Logic-based cognitive planning for conversational agents

Jorge Luis Fernandez Davila, Dominique Longin, Emiliano Lorini, Frédéric Maris

Logic-based cognitive planning for conversational agents

Jorge Luis Fernandez Davila[1,2] · Dominique Longin[1,2] · Emiliano Lorini[1,2] · Frédéric Maris[1,2]

Abstract
This paper presents a novel approach to cognitive planning based on an NP-complete logic of explicit and implicit belief whose satisfiability checking problem is reduced to SAT. We illustrate the potential for application of our model by formalizing and then implementing a human–machine interaction scenario in which an artificial agent interacts with a human agent through dialogue and tries to motivate her to practice a sport. To make persuasion effective, the artificial agent needs a model of the human's beliefs and desires which is built during interaction through a sequence of belief revision operations. We consider two cog-nitive planning algorithms and compare their performances, a brute force algorithm based on SAT and a QBF-based algorithm.

Keywords Cognitive planning · Epistemic logic · Conversational agents

1 Introduction

In social sciences, influence is defined as "change in an individual's thoughts, feelings, atti-tudes, or behaviors that results from interaction with another individual or a group" [58]. It is conceived as tightly connected with persuasion. The latter is the intentional form of influence in which an agent (the persuader) tries to make someone (the persuadee) do or believe something by giving her a good reason [12, 54].

A natural way to model persuasion is by means of epistemic logic and of its application to planning, the so-called epistemic planning. On the one hand, epistemic logic is the

✉ Dominique Longin
Dominique.Longin@irit.fr

✉ Emiliano Lorini
Emiliano.Lorini@irit.fr

✉ Frédéric Maris
Frederic.Maris@irit.fr

Jorge Luis Fernandez Davila
jorge.fernandez@scienomics.com

[1] Department of Artificial Intelligence, Institut de Recherche en Informatique de Toulouse (IRIT), 118 route de Narbonne, 31062 Toulouse, France

[2] CNRS, Toulouse INP, UT3, Université de Toulouse, Toulouse, France

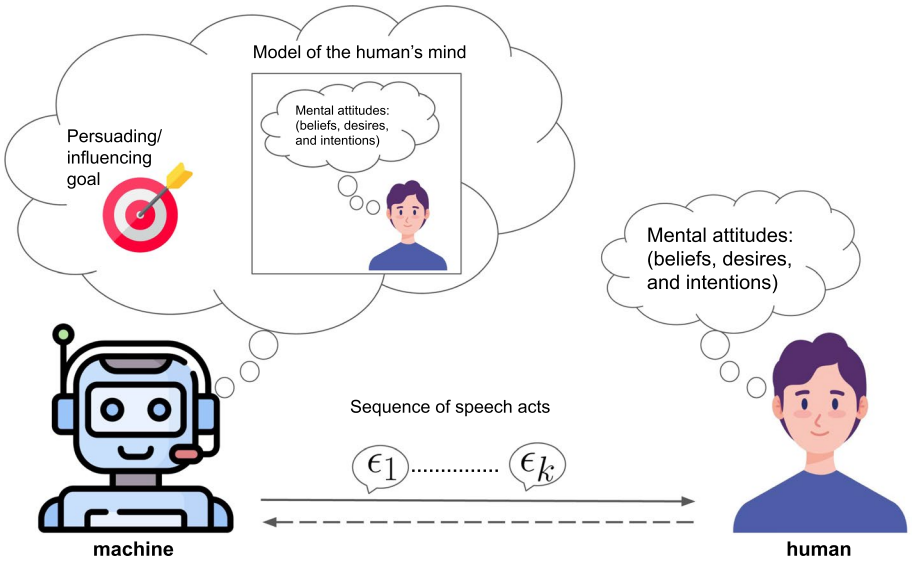


Fig. 1 Cognitive planning: conceptual schema

variant of modal logic which is devoted to the formal representation of epistemic attitudes of agents including their beliefs and knowledge. Since the pioneering work of Hintikka [27], it has been widely studied both in artificial intelligence (AI) [21, 46] and in economics [36]. It supports reasoning not only about propositional epistemic attitudes (that is, belief or knowledge about non epistemic facts) but also about higher-order epistemic attitudes (that is, belief or knowledge about some belief or knowledge—from the agent itself or from other agents). On the other hand, epistemic planning [6, 44] is a generalization of classical planning [23] whereby the goal to be achieved is not necessarily a state of the world but some belief states of one or more agents. This requires a theory of mind by the planning agent [25]. A typical goal in epistemic planning is to make a certain agent believe something.

In this work we adopt a broader perspective by using the term 'cognitive planning' instead of 'epistemic planning', where the former is considered to be a generalization of the latter. In cognitive planning, it is not only some belief state of a target agent that is to be achieved, but more generally a cognitive state. The latter could involve not only beliefs, but also intentions. Cognitive planning makes clear the distinction between *persuasion on beliefs* (i.e., inducing someone to believe that a certain fact is true) and *persuasion on intentions* (i.e., inducing someone to form a certain intention), generally called *influence*, and elucidates the connection between these two notions. Specifically, since beliefs are the input of decision-making and provide reasons for deciding and for acting, the persuader can indirectly change the persuadee's intentions by changing her beliefs, through the execution of a sequence of speech acts. In other words, in cogni-tive planning, the persuader tries to modify the persuadee's beliefs *in order to* affect persuadee's intentions. Moreover, cognitive planning takes into consideration resource boundedness and limited rationality of the interlocutor agent. This makes cognitive planning a very well-suited concept for human–machine interaction (HMI) applications in which an artificial agent is expected to interact with a human—who is by definition

resource-bounded—through dialogue and to induce her to behave in a certain way. These two aspects are exemplified in Fig. 1. The artificial agent has both (1) a model of the human's overall cognitive state, and (2) a persuading or influencing goal towards the human. Given (1) and (2), it tries to find a sequence of speech acts aimed at modifying the human's cognitive state thereby guaranteeing the achievement of its persuading/influencing goal.

We situate cognitive planning in a general architecture of an artificial agent which is expected to interact with a human user through dialogue and to motivate her to behave in a certain way or to change/adopt a certain style of life. From this perspective, cognitive planning is just a specific module of the architecture communicating with other modules including the agent's belief revision module and action execution module. For instance, the belief revision module handles the process of gathering information about the human's cognitive state including her belief and preferences. This information is needed to make the agent's cognitive planning successful. Indeed, for the agent to be able to persuade the human and to influence her behavior, it must have a correct representation of the human's cognitive state.

We formalize the cognitive planning and belief revision module of the architecture in the epistemic logic with operators for explicit and implicit belief recently proposed by [38, 39]. Since the full logic is known to be PSPACE-complete, we study an NP fragment of it that can be leveraged more easily for practical applications through a SAT solver. The logic allows us to represent, at the same time: the limited reasoning of the human agent (the persuadee), whose explicit beliefs are not necessarily closed under deduction, and the unbounded inferential capability of the artificial agent (the persuader), which is capable of computing the logical consequences of its explicit beliefs and of finding an optimal persuasion plan. It also enables us to represent in a natural way the artificial agent's belief revision process as well as its interrelationship with the agent's cognitive planning phase.

Some underlying assumptions of our approach have to be elucidated. We do not pretend to propose a general theory of cognitive planning that could be applied to any kind of multi-agent scenario involving multiple artificial and/or human agents. The latter would require a very expressive and computationally more complex multi-modal language combining epistemic modalities and modalities for motivational attitudes (e.g., desires, preferences and intentions), such as the one presented by [40]. Rather, our aim is to propose a minimal language for cognitive planning that can be used in HMI applications involving one artificial planning agent (the persuader) and possibly many human agents (the persuadees), where minimality means that it can be automated using a SAT solver.

The examples studied in the paper focus on the interaction between one artificial agent and one human. To achieve our objective, we accept the following compromise. On the side of the human, we only model her explicit beliefs and motivational attitudes (desires and intentions). On the side of the artificial planning agent, we model its explicit and implicit beliefs. The latter cover explicit/implicit beliefs about the environment as well as about the human's explicit beliefs and motivational attitudes. This is a fair compromise since the human is by definition resource-bounded and has no deductively closed beliefs. On the contrary, the artificial agent has perfect deductive capabilities, exemplified by the notion of implicit belief, that can be exploited to reason about the human's epistemic and motivational attitudes and, consequently, to influence her behavior.

It is also important to note that, given the type of interaction we are interested in modeling (one persuading machine, one human to be persuaded), we only need to represent the machine's subjective theory about the human's motivational attitudes. We do not need to represent the machine's motivational attitudes. Under this assumption, we will be able to

(1) represent the human's desires and intentions through atomic formulas, and (2) encode the machine's theory about the human's motivational attitudes directly in our minimal epistemic language.

As far as we know, there was no previous attempt (1) to come up with a simple logic-based approach to epistemic planning, and more generally to cognitive planning, which could be implemented using a SAT solver, and (2) to combine it with belief revision in order to be exploited in a concrete HMI application, wherein the elements of the agent's plans are speech acts executed during a dialogue. In this work, we are going to fill this gap. We will identify a minimal epistemic logic language that can be used to endow a persuasive artificial agent with cognitive planning and belief revision capabilities and that can be implemented using a SAT solver. This is in a nutshell the main novel contribution of our work.

