPQ sentences. This proves (i) in Theorem 12 with "regular sentences" replaced by "RPQ sentences". (ii) and (iii) then follows from Lemma 6. □

5 Strengthening the Results

We now strengthen the results obtained above. We want to define a set of formulas more inclusive than the RPQ formulas that the truth scheme $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$ can safely be instantiated with. For this we need a couple of new definitions.

Definition 14. *Let L be a first-order language and let φ be a formula in L. The set of formulas **occurring** in φ is defined as the least set containing φ and satisfying:*

- *If β is a subformula of a formula α occuring in φ, then β is occuring in φ.*
- *If $T(\ulcorner\alpha\urcorner)$ is occurring in φ, then α is occurring in φ.*

*Assume ψ is a formula occurring in φ. The occurrence is said to be **negative** if ψ occurs in a formula α where $\neg\alpha$ occurs in φ. Otherwise the occurrence is called **positive**. An occurence of ψ in φ is said to be **protected** if ψ occurs in a formula α where $\exists x\,(\mathcal{P}(x) \wedge \alpha)$ occurs in φ.*

Thus, for instance, φ occurs in formulas such as $T(\ulcorner\varphi\urcorner) \wedge \neg\psi$ and $T(\ulcorner\neg T(\ulcorner\varphi\urcorner) \wedge \psi\urcorner)$ but not in $A(\ulcorner\varphi\urcorner)$ when $A \neq T$. φ has positive occurrence in $T(\ulcorner\varphi\urcorner) \wedge \neg\psi$ but negative occurrence in $T(\ulcorner\neg T(\ulcorner\varphi\urcorner) \wedge \psi\urcorner)$.

Definition 15. *Let L be a first-order language. A formula φ in L is said to be **weakly RPQ** if, for any variable x, the formula $T(x)$ only occurs positively or protected in φ.*

Note that in an RPQ formula, every occurrence of $T(x)$ for some variable x is protected, so every RPQ formula is also weakly RPQ. Thus the set of RPQ formulas is a subset of the set of weakly RPQ formulas. It is furthermore a proper subset, since among the weakly RPQ formulas we have e.g. $\exists x(\mathit{about\_love}(x) \wedge T(x))$, which is not RPQ. The previously obtained results can be extended to the weakly RPQ formulas.

Theorem 16. *Let L be a first-order language and let U be a theory in $L - \{T\}$. If U has a Herbrand model, then U extended with the axiom scheme*

$$T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi$$

instantiated over the set of weakly RPQ sentences has a Herbrand model.

Proof. Let S denote the set of weakly RPQ formulas. Using Lemma 11, it suffices to prove that $P_{L,S}$ is locally stratified. As in the proof of Corollary 13, we can consider the modified program $Q_{L,S}$ instead. To obtain a contradiction, assume $Q_{L,S}$ is not locally stratified. Then $<_1$ in the dependency graph of $Q_{L,S}$ is not well-founded, that is, there must exist an infinite path σ containing infinitely many negative edges. As in the proof of Lemma 10, we get that σ must contain infinitely many T-edges. Let φ be the end node of such an edge. Then φ is weakly RPQ. Let σ' be the infinite subpath of σ having this node as its start node. Then every node on σ' must be weakly RPQ. As noted in the proof of Lemma 10, if every edge on σ' is

(i) either a $\wedge$-, $\neg$- or T-edge,

(ii) or of type $\langle \exists x\alpha(x), \alpha(\tau), + \rangle$ where $\alpha(x)$ does not contain $T(x)$ as a subformula,

then σ' can not be infinite. Thus, every node on σ' must have an occurrence of $T(x)$ for some variable x. Since all nodes are weakly RPQ, in each of these $T(x)$ is either positive or protected. But if $T(x)$ occurs protected in φ, there can be no infinite path starting at φ. Thus, in every formula φ on σ', $T(x)$ must occur positively (for some variable x). But this implies that all edges in the path are positive, which contradicts our assumption. □

The above theorem also relates to a result by Perlis [1985]. Perlis showed that a modified truth scheme $T(\ulcorner\varphi\urcorner) \leftrightarrow \varphi^*$ is consistent with arithmetic. It is easily seen that the set of instances of the (unmodified) truth scheme shown to be consistent by Perlis' result is contained in the set of instances shown to be consistent by Theorem 16.

In view of the results by Montague and Thomason (Theorem 3), there is a limit to how many instances of our axiom schemes we can add while still retaining consistency. The set of weakly RPQ sentences is quite close to this limit, as is made clear by the following example.

