Tractable reasoning about Agent Programming in Dynamic Preference Logic

Paper #666

ABSTRACT

An interpretation of mental attitudes of a BDI programming language is given as a mapping of program states to a BDI logic. This mapping help us to understand how the semantics of the constructs of the language relate to the concepts of the BDI paradigm. While several BDI logic have been proposed for this effect, it is not clear how models in some of these logics can be connected to the agent programs they are supposed to specify. More yet, being based on modal logic, the reasoning problems in theses logics are not tractable in general, limiting their usage to tackle real-world problems. In this work, we use of Dynamic Preference Logic to provide a semantic foundation to BDI agent programming languages and investigate tractable expressive fragments of this logic to reason about agent programs. With that, we aim to provide a way of implementing semantically grounded agent programming languages with tractable reasoning cycles.

KEYWORDS

BDI logic, Agent Programming, Dynamic Epistemic Logic

1 INTRODUCTION

In the study of rational action and agency, several different logics and formal theories for practical reasoning have been proposed. Particularly, the Belief, Desire Intention framework [6] has become a popular approach to practical reasoning in the areas of Artificial Intelligence and Autonomous Agents, giving rise to the construction of several programming languages and computer systems based on it.

Having a formal definition of the semantics is essential for proving properties about a programming language and also for assert certain properties of specific programs. For agent programming languages, a formal definition has also the advantage of clarifying the notion of agency imbued within the language semantics as well as providing a formal framework to specify and verify systems' properties and behaviour.

Recently, it has been proposed that Dynamic Preference Logic (DPL) can be used to reason about BDI Agent Programming with declarative mental attitudes [27]. Exploring representation results for this logic, the authors show that agent programs can be straightforwardly converted into models of the logic, which in turn can always be encoded by means of an agent program. This yield a computable two-way translation between specifications in the logic and agent programs.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 2018, Stockholm, Sweden

https://doi.org/doi

Dynamic Preference Logic is a dynamic modal logic which has been applied to study several different mental attitudes and their related phenomena [2, 3, 22–24] and is particularly interesting due to its great expressiveness and ability to distinguish between mental actions, i.e. actions changing the agent's mental state, and ontic actions, i.e. actions changing the environment in which the agent resides.

Reasoning in DPL is, however, not tractable in general. In fact, the satisfiability problem in DPL, and even in its static fragment, is PSPACE-hard[1]. More yet, reasoning about agent programs in this logic, as proposed in [27], involves performing exponentially many propositional satisfiability checks - which is a NP-complete problem in itself - at each step in the agent's reasoning cycle. As such, the proposal of using DPL as a logic for reasoning about agent programming - while theoretically relevant for the analysis of a programming language semantics - is of very limited practical use.

In this work, we investigate expressive fragments of the language of DPL that yield tractable reasoning problems. The reasoning problems discussed in this paper are concerned with knowing whether an agent knows (believes, desires or intends) a certain propositional property φ and how to compute the resulting mental state of an agent after performing a belief/desire revision or contraction. With this, we aim to provide a tractable fragment that may be used to implement an actual agent programming language with declarative mental attitudes having a well-defined logical semantics based on Kripke frames.

This work is structured as follows: in Section 2 we present the logic of agency proposed here, based on Dynamic Preference Logic; in Section 3 we discuss the connection between the logic proposed and Agent Programming. In Section 4, we discuss a tractable expressive subset of the language which can be used to implement agent programming languages. In section 5, we present the related work and compare their contributions to ours. Finally, in Section 6, we present our final considerations.

2 A DYNAMIC LOGIC OF FOR BDI AGENT PROGRAMMING

In this section, we propose a propositional modal logic of agency, constituted of modalities to represent the epistemic and conative state of an agent, as well as the causal effects governing the performance of actions in one's environment. Based on X [27], we use this logic to provide specifications of mental attitudes. Later, we 'dynamify' the language, i.e. extend the language with dynamic modalities to represent mental actions.

We will suppose a BDI agent has a library of plans describing which actions she can perform on the environment. For the sake of simplicity, we will assume that the plans are deterministic, completely specified and STRIPS-like. This means that we consider that

[1] This can be easily seen by the fact that DPL has a S4 modality [$\leq$].

plans are effective and atomic, i.e. not decomposable, and the agent does not reason about how to refine her intentions, but only how to select a plan to achieve her goals. Notice that these are not severe restrictions to our analysis, since we model only the reasoning *about* actions not its performance and perception. Aware of these restrictions, we introduce the notion of plan library.

Definition 2.1. *We call $\mathcal{A} = \langle \Pi, pre, pos\rangle$ a plan library, iff Π is a finite set of plans symbols, $pre : \Pi \to \mathcal{L}_0$ is a function that maps each plan to a propositional formula representing its preconditions and $pos : \Pi \to \mathcal{L}_0$ the function that maps each plan to a propositional formula representing its post-conditions. We further require that the post-conditions of any plan is a consistent conjunction of propositional literals. We say $\alpha \in \mathcal{A}$ for any plan symbol $\alpha \in \Pi$.*

Notice that similar notions of action/plan library can be found in the related literature, such as in works about classical planning [7] and in agent programming languages such as AgentSpeak(L) [17]. With this definition in mind, we can establish the language we will use as a base for our constructions.

Definition 2.2. *Let P be a set of propositional symbols and $\mathcal{A} = \langle \Pi, pre, pos\rangle$ a plan library. We define the language $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$ by the following grammar (where $p \in P$ and $\alpha \in \mathcal{A}$):*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid A\varphi \mid [\leq_P]\varphi \mid [<_P]\varphi \mid [\leq_D]\varphi \mid [<_D]\varphi \mid [\alpha]\varphi \mid I\alpha$$

In this language, we use the symbol A to denote the universal modality and the symbols $[\leq_P]$ and $[\leq_D]$ to denote box modalities over some plausibility and desirability orders in the model. The symbols $[<_P]$ and $[<_D]$ will be used to refer to the strict part of those orders. As usual, we will define $E\varphi \equiv \neg A\neg\varphi$ and $\langle\leq_\Box\rangle\varphi \equiv \neg[\leq_\Box]\neg\varphi$ with $\Box \in \{P, D\}$. The formula $[\leq_D]\varphi$ ($[\leq_P]\varphi$) means that in all words equally or more desirable (plausible) than the current one, φ holds and $[<_D]\varphi$ ($[<_P]\varphi$) that in all words strictly more desirable (plausible) than the current one, φ holds. Finally, the formulas $[\alpha]\varphi$ and $I\alpha$ mean that after carrying out the plan α, the property φ holds, and that the agent intends to execute a plan α, respectively. In the remainder of this work, we will denote by $\mathcal{L}_0$ the language of propositional logic, i.e. the language obtained by removing all modal formulas from $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$.

To interpret these formulas, we will introduce a new kind of Kripke model containing two accessibility relations - one for plausibility and one for desirability. We will call this new model an *agent model* - named this way since such a model represent the mental state of an agent.