Our work has ethical implications that are not explored in the paper. We deal with the problem of endowing an artificial agent with persuasive capabilities. The risk associated with this kind of AI models and technologies is that they could be used for manipulative purposes and for obnoxious activities. To limit this risk, it is important to combine the logic-based model of cognitive planning with a self-regulative component which specifies those communicative actions that are legally permissible and ethically acceptable in a given situation. This is in order to guarantee that the artificial agent will refrain from behaving in an illegal or ethically deplorable way. We leave the combination of the cognitive planning model and the self-regulative module to future work.

1.1 Outline of the paper

The paper is organized as follows. Section 2 is devoted to the discussion of related work. In Sect. 3, we provide a birds-eye view of the general architecture of the artificial agent with special emphasis on the interaction between the cognitive planning and the belief revision module. In Sect. 4 we introduce the language of explicit and implicit belief for the specification of the cognitive planning problem. Given that the satisfiability problem for the full language is PSPACE-hard, we study an interesting NP fragment of it that can be used in the context of a real HMI application. Then, we present an extension of the language by the notion of belief base expansion which is necessary for representing the actions of the planning agent. The second part of the paper is devoted to specify the modules of the architecture and to describe their implementation in an artificial agent. In Sect. 5, we formulate the cognitive planning problem and study its complexity. We show that the cognitive planning problem formulated in our epistemic language is Σ_2^{P}-complete. Moreover, we present the belief revision module of the agent architecture. In Sect. 6 we instantiate the cognitive planning and belief revision module in a concrete example in which an artificial assistant has to help a human user to choose a sport to practice in her leisure time. To achieve its goal, the agent needs a model of the user's beliefs and desires. Thanks to this model, the agent will be able to plan a sequence of speech acts aimed at persuading the user that a certain sport is the ideal one for her and, consequently, at inducing the user to form the intention to practice it. We also present an implementation of the example using a brute force approach to cognitive planning: the artificial agent has to generate all plans of a given length in order to find one which will enable it to achieve its goal. In Sect. 7 we present an optimal QBF-based algorithm for cognitive planning, as an alternative to the brute force algorithm. We provide an experimental comparison of the two algorithms. In Sect. 8 we

conclude. To make the paper more readable, we have decided to put all proofs of the main technical results in an annex at the end of the paper.

This is the improved and extended version of a paper presented at the AAAI-21 conference [18]. The AAAI-21 paper did not include the following things that are included in the present paper: (1) detailed proofs of the technical results, (2) detailed comparison with state of the art on epistemic planning and planning models of dialogue, (3) the belief revision module and its integration with the cognitive planning module, (4) the hardness result for complexity of cognitive planning, (5) the QBF-based approach to cognitive planning and the experimental comparison with the brute force approach. Moreover, at the time of the AAAI-21 paper (6) neither the agent architecture nor the HMI scenario were implemented. Finally, in the AAAI-21 paper (7) the HMI scenario was just sketched and not elaborated in detail by including both direction of interaction between the conversational agent and the human.

2 Related work

2.1 Formal models of persuasion

Models of persuasion in AI are mostly based on argumentation. See [55] for a general introduction to the research in this area. Some of these models are built on Walton & Krabbe's notion of persuasion dialogue in which one party seeks to persuade another party to adopt a belief or point-of-view she does not currently hold [66]. There exist models based on abstract argumentation [1, 4, 5, 9] as well probabilistic models where the persuader's uncertainty about what the persuadee knows or believes is represented [28, 29]. There exist also models based on possibility theory in which a piece of information is represented as an argument which can be more or less accepted depending on the trustworthiness of the agent who proposes it [16]. Persuasion has also been formalized with the support of logical tools, e.g., by representing the support and the conclusion of an argument by sets of literals and an argument path as a sequence of arguments such that for each argument in the sequence its support is obtained through the conclusions of the preceding arguments [59], by combining abstract argumentation with dynamic epistemic logic (DEL) [56], epistemic logic with dynamic logic [10] and epistemic logic with a logic of agency [8]. There are also been attempts to build models of persuasive multimodal communication based on reinforcement learning (RL) [67]. Their idea is that in order to enhance persuasiveness of its communication strategy, an agent must learn the user's preferences and adapt to the user's personality. Weber et al. model this learning and adaptation process in the RL framework. Unlike ours, none of the previous approaches to persuasion is based on planning.

A planning theory of persuasion, which is the core contribution of our work, requires a formal language for representing the persuadee's beliefs, the target of the persuader's communicative action. From this perspective, our work is closely connected to recent research in the area of epistemic planning.

2.2 Parsimonious approaches to epistemic planning

The concept of epistemic planning, as a generalization of classical planning, was introduced by [6, 44]. The initial proposal was to use a standard logic of knowledge or belief together with a representation of actions in terms of event models of dynamic epistemic

logic (DEL) [65]. While the DEL framework is very expressive, it turned out that the existence of a solution becomes quickly undecidable even for very simple kinds of event models [2, 7, 34]. Kominis and Geffner [30] considered epistemic planning problems with very simple event models leading to a decidable fragment and that can be compiled into a classical planning problem. They distinguish three kinds of actions: physical actions modifying the world, public updates (DEL-like public announcements), and sensing actions by means of which an agent learns whether a formula is true. Other researchers investigated another source of complexity, namely that of standard epistemic logic. There, reasoning is strictly more complex than in classical logic: the satisfiability problem is at least in PSPACE [26].

Based on earlier work by [33] on efficient multi-agent epistemic reasoning later extended by [53] with the notion of consistent and introspective belief (modal logic systems KD^n and $KD45^n$), Muise et al. [49, 51] studied epistemic planning by enforcing syntactic restrictions on the agents' beliefs. In particular, they considered state descriptions in terms of conjunctions of epistemic literals: formulas that do not contain any conjunction or disjunction. Their framework has been applied to team formation in a multi-agent setting whose fundamental aspect is the persuasion of some potential team members to join the team [52]. The aim of the initiator of the team formation process is to compute a conditional plan (conditioned on the responses of the potential team members) which guarantees the formation of a cohesive team in the pursuit of a collective goal. This planning model of team formation is based on an earlier model by [20] on the formation of collective intentions through structured dialogues.

Cooper et al. [13, 14] considered another fragment: boolean combinations of 'knowing-whether' operators defined from the primitive concept of observability followed by propositional variables. They call the resulting logic EL-O, which stands for Epistemic Logic of Observability.

Both Muise et al.'s language and Cooper et al.'s language do not allow to represent beliefs about facts expressed in disjunctive form or by logical implication. This is a big limitation from the point of view of our work in which cognitive planning is used for modeling a dialogue between an artificial agent and a human. Indeed, in a typical dialogue setting the planning agent could have conceptual information as well as causal information about the physical world, about the human's cognitive state, and about the connection between the physical world and the human's cognitive state. This kind of information is naturally expressed by means of disjunction, implication or double implication. For instance, the planning agent may believe that a certain property or event should be considered 'ideal for the human' *if and only if* it satisfies all her desires. This is an example of belief about "conceptual" information, inasmuch as it specifies the subjective interpretation of a given term by the planning agent (i.e., the meaning that the planning agent assigns to the term 'ideal for the human'). Or, the planning agent may believe that *if* the human believes that a certain action prevents her from satisfying her desires *then* she has no reasonable justification to perform it and, consequently, she will not intend to do it. This is an example of belief about the causal connection between the human's beliefs and her intentions. Finally, the planning agent may believe that *if* the human feels tired *then* she will have the desire to rest or to perform a relaxing activity. This is an example of belief about the causal connection between the human's physiological/physical state and her cognitive state (her desires).

In our work, we prefer to simplify the epistemic language by bounding nesting of the epistemic modalities, rather than by disallowing beliefs about disjunction or implication. We think this is a better compromise between simplicity and expressiveness, at least in the context of our application of cognitive planning to persuasive human–machine communication. In the epistemic language we use for cognitive planning (see Sect. 4), we can

nest explicit belief modalities, but we cannot nest implicit belief modalities. On the contrary, there is no restriction on the propositional logic level: we can represent explicit and implicit beliefs about any kind of propositional fact expressed by boolean connectives, including disjunction and implication.

2.3 Planning models of dialogue

The idea of using planning languages and algorithms for modeling dialogue between artificial and/or human agents is not new. Here, we only discuss the most relevant models for us, namely those based either on logic or on standard encoding languages for classical planning such as PDDL or STRIPS. A systematic literature review for the research in this area is provided by [64].

Muise et al. [50] propose a planning model for goal-oriented dialogue systems based on FOND (Fully Observable Non-Deterministic), where dialogue plans generated by the FOND planner are contingent plans with conditional effects. They encode their model in the PDDL (Planning Domain Definition Language) framework [22]. Another related work to be mentioned is Shams et al.'s combination of classical STRIPS planning and argumentation applied to normative contexts in which conflicts between an agent's goals and norms can arise [61]. In such contexts, it becomes useful to explain the criteria for selecting a certain plan and considering it the best plan with respect to goal achievement and norm compliance. Shams et al. use argumentation for this explanatory purpose in the context of a dialogue game. Unlike our approach, Muise et al and Shams et al do not explicitly represent the cognitive states (e.g., beliefs, preferences, intentions) of the agents involved in the dialogue.