Example. Let L be a first-order language containing three one-place predicate symbols A, $\mathcal{P}$ and T. One of the simplest examples of a formula which is not weakly RPQ in L is the formula $\exists x\,(A(x) \wedge \neg T(x))$. Let us call this formula ψ. In ψ, $T(x)$ occurs negatively and unprotected (since $A \neq \mathcal{P}$). Let U be the theory consisting of the following axioms

$$A(\ulcorner\psi\urcorner) \tag{6}$$

$$\neg A(\tau), \text{ when } \tau \text{ is a term} \neq \ulcorner\psi\urcorner. \tag{7}$$

U obviously has a Herbrand model $\mathcal{M}$, but we will show that U extended with the single axiom

$$T(\ulcorner\psi\urcorner) \leftrightarrow \psi \tag{8}$$

does not have a Herbrand model. This shows that Theorem 16 no longer holds if we to the weakly RPQ sentences add a sentence such as ψ. Assume, to obtain a contradiction, that there exists a Herbrand model $\mathcal{N}$ in which all of (6), (7) and (8) holds. Then we obtain the following contradiction:

$$\mathcal{N} \models \psi \Leftrightarrow \mathcal{N} \models \exists x(A(x) \wedge \neg T(x)) \Leftrightarrow$$

$$\mathcal{N} \models A(\tau) \wedge \neg T(\tau) \text{ for some } \tau \overset{(7)}{\Leftrightarrow}$$

$$\mathcal{N} \models A(\ulcorner\psi\urcorner) \wedge \neg T(\ulcorner\psi\urcorner) \overset{(6)}{\Leftrightarrow} \mathcal{N} \models \neg T(\ulcorner\psi\urcorner) \Leftrightarrow$$

$$\mathcal{N} \not\models T(\ulcorner\psi\urcorner) \overset{(8)}{\Leftrightarrow} \mathcal{N} \not\models \psi.$$

6 Conclusion

We have been showing that results on the consistency of the predicate approach to knowledge and belief can be proved through the use of well-known results from logic programming semantics. This connects the two research fields in a new and interesting way, and have furthermore allowed us to strengthen the previously known results on the consistency of the predicate approach. It is expected that the connection between the two fields can be pursued further to get even better consistency results. This might be done by using some of the results from the literature on logic programming semantics that strengthen Lemma 7.

Acknowledgments

I would like to thank the following persons for guiding me in the direction that led to this paper: Jørgen Fischer Nilsson, Mai Gehrke, João Alexandre Leite and Ken Satoh. Special thanks to Roy Dyckhoff for his interest in my work and his many valuable comments.

References

[Attardi and Simi, 1995] Guiseppe Attardi and Maria Simi. A formalization of viewpoints. *Fundamenta Informaticae*, 23(2-4):149–173, 1995.

[Carlucci Aiello *et al.*, 1995] Luigia Carlucci Aiello, Marta Cialdea, Daniele Nardi, and Marco Schaerf. Modal and meta languages: consistency and expressiveness. In *Meta-logics and logic programming*, pages 243–265. MIT Press, 1995.

[Davies, 1990] Nick Davies. A first order logic of truth, knowledge and belief. *Lecture Notes in Artificial Intelligence*, 478:170–179, 1990.

[des Rivières and Levesque, 1988] Jim des Rivières and Hector J. Levesque. The consistency of syntactical treatments of knowledge. *Computational Intelligence*, 4:31–41, 1988.

[Feferman, 1984] Solomon Feferman. Toward useful type-free theories I. *The Journal of Symbolic Logic*, 49(1):75–111, 1984.

[Grant *et al.*, 2000] John Grant, Sarit Kraus, and Donald Perlis. A logic for characterizing multiple bounded agents. *Autonomous Agents and Multi-Agent Systems*, 3(4):351–387, 2000.

[McCarthy, 1997] John McCarthy. Modality si! Modal logic, no! *Studia Logica*, 59(1):29–32, 1997.

[Montague, 1963] Richard Montague. Syntactical treatments of modality, with corollaries on reflection principles and finite axiomatizability. *Acta Philosophica Fennica*, 16:153–166, 1963.

[Morreau and Kraus, 1998] Michael Morreau and Sarit Kraus. Syntactical treatments of propositional attitudes. *Artificial Intelligence*, 106(1):161–177, 1998.

[Perlis, 1985] Donald Perlis. Languages with self-reference I. *Artificial Intelligence*, 25:301–322, 1985.

[Przymusinski, 1987] T. Przymusinski. On the declarative semantics of deductive databases and logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 193–216. Morgan Kaufmann, 1987.

[Sato, 1990] Taisuke Sato. Completed logic programs and their consistency. *Journal of Logic Programming*, 9(1):33–44, 1990.

[Thomason, 1980] Richmond H. Thomason. A note on syntactical treatments of modality. *Synthese*, 44(3):391–395, 1980.