Definition 2.3. *Let $\mathcal{A} = \langle \Pi, pre, pos\rangle$ be a plan library, an agent model is a tuple $M = \langle W, \leq_P, \leq_D, I, v\rangle$ where W is a set of possible worlds, and both $\leq_D$ and $\leq_P$ are pre-orders over W with well-founded strict parts $<_P$ and $<_D$, $I \subseteq \Pi$ is a set of adopted plans (or intentions) and v is a valuation function.*

To model the effect of executing a plan $\alpha \in \mathcal{A}$ given an agent model M, we will define the notion of model update, as commonly used in the area of Dynamic Epistemic Logic.

Definition 2.4. *Let $\mathcal{A} = \langle \Pi, pre, pos\rangle$ be a plan library, $\alpha \in \mathcal{A}$ a plan and $M = \langle W, \leq_P, \leq_D, I, v\rangle$ an agent model. The product update of model M by execution of plan α is defined as the model $M \otimes [\mathcal{A}, \alpha] = \langle W', \leq'_P, \leq'_D, I', v'\rangle$ where*

$$\begin{array}{lcl} W' & = & \{w \in W \mid M, w \vDash pre(\alpha)\} \\ \leq'_P & = & \leq_P \cap\, W' \times W' \\ \leq'_D & = & \leq_D \cap\, W' \times W' \\ I' & = & I \\ v'(p) & = & \begin{cases} W' & \text{if } pos(\alpha) \vDash p \\ \emptyset & \text{if } pos(\alpha) \vDash \neg p \\ v(p) \cap\ W' & \text{otherwise} \end{cases} \end{array}$$

The interpretation of the formulas is defined as usual, with each modality corresponding to an accessibility relation.

$$\begin{array}{ll} M, w \vDash [\leq_P]\varphi & \text{iff } \forall w' \in W : w' \leq_P w \Rightarrow M, w' \vDash \varphi \\ M, w \vDash [<_P]\varphi & \text{iff } \forall w' \in W : w' <_P w \Rightarrow M, w' \vDash \varphi \\ M, w \vDash [\leq_D]\varphi & \text{iff } \forall w' \in W : w' \leq_D w \Rightarrow M, w' \vDash \varphi \\ M, w \vDash [<_D]\varphi & \text{iff } \forall w' \in W : w' <_D w \Rightarrow M, w' \vDash \varphi \\ M, w \vDash [\alpha]\varphi & \text{if } M, w \vDash pre(\alpha) \text{ then } M \otimes [\mathcal{A}, \alpha], w \vDash \varphi \\ M, w \vDash I\alpha & \text{iff } \alpha \in I \end{array}$$

We define the formula $\mu_P\varphi \equiv (\varphi \wedge \neg\langle<_P\rangle\varphi)$ (similarly, $\mu_D\varphi$ is defined exchanging $\langle<_P\rangle$ for $\langle<_D\rangle$) which is satisfied only by the minimal worlds according to the order $\leq_P$ (similarly for $\leq_D$) which satisfy the formula φ. These formulas will be useful to encode mental attitudes in this logic.

2.1 Enconding mental attitudes

Following X [27], we introduce a codification of mental attitudes in the language $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$. In this work, we interpret the notion of 'possible world' as epistemically possible, not metaphisically possible. As such, the universal modality can be used to encode the knowledge held by an agent (we adopt here a S5 notion of knowledge).

$$K\varphi \equiv A\varphi$$

We encode the (KD45) notion of belief as what is true in the worlds that the agent believes to be the most plausible ones. As such, our notion $B\varphi$ means that '*it is most plausible that φ*.'

$$B\varphi \equiv A(\mu_P\top \to \varphi)$$

Encodings of the notion of desire are numerous in the literature with various meanings according to the intended application. We will require that agent's desires are consistent with each other - a common requirement in logical modelling of desires in Agent Programming. Similar to belief, we propose a codification of desires as everything that is satisfied in all most desirable worlds. In other way, we want to encode a formula $G(\varphi)$ meaning that "*in the most desirable worlds, φ holds*".

$$G(\varphi) \equiv A(\mu_D\top \to \varphi)$$

Our language possesses the notion of procedural intentions by the formula $I\alpha$. To encode Bratman's [6] notion of prospective intention, however, we will define a formula $Int\varphi$. First however, we most encode the restrictions imposed by Bratman for consistency of an intention by means of a formua $AdmInt(\varphi)$ meaning that '*it is admissible for the agent to intend that φ*'.

$$AdmInt(\varphi) \equiv G(\varphi) \wedge E(\varphi) \wedge \neg B(\varphi)$$

With this notion, we can define the notion of having an '*intention that φ*.'

$$Int(\varphi) \equiv AdmInt(\varphi) \wedge \bigvee_{\alpha \in \mathcal{A}} (I\alpha \wedge B\,(pre(\alpha) \wedge [\alpha]varphi))$$

Notice that, while we imposed several restrictions in our codification for an agent to rationally hold some prospective intention, none of these restrictions have been required for an agent to hold a procedural intention, i.e. an *intention to do* - here represented by the set of adopted plans I in the model. These restrictions, however, do hold for procedural intentions as well as for prospective intentions in Bratman's philosophy of action. To model the kind of agent that satisfies Bratman's restrictions, we define the notion of a coherent agent model, i.e. an agent having a coherent mental state.

Definition 2.5. *Let $\mathcal{A}$ be a plan library and $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be an agent model. We say a set $I \subset \Pi$ of plans is $\mathcal{A}$-coherent in M if for all $\alpha \in I$, $M \vDash B(pre(\alpha))$ and $M \vDash AdmInt(pos(\alpha))$. If I is $\mathcal{A}$-coherent in M, we say M is a coherent agent model.*

It is easy to see by our construction that procedural intentions, i.e. *intentions to do*, and prospective intentions, i.e. *intentions that*, are well-connected.

Proposition 2.6. *Let $\mathcal{A}$ be a plan library and $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be a coherent agent model, it holds that*

$$M, w \vDash I\alpha \Rightarrow M, w \vDash B(pre(\alpha)) \wedge Int(pos(\alpha))$$

2.2 Dynamic operations on agent's mental states

Once established the basic language and the encodings of mental attitudes, we define some well-behaved mental operations, which will be used to implement agents' practical reasoning. To define this operations, we will include in our language dynamic modalities such as $[\Uparrow_P \varphi]\psi$ to mean that "*after the radical upgrade of the plausibility relation by φ, ψ* holds". Here we explore three dynamic operations on agent models, each a representative of the three basic mental operations as studied by the Belief Revision Theory [1]: expansion, revision and contraction.

The first operation we introduce is that of public announcement, defined by Plaza [15]. This operation corresponds, in a sense, to the operation of expansion from Belief Revision Theory. Based in the codifications we provided in the previous section, this operation can be interpreted as the mental operation of knowledge acquisition.