Kominis and Geffner's work [31] is the closest one to ours. They show that the minimalistic approach to epistemic planning they presented earlier [30] could be leveraged for modeling simple agent dialogues where agents collaborate by requesting or volunteering information in a goal-directed manner. They provide a translation of their framework into the STRIPS language extended with negation, conditional effects, and specific axioms handling the dynamics of the agents' epistemic states. An important difference between our approach and Kominis & Geffner's is our emphasis on the connection between planning and belief revision both at the formal and architectural level that, we believe, is necessary to fully understand and model a dialogue between an artificial agent and a human agent. As we will show in Sect. 3, in our approach cognitive planning and belief revision are two interrelated modules of the conversational agent's architecture, while Kominis & Geffner do not consider belief revision. Another crucial difference between our approach and theirs is our emphasis on the reduction of cognitive planning to SAT, while their target model is STRIPS and, more generally, classical planning.

3 General architecture

The general architecture of our system is detailed in Fig. 2.

3.1 Data structures

The artificial planning agent, that for simplicity we call the machine, is endowed with three kinds of data structure: its belief base, the goal to be achieved and the repertoire of speech

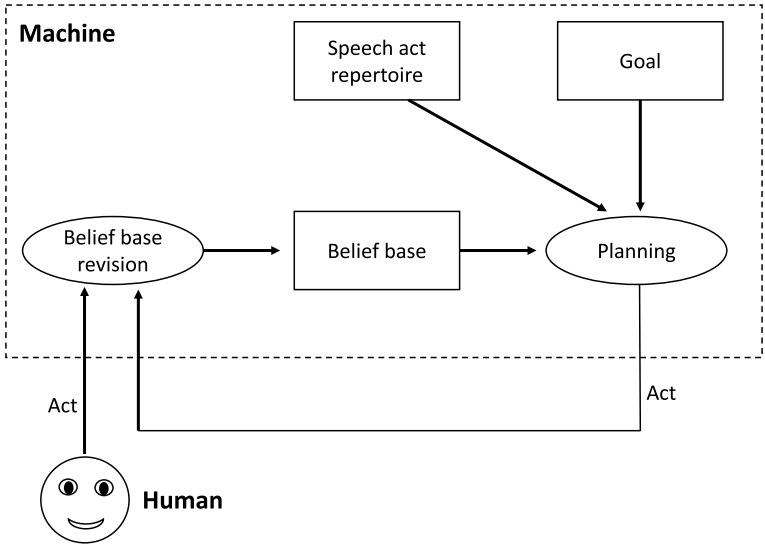


Fig. 2 General architecture

acts (or communicative actions) it can perform. Such speech acts are aimed at changing the human's cognitive state. The machine can have persuading goals, aimed at changing the human's beliefs, or influencing goals, aimed at inducing the human to form a certain intention or to behave in a certain way. The machine's belief base includes both information about the environment and information about the human's overall cognitive state and its way of functioning. In other words, the machine has a theory of the human's mind. The machine's belief base evolves during its dialogue with the human.

3.2 Exploratory and informative phase

The interaction between the machine and the human is structured in two phases the *exploratory* (or *inquiry*) phase and the *informative* phase.

In the exploratory phase the machine gathers information about the human's cognitive state. This includes information about the human's beliefs, desires and preferences. In this phase the human provides information to the machine and the machine expands or revises, when necessary, its belief base accordingly. Indeed, the information provided by the human can enrich the machine's belief base with new facts about the environment (objective facts) or about the human's cognitive state (mental facts) or make the machine's belief base inconsistent. In the latter case, the machine must revise its belief base after having incorporated the new information.

The informative phase is the core of the cognitive planning process. In this phase, the machine performs a sequence of assertions aimed at modifying the human's cognitive state (her beliefs and/or intentions). In our framework, the exploratory phase is propaedeutic to the informative phase. Indeed, for the machine to be able to lead the human to change her behavior, it must most often have information about the human's cognitive state. Such information may be acquired during the exploratory phase.

It is reasonable to suppose that if in the informative phase the machine cannot find a plan, it moves to the exploratory phase. In fact, the machine might not have sufficient

information at its disposal about the human's cognitive state to find a plan aimed at persuading or influencing her. In the exploratory phase, the machine tries to fill this knowledge gap.

The two phases can be kept separated or integrated at the planning level. In the "non-integrated" solution the exploratory phase consists in an information gathering protocol. After the protocol has terminated, the machine enters the informative phase. It is the proper planning phase whereby the machine searches for a sequence of assertions which guarantees the achievement of its persuasive or influencing goal.

In the "integrated" solution, not only the informative phase but also the exploratory phase is managed by the planning module. In the exploratory phase the machine includes in its plan not only assertions but also questions. In particular, it has to find a sequence of questions followed by a sequence of assertions such that, for some possible answer (or for all possible answers) by the human, the composition of the two sequences guarantees that the persuading or influencing goal will be achieved. It is reasonable to assume that the machine first tries to find a plan with only assertions. (Why asking questions to the human if what the machine knows about the human's cognitive state is already sufficient to persuade or influence her). However, in most cases, the machine has uncertainty and lacks information about the human's cognitive state so that it must ask questions to the human before trying to induce her attitude change through a sequence of assertions.

In this work, we will adopt the "non-integrated" solution for the sake of simplicity. Indeed, adding questions to plans would increase the complexity of the cognitive planning problem given their non-deterministic aspect (i.e., a question can seen as a speech acts with non-deterministic effects corresponding to the possible answers by the hearer). We will merely illustrate the basic functioning of the cognitive planning module in the informative phase and of the belief revision module in the exploratory phase. We leave the generalization of our approach to plans including questions for future work.

4 Logical framework

This section is devoted to present the logical framework for the formal specification of cognitive planning. We start by recalling the full language and the semantics presented by [38, 39]. The latter distinguishes explicit and from implicit belief. An agent's explicit belief is seen as a piece of information in the agent's belief base, while an implicit belief corresponds to a piece of information that is derivable from the agent's belief base (i.e., included in the deductive closure of the agent's belief base). It is interpreted over a semantics using belief bases. The central idea of this semantics is that an agent's epistemic indistinguishability relation should be computed from the agent's belief base by stipulating that *a state is considered possible by the agent if and only if it satisfies all information in the agent's belief base*.

The full multi-agent language being PSPACE-complete [39, Theorem 6], in this section we study an interesting novel NP fragment that can be used in HMI applications. In this fragment we can represent explicit beliefs of many agents and implicit beliefs of a single agent, with no nesting of implicit belief modalities. We conclude by presenting a dynamic extension of the static language by belief change operations. The latter are needed to represent the planning agent's communicative actions.

4.1 Full language and semantics

Our epistemic language distinguishes explicit belief (a fact in an agent's belief base) from implicit belief (a fact that is deducible from the agent's explicit beliefs). It is parameterized by a finite set of agents $Agt = \{1, \ldots, n\}$ and a countable infinite set of atomic propositions Atm noted $p, q, \ldots$

The language of our logic of explicit and implicit belief is defined in two steps.

First, the language $\mathcal{L}_0(Atm, Agt)$ is defined by the following grammar in BNF:

$$\alpha \quad ::= \quad p \mid \neg\alpha \mid \alpha \wedge \alpha \mid \bigtriangleup_i \alpha,$$

where p ranges over Atm and i ranges over Agt. $\mathcal{L}_0(Atm, Agt)$ is the language for representing agents' explicit beliefs. The formula $\bigtriangleup_i \alpha$ is read "i explicitly believes that α". Then, the language $\mathcal{L}(Atm, Agt)$ extends the language $\mathcal{L}_0(Atm, Agt)$ by modal operators of implicit belief and is defined by the following grammar:

$$\varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_i \varphi,$$

where α ranges over $\mathcal{L}_0(Atm, Agt)$ and i ranges over Agt. For notational convenience we write $\mathcal{L}_0$ instead of $\mathcal{L}_0(Atm, Agt)$ and $\mathcal{L}$ instead of $\mathcal{L}(Atm, Agt)$, when the context is unambiguous. The formula $\Box_i \varphi$ is read "i implicitly believes that φ" and its dual $\Diamond_i \varphi \stackrel{\text{def}}{=} \neg\Box_i\neg\varphi$ is read "φ is compatible (or consistent) with i's explicit beliefs". The other Boolean constructions $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined in the standard way.

The interpretation of language $\mathcal{L}$ exploits the notion of belief base. While the notions of possible state (or world) and epistemic alternative are primitive in the standard Kripke semantics for epistemic logic, they are defined from the primitive concept of belief base in our semantics. In particular, a state is a composite object including a description of both the agents' belief bases and the environment.[1]

Definition 1 (*State*) A state is a tuple $B = (B_1, \ldots, B_n, V)$ where: for every $i \in Agt$, $B_i \subseteq \mathcal{L}_0$ is agent i's belief base; $V \subseteq Atm$ is the actual environment. The set of all states is noted $\mathbf{S}$.

Note that an agent's belief base B_i can be infinite.[2] The sublanguage $\mathcal{L}_0(Atm, Agt)$ is interpreted w.r.t. states, as follows:

Definition 2 (*Satisfaction*) Let $B = (B_1, \ldots, B_n, V) \in \mathbf{S}$. Then:

$$\begin{aligned} B \vDash p &\iff p \in V, \\ B \vDash \neg\alpha &\iff B \nvDash \alpha, \\ B \vDash \alpha_1 \wedge \alpha_2 &\iff B \vDash \alpha_1 \text{ and } B \vDash \alpha_2, \\ B \vDash \bigtriangleup_i \alpha &\iff \alpha \in B_i. \end{aligned}$$

Observe in particular the set-theoretic interpretation of the explicit belief operator: agent i explicitly believes that α if and only if α is included in her belief base. There is no

[1] This is similar to the way states are modeled in the interpreted system semantics for multi-agent systems [21, 37].

[2] This is just a technical feature that is not required to represent the beliefs of a human or of an artificial agent.

constraint on the agents' belief bases. So, occasionally, during the exploratory phase, the belief base of the artificial agent may become inconsistent, but as we show below, consistency is then restored by belief revision (which is a process external to the logic).

A multi-agent belief model (MAB) is defined to be a state supplemented with a set of states, called *context*. The latter includes all states that are compatible with the common ground [63], i.e., the body of information that the agents commonly believe to be the case.

Definition 3 (*Multi-Agent Belief Model*) A multi-agent belief model (MAB) is a pair (B, Cxt), where $B \in \mathbf{S}$ and $Cxt \subseteq \mathbf{S}$. The class of all MABs is noted $\mathbf{M}$.

Note that we do not impose that $B \in Cxt$. When $Cxt = \mathbf{S}$ then (B, Cxt) is said to be *complete*, since $\mathbf{S}$ is conceivable as the complete (or universal) context which contains all possible states. We compute an agent's set of epistemic alternatives from the agent's belief base, as follows.

Definition 4 (*Epistemic alternatives*) Let $i \in Agt$. Then $\mathcal{R}_i$ is the binary relation on the set $\mathbf{S}$ such that, for all $B = (B_1, \ldots, B_n, V), B' = (B'_1, \ldots, B'_n, V') \in \mathbf{S}$:

$$B\mathcal{R}_i B' \quad \text{if and only if} \quad \forall \alpha \in B_i : B' \vDash \alpha.$$

$B\mathcal{R}_i B'$ means that B' is an epistemic alternative for agent i at B. So i's set of epistemic alternatives at B includes exactly those states that satisfy all i's explicit beliefs.

Definition 5 extends Definition 2 to the full language $\mathcal{L}$. Its formulas are interpreted with respect to MABs. We omit Boolean cases that are defined in the usual way.

Definition 5 (*Satisfaction*) Let $(B, Cxt) \in \mathbf{M}$. Then:

$$\begin{aligned} (B, Cxt) \vDash \alpha &\iff B \vDash \alpha, \\ (B, Cxt) \vDash \Box_i \varphi &\iff \forall B' \in Cxt, \text{ if } B\mathcal{R}_i B' \text{ then } (B', Cxt) \vDash \varphi. \end{aligned}$$

A formula $\varphi \in \mathcal{L}$ is valid in the class $\mathbf{M}$, noted $\vDash_{\mathbf{M}} \varphi$, if and only if $(B, Cxt) \vDash \varphi$ for every $(B, Cxt) \in \mathbf{M}$; it is satisfiable in $\mathbf{M}$ if and only if $\neg\varphi$ is not valid in $\mathbf{M}$.

Theorem 1 *Checking satisfiability of $\mathcal{L}(Atm, Agt)$ formulas in the class $\mathbf{M}$ is a PSPACE-hard problem.*

This theorem is a consequence of the fact that our logic contains the basic modal logic K whose satisfiability problem is PSPACE-complete [26].

4.2 NP-complete fragment

We study the following fragment of the language $\mathcal{L}$, called $\mathcal{L}_{\mathsf{Frag}}$:

$$\varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_{\mathfrak{m}} \alpha,$$

where α ranges over $\mathcal{L}_0$ and $\mathfrak{m}$ is a special agent in Agt called the 'machine'. In $\mathcal{L}_{\mathsf{Frag}}$, all agents have explicit beliefs but only agent $\mathfrak{m}$ has implicit beliefs, and moreover the latter are restricted to $\mathcal{L}_0$-formulas of type α. So there are no nested implicit beliefs for agent $\mathfrak{m}$. Agent $\mathfrak{m}$ is assumed to be the unique artificial agent in the system which is endowed with

$$\mathcal{L}_{\mathsf{Frag}} \overset{tr_1}{\dashrightarrow} \mathcal{L}_{\mathsf{Mod}} \overset{tr_2}{\dashrightarrow} \mathcal{L}_{\mathsf{Prop}}$$

Fig. 3 Summary of reduction process

unbounded reasoning and planning capabilities. The cognitive planning problem will be modeled from agent $\mathfrak{m}$'s perspective.

The following example is given to illustrate the language $\mathcal{L}_{\mathsf{Frag}}$ and its semantic interpretation.

Example 1 The scenario consists of agents $\mathfrak{m}$ and $\mathfrak{h}$. Agent $\mathfrak{h}$ is a person who applied for a loan to a bank, while agent $\mathfrak{m}$ is her virtual assistant. The loan application can be either accepted (i.e., a) or rejected (i.e., $\neg a$). Agent $\mathfrak{h}$ is informed about the result of her application through e-mail notification by the bank (i.e., n). Let $B = (B_{\mathfrak{m}}, B_{\mathfrak{h}}, V) \in \mathbf{S}$ and $(B, Cxt) \in \mathbf{M}$ such that:

$$\begin{aligned}
&B_{\mathfrak{m}} = \left\{n, (n \wedge a) \rightarrow \triangle_{\mathfrak{h}} a, (n \wedge \neg a) \rightarrow \triangle_{\mathfrak{h}} \neg a\right\},\\
&B_{\mathfrak{h}} = \{n, a\},\\
&V = \{n, a\},\\
&Cxt = \left\{B' = (B'_{\mathfrak{m}}, B'_{\mathfrak{h}}, V') \in \mathbf{S} : B' \vDash \neg(\triangle_{\mathfrak{h}} a \wedge \neg \triangle_{\mathfrak{h}} \neg a)\right\}.
\end{aligned}$$

According to the previous specification, agent $\mathfrak{m}$'s belief base includes the following facts only: notification has been sent out by the bank, and if notification has been sent out by the bank and agent $\mathfrak{h}$'s application is accepted (resp. rejected), she will explicitly believe that her application is accepted (resp. rejected). Moreover, agent $\mathfrak{h}$'s belief base includes the following facts only: notification has been sent out by the bank, and the application is accepted. Agent $\mathfrak{h}$'s explicit beliefs are correct since the two facts are objectively true. Finally, the agents' common ground only includes the information that agent $\mathfrak{h}$ cannot explicitly believe at the same time that her application is accepted and that her application is rejected. It is easy to verify that at model (B, Cxt), agent $\mathfrak{m}$ can infer that if agent $\mathfrak{h}$ explicitly believes that her application is accepted, then her application is indeed accepted, that is:

$$(B, Cxt) \vDash \square_{\mathfrak{m}} (\triangle_{\mathfrak{h}} a \rightarrow a).$$

In the rest of this section, we are going to provide a polysize reduction of the satisfiability problem of $\mathcal{L}_{\mathsf{Frag}}$ to SAT. The reduction consists of two steps which are summarized in Fig. 3.

As a first step, we define the following modal language $\mathcal{L}_{\mathsf{Mod}}$ into which the language $\mathcal{L}_{\mathsf{Frag}}$ will be translated[3]:

[3] See p. 24, just after Theorem 2, for the $\mathcal{L}_{\mathsf{Prop}}$ definition.

$$\begin{aligned}\omega &::= q \mid \neg\omega \mid \omega \wedge \omega,\\ \varphi &::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \blacksquare\omega,\end{aligned}$$

where q ranges over the following set of atomic formulas:

$$Atm^{+} = Atm \cup \{p_{\triangle_i \alpha} : i \in Agt \text{ and } \alpha \in \mathcal{L}_0(Atm, Agt)\}.$$

So $p_{\triangle_i \alpha}$ is nothing but a special atomic formula. In fact, in language $\mathcal{L}_{\mathsf{Mod}}$ a formula describing an agent's explicit belief is treated as an atomic formula.

We interpret the language $\mathcal{L}_{\mathsf{Mod}}$ w.r.t. a pair (M, w), called pointed Kripke model, where $M = (W, \Rightarrow, \pi)$, W is a non-empty set of worlds, $\Rightarrow \subseteq W \times W$ and $\pi : Atm^{+} \longrightarrow 2^{W}$.

Definition 6 The semantic interpretation for formulas in $\mathcal{L}_{\mathsf{Mod}}$ w.r.t. a pointed Kripke model (M, w) is as follows:

$$\begin{aligned}(M, w) \vDash q &\iff w \in \pi(q);\\ (M, w) \vDash \blacksquare\omega &\iff \forall v \in W, \text{ if } w \Rightarrow v \text{ then } (M, v) \vDash \omega.\end{aligned}$$

(Boolean cases are again omitted as they are defined in the usual way.)

The class of pointed Kripke models is noted $\mathbf{K}$. Satisfiability and validity of formulas in $\mathcal{L}_{\mathsf{Mod}}$ relative to the class $\mathbf{K}$ is defined in the usual way.