Definition 2.7. *Let $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be a coherent agent model and φ a formula of $\mathcal{L}_0$. We say the model $M_{!\varphi} = \langle W_{!\varphi}, \leq_{P!\varphi}, \leq_{D!\varphi}, I_{!\varphi}, v_{!\varphi}\rangle$ is the result of public announcement of φ in M, where:*

$$\begin{array}{lcl} W_{!\varphi} & = & \{w \in W \mid M, w \vDash \varphi\} \\ \leq_{\Box!\varphi} & = & \leq_{\Box} \cap (W^2_{!\varphi}) \\ I_{!\varphi} & & \text{is the maximal } \mathcal{A}-\text{coherent subset of } I \\ v_{!\varphi}(p) & = & v(p) \cap W_{!\varphi} \end{array}$$

The radical upgrade of an agents beliefs by an information φ results in a model such that all worlds satisfying φ are deemed more plausible than those not satisfying it. This operation corresponds to a operation of belief revision from belief Revision Theory. In fact, it is equivalent to Segerberg [21]'s irrevocable revision.

Definition 2.8. *Let $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be a coherent agent model and φ a formula of $\mathcal{L}_0$. We say the model $M_{\Uparrow\varphi} = \langle W, \leq_{P\Uparrow\varphi}, \leq_D, I_{\Uparrow_P\varphi}, v\rangle$ is the result of the radical upgrade on the plausibility of of M by φ, where*

$$\begin{array}{ll} \leq_{P\Uparrow\varphi} = & (\leq \setminus \{\langle w, w'\rangle \in W^2 \mid M, w \nvDash \varphi \text{ and } M, w' \vDash \varphi\}) \cup \\ & \{\langle w, w'\rangle \in W^2 \mid M, w \vDash \varphi \text{ and } M, w' \nvDash \varphi\} \\ I_{\Uparrow_P\varphi} & \text{is the maximal } \mathcal{A}-\text{coherent subset of } I \end{array}$$

We can similarly define the radical upgrade of the agents desires by the operation $\Uparrow_D \varphi$, which updates the desirability relation, instead of the plausibility relation.

Lastly, we introduce the operation of lexicographic contraction, based on the work of iterated contraction functions [16]. This operation corresponds to perform a re-ordering of the worlds in a way that both φ and $\neg\varphi$ are considered equally plausible (or equally desirable) to the agent. To define this operation in a more elegant, we will define the notion of the plausibility degree of a world.

Definition 2.9. *Let $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be a coherent agent model, $w \in W$ a possible world and $\varphi \in \mathcal{L}_{\leq}(P, \mathcal{A})$ a formula. We say that w has plausibility degree $n \in \mathbb{N}$, denoted $n = dP_{\varphi}(w)$, if $M, w \vDash \varphi$ and there is a maximal chain $w_0, w_1, w_2, \ldots, w_i$ satisfying φ s.t. w_0 is a minimal world satisfying φ and $w_n = w$. In other worlds, there is a maximal chain $w_0 <_P w_1 <_P w_2 <_P \ldots <_P w_n$ s.t. $w_i \in [\![\varphi]\!]$, $w_0 \in Min_{\leq}[\![\varphi]\!]$ and $w_n = w$. If $w \notin [\![\varphi]\!]$, we say that $dP_{\varphi}(w) = \infty$.*

We can define the desirability degree of a world w in φ, denoted by $dD_{\varphi}(w)$, the same way, taking the chain on the preference relation $\leq_D$ instead of the relation $\leq_P$. With that notion, we define the lexicographic contraction as below.

Definition 2.10. *Let $M = \langle W, \leq_P, \leq_D, I, v\rangle$ be a coherent agent model and φ a formula of $\mathcal{L}_0$. We say the model $M_{\Downarrow_P\varphi} = \langle W, \leq_{P\Downarrow\varphi}, \leq_D, I_{\Downarrow_P\varphi}v\rangle$ is the lexicographic contraction of the plausibility of M by φ, where:*

$$w \leq_{P\Downarrow\varphi} w' \text{ iff } \begin{cases} w \leq w' & \text{if } w, w' \in [\![\varphi]\!] \\ w \leq w' & \text{if } w, w' \notin [\![\varphi]\!] \\ dP_{\varphi}(w) < dP_{\neg\varphi}(w') & \text{if } w \in [\![\varphi]\!] \text{ and } w' \notin [\![\varphi]\!] \\ dP_{\neg\varphi}(w) < dP_{\varphi}(w') & \text{if } w \notin [\![\varphi]\!] \text{ and } w' \in [\![\varphi]\!] \end{cases}$$

$I_{\Downarrow_P\varphi}$ is the maximal $\mathcal{A}-$ coherent subset of I

As before, we can similarly define the lexicographic contraction on the desirability of M by φ (denoted by $M_{\Downarrow_D\varphi}$) using the desirability degrees of the worlds, instead of their plausibility degrees. These operations correspond to the contraction (or abandonment) of a belief/desire by the agent.

For each operation $\star$ defined above, we introduce in our language a new modality $[\star\varphi]\psi$ in our language, meaning "*after the operation of $\star$ by φ, ψ holds*". which can be interpreted as

$$M, w \vDash [\star\varphi]\psi \qquad \text{iff} \qquad M_{\star\varphi}, w \vDash \psi$$

An important result about the dynamified logic is that, if we consider some special kind of models, which includes the models we will use in Section 3 to reason about Agent Programming , it has the same expressibility as the static logic presented before [27]. In fact, the formulas $[!\varphi]\psi$, $[\Uparrow_{\Box} \varphi]\psi$ and $[\Downarrow_{\Box} \varphi]\psi$ are definable in the language $\mathcal{L}_{\leq}(P, \mathcal{A})$.

2.3 An Example

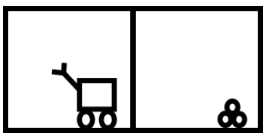

Figure 1: Example of a cleaning robot

Consider the example of a cleaning robot in a house composed of two rooms, as depicted Figure 1. The robot can see the room it is currently inside, it can move between the rooms, and it can clean the room.

We can use the proposed logic to specify the running example depicted in Figure 1. We will use the following set of propositional variables to describe our example:

$$P = \{clean(room_1), clean(room_2), at(room_1), at(room_2)\}$$

The first panel of Figure 1 can be described by the agent model M composed of four worlds w_1, w_2, w_3 and w_4, with the following valuation functions:

$$v(w_1) = \{clean(room_1), \neg clean(room_2), at(room_1), \neg at(room_2)\}$$
$$v(w_2) = \{clean(room_1), clean(room_2), at(room_1), \neg at(room_2)\}$$
$$v(w_3) = \{clean(room_1), \neg clean(room_2), \neg at(room_1), at(room_2)\}$$
$$v(w_4) = \{clean(room_1), clean(room_2), \neg at(room_1), at(room_2)\}$$

These worlds are ordered by the following plausibility and desirability relations, here we denote by $w \equiv_{\square} w'$ the fact that $w \leq_{\square} w'$ and $w' \leq_{\square} w$.

$$w1 \equiv_P w_2 <_P w_3 \equiv_P w_4$$
$$w2 \equiv_D w_4 <_D w_1 \equiv_D w_3$$

The agent described by M has the set of intentions $I = \{Clean_1(room_2)\}$ s.e.t $pre(Clean_1(room_2)) = at(room_1)$ and $pos(Clean_1(room_2)) = at(room_2) \wedge clean(room_2)\}$.