We are going to define next a translation of $\mathcal{L}_{\mathsf{Frag}}$-formulas into $\mathcal{L}_{\mathsf{Mod}}$-formulas. It has two components. A translation tr_0 of $\mathcal{L}_0$-formulas and a more general translation tr_1 which depends on it. We define $tr_0 : \mathcal{L}_0 \longrightarrow \mathcal{L}_{\mathsf{Mod}}$ first:

$$\begin{aligned}tr_0(p) &= p,\\ tr_0(\neg\alpha) &= \neg tr_0(\alpha),\\ tr_0(\alpha_1 \wedge \alpha_2) &= tr_0(\alpha_1) \wedge tr_0(\alpha_2),\\ tr_0(\triangle_i \alpha) &= p_{\triangle_i \alpha}.\end{aligned}$$

The translation $tr_1 : \mathcal{L}_{\mathsf{Frag}} \longrightarrow \mathcal{L}_{\mathsf{Mod}}$ is defined as follows:

$$\begin{aligned}tr_1(p) &= p,\\ tr_1(\neg\varphi) &= \neg tr_1(\varphi),\\ tr_1(\varphi_1 \wedge \varphi_2) &= tr_1(\varphi_1) \wedge tr_1(\varphi_2),\\ tr_1(\triangle_i \alpha) &= \begin{cases} p_{\triangle_{\mathfrak{m}} \alpha} \wedge \blacksquare tr_0(\alpha), & \text{if } i = \mathfrak{m},\\ p_{\triangle_i \alpha}, & \text{otherwise,}\end{cases}\\ tr_1(\square_{\mathfrak{m}} \beta) &= \blacksquare tr_0(\beta);\end{aligned}$$

It is easy to see that the translation tr_1 is polynomial. Indeed, except for case $tr_1(\triangle_i \alpha)$ at every step of the translation, the number of symbols does not increase. When we translate $\triangle_i \alpha$ there is no risk for exponential blow up since either the translation stops at the next step (case $i \neq \mathfrak{m}$) or it produces a conjunctive formula whose first conjunct is $p_{\triangle_{\mathfrak{m}} \alpha}$, which is not translated further, and whose second conjunct is translated at the next step using tr_0 (case $i = \mathfrak{m}$). It is is trivial to verify that the translation tr_0 is polynomial.

As the following theorem indicates, the translation tr_1 guarantees the transfer of satisfiability from model class **M** to model class **K**.

Theorem 2 *Let $\varphi \in \mathcal{L}_{\mathsf{Frag}}$. Then, φ is satisfiable in the class* **M** *if and only if $tr_1(\varphi)$ is satisfiable in the class* **K**.

(The proof is provided in "Appendix".)

As a last step, we provide a polysize reduction of $\mathcal{L}_{\mathsf{Mod}}$-satisfiability to SAT, where the underlying propositional logic language $\mathcal{L}_{\mathsf{Prop}}$ is built from the following set of atomic propositions:

$$Atm^{++} = \{q_x : q \in Atm^{+} \text{ and } x \in \mathbb{N}\} \cup \{r_{x,y} : x, y \in \mathbb{N}\}.$$

The set Atm^{++} includes two types of atomic propositions: one of the form q_x denoting the fact that q is true at world x and the other of the form $r_{x,y}$ denoting the fact that world x is related to world y.

Let $tr_2 : \mathcal{L}_{\mathsf{Mod}} \times \mathbb{N} \times \mathbb{N} \longrightarrow \mathcal{L}_{\mathsf{Prop}}$ be the following translation function:

$$\begin{aligned}
tr_2(q, x, y) &= q_x, \\
tr_2(\neg\varphi, x, y) &= \neg tr_2(\varphi, x, y), \\
tr_2(\varphi_1 \wedge \varphi_2, x, y) &= tr_2(\varphi_1, x, y) \wedge tr_2(\varphi_2, x, y), \\
tr_2(\blacksquare\omega, x, y) &= \bigwedge_{0 \le z \le y} \big(r_{x,z} \rightarrow tr_2(\omega, z, y)\big).
\end{aligned}$$

Translation tr_2 is similar to the translation of modal logic S5 into propositional logic given by [11] and, more generally, to the standard translation of modal logic into FOL in which accessibility relations are encoded by special predicates. In our framework, a third argument has been added. It specifies the maximum size of the model we restricted to, ensuring that the translation is not going to be of exponential size.

The size of an $\mathcal{L}_{\mathsf{Mod}}$ formula, $size(\varphi)$, is defined by:

$$\begin{aligned}
size(p) &= 1, \\
size(\varphi_1 \wedge \varphi_2) &= size(\varphi_1) + size(\varphi_2) + 1, \\
size(\neg\varphi) &= size(\varphi) + 1, \\
size(\blacksquare\omega) &= size(\omega) + 1.
\end{aligned}$$

Note that the size of $tr_2\big(\varphi, 0, size(\varphi)\big)$ is polynomial in the size of φ.

The following theorem is analogous to the result for the standard translation of modal logic to FOL. It is proved by a straightforward adaptation of [32, Lemma 6.1] about polysize-model property for modal logic S5 to our case. For this reason, we do not need to give the detailed proof.

Theorem 3 *Let $\varphi \in \mathcal{L}_{\mathsf{Mod}}$. Then, φ is satisfiable in the class* **K** *if and only if $tr_2\big(\varphi, 0, size(\varphi)\big)$ is satisfiable in propositional logic.*

The size of $tr_2\big(\varphi, 0, size(\varphi)\big)$ is polynomial in $size(\varphi)$. The only case to pay attention is the translation $tr_2\big(\blacksquare\omega, 0, size(\varphi)\big)$ for some subformula $\blacksquare\omega$ during the translation process. We generate a formula with y conjuncts. For each conjunct we only need to translate the

propositional formula ω. It is trivial to verify that the translation of a propositional formula is linear in the size of the formula to be translated. This guarantees that there is no exponential blow up when translating a subformula $\blacksquare\omega$ during the translation process.

Thanks to Theorems 2 and 3 we state the following complexity result.

Theorem 4 *Checking satisfiability of formulas in* $\mathcal{L}_{\mathsf{Frag}}$ *in the class* $\mathbf{M}$ *is an NP-complete problem.*

(The proof is provided in "Appendix")

4.3 Dynamic extension

In this section, we extend the language $\mathcal{L}_{\mathsf{Frag}}$ by belief expansion operations. Specifically, we introduce the following language $\mathcal{L}^{+}_{\mathsf{Frag}}$:

$$\varphi \quad ::= \quad \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_{\mathfrak{m}}\alpha \mid [+_i\alpha]\varphi,$$

where α ranges over the language $\mathcal{L}_0$ and i ranges over Agt. The formula $[+_i\alpha]\varphi$ is read "φ holds after agent i has privately expanded her belief base with α". Events of type $+_i\alpha$ are generically called informative actions.

The notion of belief expansion we consider is rather simple: it consists in adding a single piece of information (one formula in $\mathcal{L}_0$) to the belief base of an agent i. It is worth noting that the dynamic operators $[+_i\alpha]$ are commutative. Therefore, an expansion with multiple information at a time can be simulated by a sequence of expansion operations with one formula. Specifically, $\left[+_i \{\alpha_1, \ldots, \alpha_k\}\right]$ can be defined as an abbreviation of $\left[+_i \sigma(\alpha_1)\right] \ldots \left[+_i \sigma(\alpha_k)\right]$, where σ is any permutation of the set $\{\alpha_1, \ldots, \alpha_k\}$.

Our extension with belief base expansion operators has the following semantics relative to a MAB:

Definition 7 (*Satisfaction relation, cont.*) Let $B = (B_1, \ldots, B_n, V) \in \mathbf{S}$ and let $(B, Cxt) \in \mathbf{M}$. Then:

$$(B, Cxt) \vDash [+_i\alpha]\varphi \quad \Longleftrightarrow \quad (B^{+_i\alpha}, Cxt) \vDash \varphi$$

with $V^{+_i\alpha} = V$, $B_i^{+_i\alpha} = B_i \cup \{\alpha\}$ and $B_j^{+_i\alpha} = B_j$ for all $j \neq i$.

Intuitively speaking, the private expansion of i's belief base by α simply consists in agent i adding the information that α to her belief base, while all other agents keep their belief bases unchanged.

Let us go back to Example 1 to illustrate the update semantics for belief expansion.