Clearly, M is a coherent agent model and it holds that $M \models K(at(room_1) \leftrightarrow \neg at(room_2))$, i.e. the agent knows it can only be at one room at a time. Also, the agent believes that it is currently at room 1, i.e. it holds that $M \models B(at(room_1))$, the agent desires that both rooms are clean, i.e. it holds that $M \models G(clean(room_1) \wedge clean(room_2))$ and the agent intends for $romm_2$ to be clean, i.e. $M \models Int(clean(room_2)$. Also, after being informed that the room 2 is, indeed, not clean, the agent comes to update her plausibility relation (by means of a radical upgrade) in the following way:

$$w1 <_P w_3 <_P w_2 <_P w_4$$
$$w2 \equiv_D w_4 \leq_D w_1 \equiv_D w_3$$

3 REASONING ABOUT BDI AGENTS USING DYNAMIC PREFERENCE LOGIC

An interesting property of Preference Logic - the logic used as a foundation to construct $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$ - is that preference models can be encoded by means of some structures known as priority graphs [14]. Exploring this connection, we will show how we can use agent programs with stratified mental attitudes, e.g. beliefs annotated with their credence/plausibility level, to obtain a model for the agent's mental state.

In most BDI agent programming languages, an agent program is defined by means of a tuple $ag = \langle K, B, D, I\rangle$, where K, B and G are sets of (ranked) propositional formulas representing the agent's knowledge, beliefs and desires, respectively, and I is a set of plans adopted by the agent. Since a set of (ranked) formulas is nothing more than an order over formulas, we can construct an agent structure $\mathcal{G}$ which induces an agent model $M_{\mathcal{G}}$ representing the mental state of the agent program ag.

Definition 3.1. *Let P be a set of propositional variables and $\mathcal{A}$ a plan library. We call an agent program over $\mathcal{A}$, a tuple $ag = \langle K, B, D, I\rangle$ where:*

- *$K \subset \mathcal{L}_0(P)$, is a finite set of propositional formulas called the knowledge base;*
- *$B \subset \mathcal{L}_0(P) \times \mathbb{N}^*$ is a finite set of pairs $\langle \varphi, i\rangle$, called a stratified belief base, where φ is a propositional formula and i is a natural number, called the plausibility or rank of φ in B.*
- *$D \subset \mathcal{L}_0(P) \times \mathbb{N}^*$ is a finite set of pairs $\langle \varphi, i\rangle$, called a stratified goal base, where φ is a propositional formula and i is a natural number, called the desirability or rank of φ in D.*
- *$I \subset \mathcal{A}$ is a finite set of plans, called the (procedural) intention base.*

When the plan library $\mathcal{A}$ is clear, we will often call the tuple $ag = \langle K, B, D, I\rangle$ an agent program.

An agent program is a tuple $ag = \langle K, B, D, I\rangle$ describing the agent's knowledge base, belief base, desire base and (procedural) intention base, i.e. adopted plans, respectively. As such, we define the mental attitudes of an agent, i.e. what she knows, believes, etc. by means of them.

Definition 3.2. *Let $ag = \langle K, B, D, I\rangle$ be an agent program and $\varphi \in \mathcal{L}_0$. We say agent 'ag knows φ', denoted by $ag \models K\varphi$, iff $K \models \varphi$. We call $Know(ag) = \{\varphi \in \mathcal{L}_0 \mid K \models \varphi\}$ the knowledge set of agent ag.*

Notice that our belief and desire bases are stratified, in the sense that the beliefs/desires of an agent are ranked from the most plausible/desirable to the less plausible/desirable. Since some of these beliefs may be contradictory with each other, we must compute the maximal set of beliefs/desires that is consistent - respecting the stratification of the base.

Definition 3.3. *Let $\Gamma \subset \mathcal{L}_0 \times \mathbb{N}$ be a finite set of pairs $\langle \varphi, i\rangle$ and let $\Gamma_i = \{\varphi \mid \langle \varphi, i\rangle \in B\}$. We define the maximal consistent subset of Γ, the set $\Gamma^{Max} \subset \mathcal{L}_0$, s.t.*

- *$\Gamma^{Max} \subseteq \bigcup \Gamma_i$ and if $\langle \varphi, i\rangle \in \Gamma$ and $\varphi \in \Gamma^{Max}$ then $\Gamma_i \subseteq \Gamma^{Max}$;*
- *$\forall \Gamma' \subseteq \Gamma : (\exists \Gamma_i \subseteq \Gamma' \wedge \Gamma_i \nsubseteq \Gamma^{Max} \Rightarrow \Gamma' \models \bot$ or $\exists \Gamma_j \subseteq \Gamma^{Max} \wedge \Gamma_j \nsubseteq \Gamma'$ and $j < i)$*

With this in mind, we can provide interpretations of the notions of belief and desire by means of such bases.

Definition 3.4. *Let $ag = \langle K, B, D, I\rangle$ be an agent program and $\varphi \in \mathcal{L}_0$. We say 'agent ag believes in φ', denoted by $ag \models B\varphi$, iff $B^{Max} \models \varphi$. We denote by $Bel(ag) = Cn(B_{Max})$ the belief set of agent ag.*

Similarly for the agent's desires.

Definition 3.5. *Let $ag = \langle K, B, D, I\rangle$ be an agent program and $\varphi \in \mathcal{L}_0$. We say that 'the agent ag desires φ', denoted $ag \models G\varphi$, iff $D^{Max}\cup \models \{\varphi\}$. We denote by $Goal(ag) = \{\varphi \in \mathcal{L}_0 \mid ag \models G\varphi\}$ the goal set of the agent ag.*

Finally, by means of the agent's procedural intentions, we can define her intentions.

Definition 3.6. *Let $ag = \langle K, B, D, I\rangle$ be an agent program and $\varphi \in \mathcal{L}_0$. We say 'the agent ag intends φ', denoted by $ag \models I\varphi$, iff $ag \models G\varphi$ and $\exists \alpha \in I$, s.t. $pos(\alpha) \models \varphi$. We denote by $Int(ag) = \{\varphi \in \mathcal{L}_0 \mid ag \models I\varphi\}$ the (declarative) intention set of ag*

While we placed no condition on agent programs, since in this work we adhere to Bratman's [6] notion of intention, our declarative mental attitudes must satisfy some constraints in order for the agent to be considered rational.

Definition 3.7. *Let $ag = \langle K, B, D, I\rangle$ be an agent program. We say ag is coherent iff all of the conditions below hold.*

1. *the agent's knowledge is consistent, i.e. $K \not\models \bot$*
2. *the agent's knowledge is consistent with her beliefs, i.e. $\varphi \in K$ iff $\langle \varphi, 0\rangle \in B$*
3. *the agent is not delusional, i.e. her desires are consistent with her knowledge, i.e. $\varphi \in K$ iff $\langle \varphi, 0\rangle \in D$*
4. *for any plan $\alpha \in I$, there is a goal $\varphi \in D^{Max}$, such that $pos(\alpha) \models_{\mathcal{L}_0} \varphi$;*
5. *the plans of the agent are pursuable: $\forall \alpha \in I\ Bel(ag) \models pre(\alpha)$;*
6. *the plans of the agent are jointly-consistent: $\{pos(\alpha) \mid \alpha \in I\} \not\models \bot$;*
7. *the plans of the agent are relevant: $\forall \alpha \in I, Bel(ag) \not\models pos(\alpha)$.*

Liu [14] shows that preference relations can be equivalently represented by means of syntactical constructs, known as priority graphs. A priority graph, however, is nothing more than a partial order over propositional formulas, much similar to the stratified bases we introduced here. As such, we can use the same reasoning to compute the plausibility and desirability orders of an agent model by means belief and desire bases.