Example 2 At model (B, Cxt), after being informed that agent $\mathfrak{h}$ believes that her application is accepted, agent $\mathfrak{m}$ can infer that $\mathfrak{h}$'s application is indeed accepted, that is:

$$(B, Cxt) \vDash [+_{\mathfrak{m}} \triangle_{\mathfrak{h}} a] \Box_{\mathfrak{m}} a.$$

Before illustrating the computational properties of our dynamic extension, we would like to emphasize some of its features. In this work we do not consider dynamic operators for belief contraction or revision that would allow an agent to retract information

from its belief base. We see belief base revision as an independent module of the agent architecture (see Sect. 5.2) and do not represent it at the planning level. Therefore, we can avoid altogether the unnecessary complication of including dynamic operators for revision in the logical language. More details about the extensions of our epistemic language with belief base revision operators can be found in [39, 42]. Secondly, we take a weak logic of epistemic attitudes for the human since we only model her explicit beliefs. In our model, the human may explicitly believe that $p \vee q$ (resp. $p \to q$) and learn that $\neg p$ (resp. p), without explicitly believing that q as a consequence. In formal terms, $\triangle_{\mathfrak{h}}(p \vee q) \wedge [+_{\mathfrak{h}} \neg p] \triangle_{\mathfrak{h}} q$ and $\triangle_{\mathfrak{h}}(p \to q) \wedge [+_{\mathfrak{h}} p] \triangle_{\mathfrak{h}} q$ are not valid in our setting. The reason why we do not take the previous formulas to be valid is that we do not want to make any assumption on the deductive capabilities of the human that, as clearly observed in cognitive psychology, are imperfect and context-dependent [15]. Consequently, the machine might only have in its belief base partial information about the human's deductive capabilities. However, adding to the formal semantics local deductive principles like the previous one would only require a minor change. For example, it would be sufficient to redefine the updated model $B_{\mathfrak{h}}^{+_{\mathfrak{h}}\alpha}$ as follows to guarantee that the human infers all direct consequences of what she learns:

$$B_{\mathfrak{h}}^{+_{\mathfrak{h}}\alpha} = B_{\mathfrak{h}} \cup \{\alpha\} \cup \{\alpha' : \alpha \to \alpha' \in B_{\mathfrak{h}}\}.$$

Under this variant of the update semantics, the following equivalence would become valid:

$$[+_{\mathfrak{h}}\alpha] \triangle_{\mathfrak{h}} \alpha' \leftrightarrow \begin{cases} \top, & \text{if } \alpha' = \alpha, \\ \triangle_{\mathfrak{h}}(\alpha \to \alpha'), & \text{otherwise.} \end{cases}$$

It is easy to show that the complexity results we provide are invariant under this slight modification of the semantics. But we do not investigate it in the paper.

Proposition 1 *The following equivalences are valid in the class* $\mathbf{M}$:

$$[+_i\alpha]\alpha' \leftrightarrow \begin{cases} \top, & \text{if } \alpha' = \triangle_i\alpha, \\ \alpha', & \text{otherwise;} \end{cases}$$

$$[+_i\alpha]\neg\varphi \leftrightarrow \neg[+_i\alpha]\varphi;$$

$$[+_i\alpha](\varphi_1 \wedge \varphi_2) \leftrightarrow [+_i\alpha]\varphi_1 \wedge [+_i\alpha]\varphi_2;$$

$$[+_i\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow \begin{cases} \Box_{\mathfrak{m}}(\alpha \to \alpha'), & \text{if } i = \mathfrak{m}, \\ \Box_{\mathfrak{m}}\alpha', & \text{otherwise;} \end{cases}$$

(The proof is provided in "Appendix")

Thanks to the equivalences of Proposition 1 we can define the following reduction *red* transforming every $\mathcal{L}^{+}_{\mathsf{Frag}}$ formula φ into an equivalent $\mathcal{L}_{\mathsf{Frag}}$ formula $red(\varphi)$:

$$
\begin{aligned}
red(p) &= p,\\
red(\bigtriangleup_i \alpha) &= \bigtriangleup_i \alpha,\\
red(\neg\varphi) &= \neg red(\varphi),\\
red(\varphi_1 \wedge \varphi_2) &= red(\varphi_1) \wedge red(\varphi_2),\\
red(\Box_{\mathfrak{m}} \alpha) &= \Box_{\mathfrak{m}} red(\alpha),\\
red([+_i\alpha]\alpha') &= \begin{cases} \top, & \text{if } \alpha' = \bigtriangleup_i \alpha,\\ red(\alpha'), & \text{otherwise;}\end{cases}\\
red([+_i\alpha]\neg\varphi) &= red(\neg[+_i\alpha]\varphi),\\
red\big([+_i\alpha](\varphi_1 \wedge \varphi_2)\big) &= red([+_i\alpha]\varphi_1 \wedge [+_i\alpha]\varphi_2),\\
red([+_i\alpha]\Box_{\mathfrak{m}}\alpha') &= \begin{cases} red\big(\Box_{\mathfrak{m}}(\alpha \rightarrow \alpha')\big), & \text{if } i = \mathfrak{m},\\ red(\Box_{\mathfrak{m}}\alpha') & \text{otherwise;}\end{cases}
\end{aligned}
$$

$$
red([+_i\alpha_1][+_j\alpha_2]\varphi) = red\big([+_i\alpha_1]red([+_j\alpha_2]\varphi)\big).
$$

Proposition 2 *Let $\varphi \in \mathcal{L}^{+}_{\mathsf{Frag}}$. Then, $\varphi \leftrightarrow red(\varphi)$ is valid in the class* $\mathbf{M}$*, and $red(\varphi) \in \mathcal{L}_{\mathsf{Frag}}$.*

(The proof is provided in "Appendix")

The following proposition is a direct corollary of the previous one. The right-to-left direction can be proved by contraposition: suppose φ is valid in $\mathbf{M}$; thus, thanks to Proposition 2, $red(\varphi)$ is valid in $\mathbf{M}$ too. The left-to-right direction is proved in an analogous way.

Proposition 3 *Let $\varphi \in \mathcal{L}^{+}_{\mathsf{Frag}}$. Then, φ is satisfiable in the class* $\mathbf{M}$ *iff $red(\varphi)$ is satisfiable too.*

The following theorem is a consequence of Theorem 4, Proposition 2 and the fact that the size of $red(\varphi)$ is polynomial in the size of φ.

Theorem 5 *Checking satisfiability of formulas in $\mathcal{L}^{+}_{\mathsf{Frag}}$ in the class* $\mathbf{M}$ *is an NP-complete problem.*

Before concluding this section, we define the concept of logical consequence for the language $\mathcal{L}^{+}_{\mathsf{Frag}}$ which will be used in the formulation of the cognitive planning problem.

Let Σ be a finite subset of $\mathcal{L}_0$ and let $\varphi \in \mathcal{L}^{+}_{\mathsf{Frag}}$. We define $\mathbf{S}(\Sigma) = \{B \in \mathbf{S} : \forall \alpha \in \Sigma, B \models \alpha\}$. We say that φ is a logical consequence of Σ in the class $\mathbf{M}$, noted $\Sigma \models_{\mathbf{M}} \varphi$, if and only if, for every $(B, Cxt) \in \mathbf{M}$ such that $Cxt \subseteq \mathbf{S}(\Sigma)$ we have $(B, Cxt) \models \varphi$. We say that φ is Σ-satisfiable in the class $\mathbf{M}$ if and only if, $\neg\varphi$ is not a logical consequence of Σ in $\mathbf{M}$. Clearly, φ is valid if and only if φ is a logical consequence of $\emptyset$, and φ is satisfiable if and only if φ is $\emptyset$-satisfiable.

As the following deduction theorem indicates, the logical consequence problem with a finite set of premises can be reduced to the validity problem and, consequently, to the satisfiability problem.

Proposition 4 *Let* $\varphi \in \mathcal{L}^{+}_{\mathsf{Frag}}$ *and let* $\Sigma \subset \mathcal{L}_0$ *be finite. Then,* $\Sigma \models_{\mathbf{M}} \varphi$ *if and only if* $\models_{\mathbf{M}} \left(\bigwedge_{\alpha \in \Sigma} \Box_{\mathfrak{m}} \alpha \right) \rightarrow \varphi$.

(The proof is provided in "Appendix")

The previous result is important since in Sect. 5 the cognitive planning problem will be formulated using the left side of the equivalence, with Σ representing agent $\mathfrak{m}$'s model of agent $\mathfrak{h}$'s mind. Thanks to Proposition 4, we will be able to relate cognitive planning to validity and satisfiability checking.

5 From cognitive planning to belief revision

In this section, we provide a formal specification of the two modules of the agent architecture, the cognitive planning and the belief revision module. We mainly concentrate on the complexity of cognitive planning formulated in the epistemic logic framework we presented in Sect. 4.

5.1 Cognitive planning problem and its complexity

We specify the cognitive planning problem in a two-agent version of the language $\mathcal{L}^{+}_{\mathsf{Frag}}$ presented in Sect. 4.3. In particular, we consider a finite set of agents $Agt = \{\mathfrak{h}, \mathfrak{m}\}$, where agent $\mathfrak{m}$ is assumed to be an artificial agent which interacts with the resource-bounded human agent $\mathfrak{h}$. The cognitive planning problem consists in finding an executable sequence of speech acts such that if agent $\mathfrak{m}$ performs it, at the end of its execution it will believe that its goal α_G is achieved. In other words, the solution of a cognitive planning problem is an executable sequence of speech acts by agent $\mathfrak{m}$ which guarantees the achievement of agent $\mathfrak{m}$'s goal α_G.