Definition 3.8. *Let $\Gamma \subset \mathcal{L}_0 \times \mathbb{N}$ be a stratified base, W a set of possible worlds and $v : \mathcal{L}_0 \to W$ a valuation function. Considering $\Gamma_i = \{\varphi \mid \langle \varphi, i\rangle \in \Gamma\}$ and $w \models X$ to stand for $\forall \varphi \in X : (w \in v(\varphi))$, then we define the pre-order $\leq_\Gamma \subseteq W \times W$ s.t.*

$$w \leq_\Gamma w' \text{ iff } \quad \forall i \in \mathbb{N} : \quad (w' \models \Gamma_i \Rightarrow w \models \Gamma_i) \vee (\exists j < i \text{ s.t. } (w \models \Gamma_j \text{ and } w' \not\models \Gamma_j))$$

Using this construction, we are able to construct an agent model from an agent program.

Definition 3.9. *Let $ag = \langle K, B, D, I\rangle$ be an agent program, we define the model induced by ag as $M_{ag} = \langle [\![K]\!], \leq_B, \leq_D, I, v\rangle$ where $[\![K]\!] \subset 2^P$ are all the propositional valuations that satisfy the set K, $\leq_B \subset [\![K]\!] \times [\![K]\!]$ and $\leq_D \subseteq [\![K]\!] \times [\![K]\!]$ are the preference relations induced by the bases B and D, and $w \in v(p)$ iff $p \in w$.*

Finally, since preference relations can be always be encoded as priority graphs [14], we can always compute agent programs describing mental models.

Proposition 3.10. *Let $M = \langle W, \leq_P \leq_D, I, v\rangle$ be an agent model, with $W \subseteq 2^P$, then there is an agent program $ag = \langle K, B, D, I\rangle$ s.t $M = M_{ag}$. More yet, M is a coherent agent model iff ag is a coherent agent program.*

From this result and the encodings of mental attitudes in both the logic and in agent programs, it is not difficult to see that the mental notions coincide.

Corollary 3.11. *Let $ag = \langle K, B, D, I\rangle$ be a coherent agent program and $\varphi \in \mathcal{L}_0$ be a propositional formula, then*

$$\begin{array}{lll} M_{ag} \models B(\varphi) & \text{iff} & B^{Max} \models \varphi \\ M_{ag} \models G(\varphi) & \text{iff} & D^{Max} \models \varphi \\ M_{ag} \models I\alpha & \text{iff} & \alpha \in I \\ M_{ag} \models Int(\varphi) & \text{iff} & ag \models I\varphi \end{array}$$

At this point, we have two considerations to make about the codification presented here. First, regarding the complexity of reasoning about agent programs attitudes, notice that to compute an agents beliefs (or goals), it requires a linear number of propositional satisfiability checks on the depth of the base, i.e. on the higher rank appearing on the base. Second, regarding the notion of mental attitudes encoded here, notice that we adopted a very simple notion of goal as a maximal set of consistent desires - consistent with other works in BDI programming [9, 18]. It is not difficult, however, to treat other notions, such as that of Van Riemsdijk et al [25], in our framework. To do so, it suffices to redefine the notion of maximal consist subset of a stratified base provided in Definition 3.3 and the respective encoding in the logic.

The choice we made in this work was based both on the simplicity of the resulting framework, due to a reduced complexity in reasoning[2] and on the fact that the difference between the two formalisations disappear under the restrictions proposed in Section 4. As such, given our objective of investigating tractable fragments of the logic which are useful in practice, we believe these choices are justified.

3.1 Revisiting the example

We can use our agent programs to specify the running example depicted in Figure 1. We will use the following set of propositional variables to describe our example:

$$P = \{clean(room_1), clean(room_2), at(room_1), at(room_2)\}$$

The first panel of Figure 1 can be described by the coherent agent program $ag = \langle K, B, D, I\rangle$ where

$$\begin{array}{lcl} K & = & \{at(room_1) \leftrightarrow \neg at(room_2), clean(romm_1)\} \\ B & = & \{\langle at(room_1) \leftrightarrow \neg at(room_2), 0\rangle, \langle clean(room_1), 0\rangle, \\ & & \langle at(room_1), 1\rangle\} \\ D & = & \{\langle at(room_1) \leftrightarrow \neg at(room_2), 0\rangle, \langle clean(room_1), 0\rangle, \\ & & \langle clean(room_1), 1\rangle, \langle clean(room_2), 1\rangle\} \\ I & = & \{Clean_1(room_2)\} \text{ with } pre(Clean_1(room_2)) = at(room_1) \\ & & \text{and } pos(Clean_1(room_2)) = at(room_2) \wedge clean(room_2) \end{array}$$

After being informed that $room_2$ is not clean, the robot comes to believe that $room_2$ is not clean, represented by the program

[2] To compute Van Riemsdijk's goals in an agent program it is necessary to perform a exponential number of propositional satisfiability checks on the size of the base, which can be significantly higher than the depth of the base.

$ag_{\Uparrow_P \neg clean(room_2)} = \langle K, B', D, I' \rangle$ where:

$$
\begin{array}{lcl}
K & = & \{at(room_1) \leftrightarrow \neg at(room_2), clean(romm_1)\} \\
B' & = & \{\langle at(room_1) \leftrightarrow \neg at(room_2), 0\rangle, \langle clean(room_1), 0\rangle, \\
 & & \langle \neg clean(room_2), 1\rangle, \langle at(room_1), 2\rangle\} \\
D & = & \{\langle at(room_1) \leftrightarrow \neg at(room_2), 0\rangle, \langle clean(room_1), 0\rangle, \\
 & & \langle clean(room_1), 1\rangle, \langle clean(room_2), 1\rangle\} \\
I' & = & \{Clean_1(room_2)\} \text{ with } pre(Clean_1(room_2)) = at(room_1) \\
 & & \text{and } pos(Clean_1(room_2)) = at(room_2) \wedge clean(room_2)
\end{array}
$$

4 TRACTABLE FRAGMENTS OF DPL

We have seen so far that we can use the logic $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$ to reason about agent programs. The computational complexity of reasoning about agents, however, is far too great to be useful for real-world problems. As such, programming languages based on the constructs proposed so far have very limited applicability - beyond a theoretical exercise.

In this section, we investigate some restrictions on the kinds of agent programs and agent models that guarantee that the reasoning problems in the resulting logic are tractable. As such, we search for a tractable fragment of our theory which can be used to as a semantic foundation to the construction of real agent programming languages with declarative mental attitudes.

Notice that, the agent programs we defined in Section 3 are composed by propositional formulas and reasoning using these programs usually involves checking propositional satisfiability of set of formulas. Propositional satisfiability, however, is a well-known NP-complete problem. As such, to obtain a restriction on agent programs that yield tractable reasoning problems, we must restrict ourselves to a fragment of propositional logic for which satisfiability is also tractable. Luckily, there are several well-known of such fragments which could be used. In our work, we adopt the conjunctive fragment, i.e. the fragment of propositional logic composed only of conjunction of literals. This fragment is tractable and retains good expressiveness to model agent reasoning. Let's then define the notion of a stratified base and of conjunctive agent program.

Definition 4.1. *Let $\Gamma \subset \mathcal{L}_0 \times \mathbb{N}$ be a stratified base, we say Γ is conjunctive iff for all $\langle \varphi, i\rangle \in \Gamma$, φ is a conjunction of literals, i.e. $\varphi = \bigwedge l_k$, with $l_k = p_k$ or $l_k = \neg p_k$.*

A conjunctive agent program is, thus, an agent program in which all of its bases are conjunctive.

Definition 4.2. *Let $ag = \langle K, B, D, I\rangle$ be an agent program, we say ag is a conjunctive agent program iff K is a set of conjunctive formulas, B and D are conjunctive stratified bases and for any plan $\alpha \in I$, $pre(\alpha)$ and $pos(\alpha)$ are conjunctive formulas.*

Notice that, by Definitions 3.2, 3.4 and 3.5, to decide whether an agent knows, believes, desires or intends a certain formula φ, we must perform propositional satisfiability checks involving the sets K, B^{Max}, D^{Max} and I. Since propositional satisfiability is tractable, to guarantee tractability of computing the mental attitudes of an agent, we must only guarantee we can compute the maximal consistent subsets, such as B^{Max} and D^{Max}, in polynomial time. To do so, we provide the Algorithm $Max(\Gamma)$, depicted in Figure 2.

Proposition 4.3. *Let $\Gamma \subset \mathcal{L}_0 \times \mathbb{N}$ be a conjunctive stratified base, then the algorithm Max presented in Figure 2 is correct and computes Γ^{Max} in $O(nm)$ time, where n is the size of Γ and m is the size of the biggest formula in Γ.*

Algorithm $Max(\Gamma)$
Input : a conjunctive stratified base Γ
Output : the maximal consistent subset Γ^{Max} of the base Γ

```
[1]  Γ^Max := {}
[2]  n := maximal depth of Γ
[3]  for i := 1 to n
[4]      Γ_i := {l | ⟨φ, i⟩ ∈ Γ and l appears in φ}
[5]      if ¬l ∈ Γ_i and l ∈ Γ_i for some l then
[6]          break
[7]      else
[8]          if l ∈ Γ_i and ¬l ∈ Γ^Max for some l then
[9]              continue
[10]         else
[11]             Γ^Max := Γ^Max ∪ Γ_i
[12] return Γ^Max
```

Figure 2: Algorithm for computing the maximal consistent subset of Γ.

As a consequence, we can always decide whether a conjunctive agent program knows (beliefs, desires or intends) a certain formula φ in polynomial time.

Corollary 4.4. *Let $ag = \langle K, B, D, I\rangle$ be a conjunctive agent program and $\varphi \in \mathcal{L}_0$ a propositional formula. We can compute whether $ag \models K(\varphi)$ ($ag \models B(\varphi)$ or $ag \models D(\varphi)$) in polynomial time in the size of K (B or D) and φ.*

Corollary 4.4 guarantees that the static properties of an agent are computed in polynomial time, i.e. that we can reason about the agent's mental state at any point in time in the program execution. The execution of an agent program, however, is usually determined by it reasoning cycle, i.e. the execution of certain mental changing operations that describe how an agent reason about her environment and how she makes choices. The mental operations are usually described by means of changes in the agents knowledge, beliefs, desires and (procedural) intentions, i.e. adopted plans.As such, to provide a truly tractable semantic framework to reason about agent programming, we must ensure that these mental changing operations can be computed in polynomial time.

We now dedicate our attention to this problem. We aim to provide tractable operations on agent programs to compute the dynamic operations discussed in Section 2. Since these dynamic operations are representative of the most relevant mental changing operations in the literature, we argue that any well-motivated reasoning cycle can be implemented with them - or changed only slightly to be implemented with them.

First, based on the work of Girard [10] and of Liu [14], let's show how we can compute knowledge acquisition - interpreted here as a public announcement - can be computed using agent programs.

Proposition 4.5. *Let $ag = \langle K, B, D, I\rangle$ be an agent program and $\varphi \in \mathcal{L}_0$, let yet $ag' = \langle K \cup \{\varphi\}, B', D', I'\rangle$, where*

$$\begin{aligned} B' = \quad & (B \cup \{\langle \varphi, 0\rangle\}) \\ D' = \quad & (D \cup \{\langle \varphi, 0\rangle\}) \\ I' = \quad & \{\alpha \in I \mid (B')^{Max} \vDash pre(\alpha) \text{ and } (B')^{Max} \nvDash pos(\alpha) \text{ and} \\ & \exists \varphi \in D' : pos(\alpha) \vDash \varphi\} \end{aligned}$$

be the agent program resulting of agent ag obtaining a knowledge φ. Then $M_{ag'} = M_{ag_{!\varphi}}$. We denote ag' by $ag_{!\varphi}$.

As a result of this encoding, we can compute knowledge acquisition/public announcement in polynomial time.

Corollary 4.6. *Let $ag = \langle K, B, D, I\rangle$ be a conjunctive agent program and $\varphi, \psi \in \mathcal{L}_0$ conjunctive propositional formulas. We can compute whether $ag_{!\varphi} \vDash K(\psi)$ ($ag_{!\varphi} \vDash B(\psi)$ or $ag_{!\varphi} \vDash D(\psi)$) in polynomial time in the size of K (B or D), φ and ψ.*

As Radical Upgrade can also be computed by means of transformation on the agent programs, we can represent the mental operation of belief revision in our framework.

Proposition 4.7. *Let $ag = \langle K, B, D, I\rangle$ be a coherent agent program and $\varphi \in \mathcal{L}_0$, let yet $ag' = \langle K, B', D, I'\rangle$, where*

$$\begin{aligned} B' = \quad & (B \cup \{\langle \varphi, 1\rangle\}) \\ I' = \quad & \{\alpha \in I \mid (B')^{Max} \vDash pre(\alpha) \text{ and } (B')^{Max} \nvDash pos(\alpha)\} \end{aligned}$$

be the agent program resulting of agent ag revising her beliefs by information φ. Then $M_{ag'} = M_{ag_{\Uparrow_P \varphi}}$. We denote ag' by $ag_{\Uparrow_P \varphi}$.

As a corollary, reasoning about the resulting mental state of the agent after belief revision is a tractable problem.

Corollary 4.8. *Let $ag = \langle K, B, D, I\rangle$ be a conjunctive agent program and $\varphi, \psi \in \mathcal{L}_0$ conjunctive propositional formulas, we can compute whether $ag_{\Uparrow_P \varphi} \vDash K(\psi)$ ($ag_{\Uparrow_P \varphi} \vDash B(\psi)$ or $ag_{\Uparrow_P \varphi} \vDash D(\psi)$) in polynomial time in the size of K (B or D), φ and ψ.*

A similar result can be stated for the radical upgrade of the agents desires, instead of beliefs. This operation represents the adoption of a given goal[3] .