5.1.1 Speech acts

Let $Act_{\mathfrak{m}} = \{+_{\mathfrak{m}}\alpha : \alpha \in \mathcal{L}_0\}$ be agent $\mathfrak{m}$'s set of belief expansion operations (or informative actions) and let elements of $Act_{\mathfrak{m}}$ be noted $\epsilon, \epsilon', \ldots$ Speech acts of type 'assertion' are formalized as follows:

$$assert(\mathfrak{m},\mathfrak{h},\alpha) \stackrel{\text{def}}{=} +_{\mathfrak{m}} \triangle_{\mathfrak{h}} \triangle_{\mathfrak{m}} \alpha.$$

The event $assert(\mathfrak{m},\mathfrak{h},\alpha)$ captures the speech act "agent $\mathfrak{m}$ asserts to agent $\mathfrak{h}$ that α". The latter is assumed to coincide with the perlocutionary effect [60, Sect. 6.2] of the speaker learning that the hearer has learnt that the speaker believes that α.[4] We distinguish simple *assertions* from actions of *convincing*:

[4] We implicitly assume that, by default, $\mathfrak{m}$ believes that $\mathfrak{h}$ trusts its sincerity, so that $\mathfrak{h}$ will believe that $\mathfrak{m}$ believes what it says.

$$convince(\mathfrak{m},\mathfrak{h},\alpha) \stackrel{\text{def}}{=} +_{\mathfrak{m}} \triangle_{\mathfrak{h}}\alpha.$$

The event $convince(\mathfrak{m},\mathfrak{h},\alpha)$ captures the action "agent $\mathfrak{m}$ convinces agent $\mathfrak{h}$ that α".[5] We have $assert(\mathfrak{m},\mathfrak{h},\alpha) = convince(\mathfrak{m},\mathfrak{h}, \triangle_{\mathfrak{m}} \alpha)$. We assume 'to assert' and 'to convince' correspond to different utterances. While 'to assert' corresponds to the speaker's utterances of the form "I think that α is true!" and "In my opinion, α is true!", 'to convince' corresponds to the speaker's utterances of the form "α is true!" and "it is the case that α!".

The previous abbreviations and, more generally, the idea of describing speech acts of a communicative plan performed by agent $\mathfrak{m}$ with $\mathfrak{m}$'s private belief expansion operations is justified by the fact that we model cognitive planning from the perspective of the planning agent $\mathfrak{m}$. Therefore, we only need to represent the effects of actions on agent $\mathfrak{m}$'s beliefs. This does not mean that we assume that a speech act of agent $\mathfrak{m}$ does not change agent $\mathfrak{h}$'s beliefs. We simply do not model the effects of a speech act on $\mathfrak{h}$'s beliefs. In order to model the effects of the speech act on both sides of interaction, we would need to define the act of asserting as the sequence of two belief expansion operations $+_{\mathfrak{m}} \triangle_{\mathfrak{h}} \triangle_{\mathfrak{m}}\alpha; +_{\mathfrak{h}} \triangle_{\mathfrak{m}}\alpha$, and the act of convincing as the sequence of two belief expansion operations $+_{\mathfrak{m}} \triangle_{\mathfrak{h}} \alpha; +_{\mathfrak{h}} \alpha$. But this is out of the scope of our model.

5.1.2 Executability preconditions

We assume informative actions in $Act_{\mathfrak{m}}$ have executability preconditions that are specified by the following function: $\mathcal{P} : Act_{\mathfrak{m}} \longrightarrow \mathcal{L}_{\mathsf{Frag}}$. We assume that an informative action ϵ *can* take place if its executability precondition $\mathcal{P}(\epsilon)$ holds.

We use the executability precondition function $\mathcal{P}$ to define the following operator of possible occurrence of an event:

$$\langle\langle\epsilon\rangle\rangle\varphi \stackrel{\text{def}}{=} \mathcal{P}(\epsilon) \wedge [\epsilon]\varphi,$$

with $\epsilon \in Act_{\mathfrak{m}}$. The abbreviation $\langle\langle\epsilon\rangle\rangle\varphi$ has to be read "the informative action ϵ can take place and φ necessarily holds after its occurrence".

5.1.3 Formal specification

We conclude this section with a formal specification of the cognitive planning problem.

Definition 8 (*Cognitive planning problem*) A cognitive planning problem is a tuple $\langle\Sigma, Op, \alpha_G\rangle$ where:

- $\Sigma \subset \mathcal{L}_0$ is a finite set of agent $\mathfrak{m}$'s available information,
- $Op \subset Act_{\mathfrak{m}}$ is a finite set of agent $\mathfrak{m}$'s informative actions,
- $\alpha_G \in \mathcal{L}_0$ is agent $\mathfrak{m}$'s goal.

Informally speaking, a cognitive planning problem is the problem of finding an executable sequence of informative actions which guarantees that, at the end of the sequence, the planning

[5] As for convincing we assume that, by default, $\mathfrak{m}$ believes that $\mathfrak{h}$ trusts its competence, so that $\mathfrak{h}$ will believe what $\mathfrak{m}$ says. For a logical analysis of trust in sincerity and competence in communication, see [19, 41].

agent $\mathfrak{m}$ believes that its goal α_G is achieved. Typically, α_G is a persuading or influencing goal, i.e., the goal of affecting agent's $\mathfrak{h}$ cognitive state (including her beliefs and intentions) in a certain way. A solution plan to a cognitive planning problem $\langle \Sigma, Op, \alpha_G \rangle$ is a sequence of informative actions $\epsilon_1, \ldots, \epsilon_k$ from Op for some k such that $\Sigma \models_{\mathbf{M}} \langle\langle \epsilon_1 \rangle\rangle \ldots \langle\langle \epsilon_k \rangle\rangle \Box_{\mathfrak{m}} \alpha_G$.

5.1.4 Complexity results

For simplifying our notation, we introduce the following notation:

$$B_i^{+_i(\alpha_1,\ldots,\alpha_k,\alpha_{k+1})} = \left(B_i^{+_i(\alpha_1,\ldots,\alpha_k)}\right)^{+_i\alpha_{k+1}}$$

defined inductively for sequences of formulas.

As the following proposition highlights, checking existence of a solution for a cognitive planning problem has the poly-size property. Indeed, it can be easily seen that if an operator has been executed in a plan, another future occurrence of the same operator will not change the planning state due to the monotonicity of private belief expansion operations, that is,

$$B_i^{+_i(\alpha_1,\ldots,\alpha_j,\ldots,\alpha_h,\alpha_j)} = B_i^{+_i(\alpha_1,\ldots,\alpha_j,\ldots,\alpha_h)}.$$

There is a parallel between the previous property and the result presented in [44, Lemma 5] showing that action models with propositional preconditions commute making DEL planning tractable.

Proposition 5 *A cognitive planning problem $\langle \Sigma, Op, \alpha_G \rangle$ has a solution plan if and only if it has a poly-size solution plan $\epsilon_1, \ldots, \epsilon_k$ with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$.*

The previous proposition is crucial for proving the following complexity upper bound.

Theorem 6 *Checking plan existence for a cognitive planning problem is in Σ_2^{P}.*

(The proof is provided in "Appendix")

The following theorem provides a complexity lower bound for the cognitive planning problem.

Theorem 7 *Checking plan existence for a cognitive planning problem is Σ_2^{P}-hard.*

(The proof is provided in "Appendix")

5.2 Belief revision module

In this section, we describe the belief revision module of the architecture we sketched in Sect. 3. As we emphasized above, such a module is necessary for updating the machine's belief base during the exploratory phase of the interaction.

Let $\mathcal{L}_{\mathsf{PROP}}$ be the propositional language built from the following set of atomic formulas:

$$Atm^+ = Atm \cup \{p_{\triangle_i\alpha} : \triangle_i\alpha \in \mathcal{L}_0\}.$$

Moreover, let tr_{PROP} be the following translation from the language $\mathcal{L}_0$ defined in Sect. 4 to $\mathcal{L}_{\mathsf{PROP}}$:

$$\begin{aligned} tr_{\mathsf{PROP}}(p) &= p, \\ tr_{\mathsf{PROP}}(\neg\alpha) &= \neg tr_{\mathsf{PROP}}(\alpha), \\ tr_{\mathsf{PROP}}(\alpha_1 \wedge \alpha_2) &= tr_{\mathsf{PROP}}(\alpha_1) \wedge tr_{\mathsf{PROP}}(\alpha_2), \\ tr_{\mathsf{PROP}}(\triangle_i \alpha) &= p_{\triangle_i \alpha}. \end{aligned}$$

For each finite $X \subseteq \mathcal{L}_0$, we define $tr_{\mathsf{PROP}}(X) = \{tr_{\mathsf{PROP}}(\alpha) : \alpha \in X\}$. Moreover, we say that X is propositionally consistent if and only if $\bot \notin Cn\big(tr_{\mathsf{PROP}}(X)\big)$, where Cn is the classical deductive closure operator over the propositional language $\mathcal{L}_{\mathsf{PROP}}$. Clearly, the latter is equivalent to saying that $\bigwedge_{\alpha \in X} tr_{\mathsf{PROP}}(\alpha)$ is satisfiable in propositional logic.

Let $\Sigma_{core}, \Sigma_{mut} \subseteq \mathcal{L}_0$ denote, respectively, the core (or, immutable) information in agent $\mathfrak{m}$'s belief base and the volatile (or, mutable) information in agent $\mathfrak{m}$'s belief base. Agent $\mathfrak{m}$'s core beliefs are stable and do not change under belief revision. On the contrary, volatile beliefs can change due to a belief revision operation. Moreover, let $\Sigma_{input} \subseteq \mathcal{L}_0$ be agent $\mathfrak{m}$'s input information set. We define $\Sigma_{base} = \Sigma_{core} \cup \Sigma_{mut}$. The set Σ_{base} is nothing but the first parameter of the cognitive planning problem we specified in Definition 8, namely, agent $\mathfrak{m}$'s available information. This information changes through interaction with the other agent(s). This dynamic component of the agent architecture is handled by the belief revision module.