For the lexicographic contraction, we can easily compute the result of contracting a conjunctive formula from a conjunctive stratified base by forgetting the propositional variables appearing in this formula. To implement this, we use the algorithm depicted in Figure 3.

Proposition 4.9. *Let $ag = \langle K, B, D, I\rangle$ be a coherent agent program and $\varphi \in \mathcal{L}_0$, let yet $ag' = \langle K, B', D, I'\rangle$, where*

$$\begin{aligned} B' = \quad & Cont(B, \varphi) \\ I' = \quad & \{\alpha \in I \mid (B')^{Max} \vDash pre(\alpha) \text{ and } (B')^{Max} \nvDash pos(\alpha)\} \end{aligned}$$

be the agent program resulting of agent ag contracting her beliefs by information φ. Then $M_{ag'} = M_{ag_{\Downarrow_P \varphi}}$, i.e. the algorithm $Cont(\Gamma, \varphi)$ depicted in Figure 3 is correct. We denote ag' by $ag_{\Downarrow_P \varphi}$.

As before, we can reason about the changes in the mental state of the agent after the contraction of a belief ,or similarly the withdraw of a goal, in polynomial time to the size of the agent program and the formulas.

[3]Remember that, as the goals of an agent are consistent, goal adoption must be performed by a revision operation, not an expansion, in the terminology of AGM [1].

Algorithm $Cont(\Gamma, \varphi)$
Input : a conjunctive stratified base Γ
a conjunctive propositional formula φ
Output : $\Gamma_{\Downarrow\varphi}$ the lexicographic contraction of Γ by φ

[1] $\Gamma_{\Downarrow\varphi}$:= {}
[1] **for each** $\langle \psi, i\rangle \in \Gamma$
[2] ψ' := ψ
[2] **for each** propositional symbol p appearing in φ
[2] ψ := $\psi[\top|\neg p][\top|p]$
[3] $\Gamma_{\Downarrow\varphi}$:= $\Gamma_{\Downarrow\varphi} \cup \{\langle \psi', i\rangle\}$
[12] **return** $\Gamma_{\Downarrow\varphi}$

Figure 3: Algorithm for computing the contraction of a base Γ by a formula φ.

Corollary 4.10. *Let $ag = \langle K, B, D, I\rangle$ be a conjunctive agent program and $\varphi, \psi \in \mathcal{L}_0$ conjunctive propositional formulas, we can compute whether $ag_{\Downarrow_P \varphi} \vDash K(\psi)$ ($ag_{\Downarrow_P \varphi} \vDash B(\psi)$ or $ag_{\Downarrow_P \varphi} \vDash D(\psi)$) in polynomial time in the size of K (B or D), φ and ψ.*

The results above state restrictions on the kind of agent programs that yield tractable reasoning about agent programming. Since agent programs can be understood as are syntactic representations of agent models, the results above may also be interpreted as a restriction on the agency logic $\mathcal{L}_{\leq_P, \leq_D}(P, \mathcal{A})$ that yileds tractable reasoning.

Corollary 4.11. *Let P be a finite set of propositional variables. Consider the class $\mathfrak{M}$ of coherent agent models $M = \langle W, \leq_P, \leq_D, I, v\rangle$ s.t. $\leq_P$ and $\leq_D$ are linear pre-orders and there is a injective function $f : W \to 2^P$, s.t.*

$$f(w) = X \text{ iff } \forall p \in P : (w \in v(p) \leftrightarrow p \in X).$$

For any conjunctive propositional formula $\varphi \in \mathcal{L}_0$ and $\Gamma \subset \{\Box\psi \mid \psi \in \mathcal{L}_0 \text{ and } \Box \in \{K, B, D\}\}$ a set of formulas, it is tractable to decide whether $\Gamma \vDash_{\mathfrak{M}} K(\varphi)$, $\Gamma \vDash_{\mathfrak{M}} B(\varphi)$, or $\Gamma \vDash_{\mathfrak{M}} D(\varphi)$.

Proof. Notice that Γ defines sets B, D and K of conjunctive propositional formulas, s.t. Γ is satisfiable iff $B \cup K$, $D \cup K$ are satisfiable. Clearly, for any conjunctive propositional formula ψ, $\Gamma \vDash_{\mathfrak{M}} K\psi$ iff $K \vDash \psi$ (similar to $B(\psi)$ and $B \cup K$, and $G(\psi)$ and $D \cup K$). Since K, B and D are sets of conjunctive formulas, it is tractable to decide whether $K \vDash \psi$ ($B \cup K \vDash \psi$ or $D \cup K \vDash \psi$). □

The result above states that in this restriction of the logic, logical consequence is tractable. More yet, it is easy to see that for this restricted logic, the model checking problem for static and dynamic mental formulas is also tractable.

Corollary 4.12. *Let P be a finite set of propositional variables. Consider the class $\mathfrak{M}$ as before. For any model $M \in \mathfrak{M}$, it is polynomial in the size of P, on the size of M and on the size the formulas φ and ψ to decide whether $M \vDash [\star\varphi]\Box\psi$, for any conjunctive propositional formulas $\varphi, \psi \in \mathcal{L}_0$, $\Box \in \{K, B, D\}$ and $\star \in \{!, \Uparrow_P, \Uparrow_D, \Downarrow_P, \Downarrow_D\}$.*

Proof. Notice that for any model M (necessarily finite), we can compute an equivalent agent program ag_M in time linear to

the size of the model times the number of propositional variables. Since the preference relations $\leq_{\Box}$ are linear, to compute the base $\Box$, it suffices to compute for each equivalence class number of variables that are satisfied in all worlds of the class. This can be computed in $O(|W| \cdot |P|)$. To compute K it suffices to find all the propositional symbols that are satisfied by all worlds in M - again it can be computed in $O(|W| \cdot |P|)$. As we have seen, we can compute $ag_{M \star \varphi} \vdash \Box\psi$ in time polynomial to the size of ag_M and on the size of φ and ψ. $\Box$

With this results we provided a restriction of the logic which for which reasoning about agents' mental states is tractable and provided a way to translate from agent models to agent programs.

5 RELATED WORK

From the Agent Programming perspective, the two most important works on modelling BDI mental attitudes are, in our opinion, the seminal work of Cohen and Levesque [8] and the work of Rao and Georgeff [19] describing the logic BDI-CTL. While their contribution to the area is undeniable, much criticism has been drawn to both approaches. Particularly, both approaches have proven to be difficult to connect with agent programming languages, by the use of a possible-world model semantics - vastly different from the syntactical representations used in agent programming.

Other work have also been proposed for studying the declarative interpretation of mental attitudes in concrete agent programming languages. Works as that of Wobcke [26] and of Hindriks and Van der Hoek [12] propose ways to connect the semantics of a given programming language to some appropriate logic to reason about agent's mental attitudes. While they are important in allowing us to analyse the mental attitudes diffused in the semantics of the language, since these logics cannot represent mental actions, the transformations in the agent program, which are defined in the programming language semantics, cannot be understood within the logic used to analyse these mental attitudes and thus the dynamic properties of these attitudes cannot be reasoned about in the logic. Also, in this approach, it is not clear how to establish the contrary connection, i.e. how to create or change programs to guarantee a certain property in the theory of intentions. In our work, since we can translate both ways, from the logic to agent programs and back, this is not an issue.

On the other way, works as that of Bordini and Moreira [4] present a declarative interpretation of BDI attitudes based on the actual implementation of these concepts in a concrete agent programming language. The aim of their work is to analyse Rao and Georgeff's [19] asymmetry properties in the formal semantics of the language AgentSpeak(L). The result is that, due to implementation considerations of the programming language, the logic suffers from a great expressibility limitation, not being able to represent several important properties about mental states.What is shown in their investigation is that, due to several expressive restrictions in the language, the procedural encoding of mental attitudes in some (early) agent programming languages is very far from the declarative concepts in which they are based.

Perhaps the work most related to ours in spirit is that of [13]. They propose a dynamic logic for agents and show that this logic can be understood as a verification logic, i.e. it has an equivalent state-based semantics based on the an operational semantics. The main difference of their approach to ours is that the authors choose to work in a framework closely related to situation calculus. The mental actions involved in decision making and in mental change are, thus, only implicitly defined, while the inclusion of such actions in the language is exactly the main advantage advocated by us. In some sense, our work can be seen as a generalisation of their work, since by employing Dynamic Preference Logic the equivalence they seek between operational semantics and declarative semantics can be automatically achieved by the results of Liu [14].

Recently, Herzig et al. [11] pointed out some deficiencies in the formal frameworks for specifying BDI agents which are available in the literature. The authors point out the advantages of a formal theory with a close relationship with the work in belief dynamics and with agent programming.

6 FINAL CONSIDERATIONS

Our work has investigated the use of a Dynamic Preference Logic to encode BDI mental attitudes and its connections to Agent Programming. More yet, we provided an expressive fragment of the logic for which reasoning about agents' mental states is tractable and how this can be computed by means of agent programs. With this, we believe we provided a roadmap to use Dynamic Preference Logic as a semantic framework to specify and also implement the formal semantics of BDI agent programming languages with declarative mental attitudes.

We wish to point out that, while we provide a fairly simple encoding of the mental attitudes in this worl, the logic discussed here is expressive enough to encode different notions of desires, goals and intentions. For example, we can represent the semantics of goals as proposed by Van Riemsdijk et al [25] in our framework.

Also notice that we mostly restrict our analysis to coherent agents. This restriction was done simply to ensure that the mental state represented by the agent models and agent programs are the same. To employ this logic to an agent program satisfying different constrains of mental coherency, it would be necessary to redefine in the logic the mental attitudes, we encoded in Section 2. We wish to point out, however, that our requirements for the semantics of the mental attitudes are very standard and, in fact, quite compatible with the philosophical requirements for these notions [6].

Regarding the requirements proposed by Herzig et al. [11] for a formal theory of agent programming, we believe our work tackles most of the problems identified by those authors. It remains, however, to provide a greater connection of our logics with the work areas as *planning* and *game theory*. We point out, however, that we have powerful evidences that such connections can be done, c.f. the work of Andersen et al. [2] on planning in the dynamic logics and that of Boutilier [5] and of Roy [20] for the connection between similar logics and game theory.

As a future work, we aim to implement a simple fragment of an agent programming language implementing declarative mental attitudes in this language by means of the codifications proposed in this work. We believe such an implementation can be used to understand the notions of mental attitudes imbued in this language.

REFERENCES

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[2] M. B. Andersen, T. Bolander, and M. H. Jensen. Don't plan for the unexpected: Planning based on plausibility models. *Logique et Analyse*, 1(1), 2014.

[3] A. Baltag and S. Smets. A qualitative theory of dynamic interactive belief revision. *Texts in logic and games*, 3:9–58, 2008.

[4] R. Bordini and A. Moreira. Proving BDI properties of agent-oriented programming languages: The asymmetry thesis principles in AgentSpeak (L). *Annals of Mathematics and Artificial Intelligence*, 42(1):197–226, 2004.

[5] C. Boutilier. Toward a logic for qualitative decision theory. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pages 75–86, New York, US, 1994. Morgan Kaufmann.

[6] M. E. Bratman. *Intention, plans, and practical reason*. Harvard University Press, Cambridge, US, 1999.

[7] T. Bylander. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.

[8] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[9] P. R. Cohen, J. L. Morgan, and M. E. Pollack. *Intentions in communication*. MIT press, Cambridge, US, 1990.

[10] P. Girard. *Modal logic for belief and preference change*. PhD thesis, Stanford University, 2008.

[11] A. Herzig, E. Lorini, L. Perrussel, and Z. Xiao. BDI logics for BDI architectures: old problems, new perspectives. *KI-Künstliche Intelligenz*, pages 1–11, 2016.

[12] K. Hindriks and W. Van der Hoek. Goal agents instantiate intention logic. In *Logics in Artificial Intelligence*, pages 232–244. Springer, New York, US, 2008.

[13] K. V. Hindriks and J.-J. C. Meyer. Toward a programming theory for rational agents. *Autonomous Agents and Multi-Agent Systems*, 19(1):4–29, 2009.

[14] F. Liu. *Reasoning about preference dynamics*, volume 354. Springer, New York, US, 2011.

[15] J. Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.

[16] R. Ramachandran, A. C. Nayak, and M. A. Orgun. Three approaches to iterated belief contraction. *Journal of philosophical logic*, 41(1):115–142, 2012.

[17] A. S. Rao. Agentspeak (l): BDI agents speak out in a logical computable language. In *Agents Breaking Away*, pages 42–55. Springer, New York, US, 1996.

[18] A. S. Rao and M. P. Georgeff. BDI agents: From theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems*, volume 95, pages 312–319, Palo Alto, US, 1995. AAAI Press.

[19] A. S. Rao and M. P. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–343, 1998.

[20] O. Roy. A dynamic-epistemic hybrid logic for intentions and information changes in strategic games. *Synthese*, 171(2):291–320, 2009.

[21] K. Segerberg. Irrevocable belief revision in dynamic doxastic logic. *Notre Dame journal of formal logic*, 39(3):287–306, 1998.

[22] J. Van Benthem. Dynamic logic for belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007.

[23] J. Van Benthem. For better or for worse: Dynamic logics of preference. In *Preference Change*, volume 42 of *Theory and Decision Library A:*, pages 57–84. Springer Netherlands, Dordrecht, NL, 2009.

[24] J. Van Benthem, D. Grossi, and F. Liu. Priority structures in deontic logic. *Theoria*, 80(2):116–152, 2014.

[25] M. B. Van Riemsdijk, M. Dastani, and J.-J. C. Meyer. Goals in conflict: semantic foundations of goals in agent programming. *Autonomous Agents and Multi-Agent Systems*, 18(3):471–500, 2009.

[26] W. Wobcke. Model theory for PRS-like agents: Modelling belief update and action attempts. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, pages 595–604. Springer-Verlag, Berlin, DE, 2004.

[27] X. *Reference Ommited for Anonimity Purposes*. PhD thesis, Anon University, 2016.