In particular, the revision of $(\Sigma_{core}, \Sigma_{mut})$ by input Σ_{input}, noted $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$, is formally defined as follows:

1. if $\Sigma_{core} \cup \Sigma_{input}$ is not propositionally consistent then $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) = (\Sigma_{core}, \Sigma_{mut})$,
2. otherwise, $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) = (\Sigma'_{core}, \Sigma'_{mut})$, with $\Sigma'_{core} = \Sigma_{core}$ and

$$\Sigma'_{mut} = \bigcap_{X \in MCS(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})} X,$$

where $X \in MCS(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$ if and only if:

- $X \subseteq \Sigma_{mut} \cup \Sigma_{input}$,
- $\Sigma_{input} \subseteq X$,
- $X \cup \Sigma_{core}$ is propositionally consistent, and
- there is no $X' \subseteq \Sigma_{mut} \cup \Sigma_{input}$ such that $X \subset X'$ and $X' \cup \Sigma_{core}$ is propositionally consistent.

The revision function Rev has the following effects on agent $\mathfrak{m}$'s beliefs: (1) the core belief base is not modified, while (2) the input Σ_{input} is added to the mutable belief base only if it is consistent with the core beliefs. If the latter is the case, then the updated mutable belief base is equal to the intersection of the subsets of the mutable belief base which

are maximally consistent with respect to the core belief base and which include the input Σ_{input}.[6] This guarantees that belief revision satisfies minimal change. The function *Rev* is a screened revision operator as defined by [45]. The latter was recently generalized to the multi-agent case by [42].

Let $Rev(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) = (\Sigma'_{core}, \Sigma'_{mut})$. For notational convenience, we write $Rev^{core}(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$ to denote Σ'_{core} and $Rev^{mut}(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$ to denote Σ'_{mut}. Note that, if Σ_{base} is propositionally consistent, then $Rev^{core}(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input}) \cup Rev^{mut}(\Sigma_{core}, \Sigma_{mut}, \Sigma_{input})$ is propositionally consistent too.

The belief revision machinery described above is fundamental for completing the interaction cycle between the machine and the human. In the application we will present in Sect. 6, agent $\mathfrak{m}$'s core belief base includes general principles about agent $\mathfrak{h}$'s mind that are assumed to be immutable during the interaction, e.g., the fact that agent $\mathfrak{h}$ has at least one desire or the fact that something is ideal for agent $\mathfrak{h}$ if it satisfies all her desires. Agent $\mathfrak{m}$'s mutable belief base will be constantly updated during the interaction in the light of the information provided by agent $\mathfrak{h}$. For example, agent $\mathfrak{h}$ can provide to agent $\mathfrak{m}$ information about her desires and $\mathfrak{h}$ will update its mutable belief base accordingly. If necessary $\mathfrak{m}$ will have to revise its mutable belief base, e.g., if $\mathfrak{h}$ tells to $\mathfrak{m}$ that her desires have changed.

6 Application: an artificial assistant

In this section, we present an example illustrating the interrelationship between the cognitive planning module and the belief revision module of the architecture we sketched in Sect. 3. The two modules were formally defined in Sects. 5.1 and 5.2. The example explores both directions of the interaction between agent $\mathfrak{h}$ and agent $\mathfrak{m}$.

6.1 Preliminary notions

We consider a HMI scenario in which agent $\mathfrak{m}$ is the artificial assistant of the human agent $\mathfrak{h}$. Agent $\mathfrak{h}$ has to choose a sport to practice since her doctor recommended her to do a regular physical activity to be in good health. Agent $\mathfrak{m}$'s aim is to help agent $\mathfrak{h}$ to make the right choice, given her actual beliefs and desires. The finite set of sport activities from which $\mathfrak{h}$ can choose is noted *Opt*. Elements of *Opt* are noted $o, o', \ldots$ Each option in *Opt* is identified with a finite set of variables *Var*. Each variable x in *Var* takes a value from its corresponding finite set of values Val_x.

[6] Note that the revision function *Rev* does not expand agent $\mathfrak{m}$'s core belief set Σ_{core} with the input information set Σ_{input}. It would be interesting to introduce a function $f_{\mathsf{appr}} : \mathcal{L}_0 \longrightarrow \{0, 1\}$ which specifies for every formula α in $\mathcal{L}_0$ whether the information α is completely apprehensible by agent $\mathfrak{m}$ (i.e., $f_{\mathsf{appr}}(\alpha) = 1$) or not (i.e., $f_{\mathsf{appr}}(\alpha) = 0$). Specifically, $f_{\mathsf{appr}}(\alpha) = 1$ means that if agent $\mathfrak{m}$ learns that α is true then, as a consequence, it will firmly believe that α is true thereby adding α not only to its set of mutable beliefs but also to its set of core beliefs. The function f_{appr} would allow us to define a variant of belief revision according to which if $\Sigma_{core} \cup \Sigma_{input}$ is propositionally consistent, then the core belief set Σ_{core} is expanded by all formulas α in Σ_{input} such that $f_{\mathsf{appr}}(\alpha) = 1$, that is, $\Sigma'_{core} = \Sigma_{core} \cup \{\alpha \in \Sigma_{input} : f_{\mathsf{appr}}(\alpha) = 1\}$.

In this example, we suppose that *Opt* is composed of the following eight elements: swimming (sw), running (ru), horse riding (hr), tennis (te), soccer (so), yoga (yo), diving (di) and squash (sq). Moreover, there are exactly six variables in *Var* which are used to classify the available options: environment (**env**), location (**loc**), sociality (**soc**), cost (**cost**), dangerousness (**dan**) and intensity (**intens**). The set of values for the variables are:

$$\begin{aligned}
Val_{\mathbf{env}} &= \{land, water\},\\
Val_{\mathbf{loc}} &= \{indoor, outdoor, mixed\},\\
Val_{\mathbf{soc}} &= \{single, team, mixed\},\\
Val_{\mathbf{cost}} &= \{low, med, high\},\\
Val_{\mathbf{dan}} &= \{low, med, high\},\\
Val_{\mathbf{intens}} &= \{low, med, high\}.
\end{aligned}$$

The set of assignments for variable x is defined as follows:

$$Assign_x = \{x \mapsto v : v \in Val_x\}.$$

The set of variable assignments is

$$Assign = \bigcup_{x \in Var} Assign_x.$$

Elements of *Assign* are noted $a, a', \ldots$

We assume that the content of an *atomic* desire is a variable assignment or its negation. That is, agent $\mathfrak{h}$'s atomic desire can be any element from the following set:

$$Des_0 = Assign \cup \{\sim a : a \in Assign\}.$$

Elements of Des_0 are noted $d, d', \ldots$. For example, the fact that $\mathfrak{h}$ has $\mathbf{loc} \mapsto indoor$ as a desire means that $\mathfrak{h}$ would like to practice an indoor activity, while if $\mathfrak{h}$'s desire is $\sim \mathbf{cost} \mapsto high$, then $\mathfrak{h}$ would like to practice an activity whose cost is not high. Agent $\mathfrak{h}$'s desires are either atomic desires or conditional desires. That is, $\mathfrak{h}$'s desire can be any element from the following set:

$$Des = Des_0 \cup \left\{[d_1, \ldots, d_k] \rightsquigarrow d : d_1, \ldots, d_k, d \in Des_0\right\}.$$

Elements of *Des* are noted $\gamma, \gamma', \ldots$ For example, if agent $\mathfrak{h}$ has $[\mathbf{cost} \mapsto high] \rightsquigarrow \mathbf{dan} \mapsto low$ as a desire, then she would like to practice a sport whose dangerousness level is low, if its cost is high. We define $2^{Des*} = 2^{Des} \setminus \emptyset$.

Let us assume that the set *Atm* includes four types of atomic formulas, for every $x \mapsto v \in Assign$, $o, o' \in Opt$ and $\Gamma \in 2^{Des*}$: (1) $\mathsf{val}(o, x \mapsto v)$ standing for "option o has value v for variable x", (2) $\mathsf{ideal}(\mathfrak{h}, o)$ standing for "o is an ideal option for agent $\mathfrak{h}$", (3) $\mathsf{justif}(\mathfrak{h}, o)$ standing for "agent $\mathfrak{h}$ has a justification for choosing option o", and (4) $\mathsf{des}(\mathfrak{h}, \Gamma)$ standing for "Γ is agent $\mathfrak{h}$'s set of desires".

The following function f_{comp} specifies, for every option $o \in Opt$ and possible desire $\gamma \in Des$, the condition guaranteeing that o satisfies (or, complies with) γ:

$$\begin{aligned}
&f_{comp}(o, a) = \mathsf{val}(o, a),\\
&f_{comp}(o, \sim a) = \neg \mathsf{val}(o, a),\\
&f_{comp}\big(o, [d_1, \ldots, d_k] \rightsquigarrow d\big) = \neg f_{comp}(o, d_1) \vee \ldots \vee \neg f_{comp}(o, d_k) \vee f_{comp}(o, d).
\end{aligned}$$

The following function $f^{\mathfrak{h}}_{comp}$ specifies, for every option $o \in Opt$ and possible desire $\gamma \in Des$, the condition guaranteeing that agent $\mathfrak{h}$ believes that o satisfies γ:

$$
\begin{aligned}
&f^{\mathfrak{h}}_{comp}(o, a) = \triangle_{\mathfrak{h}} f_{comp}(o, a),\\
&f^{\mathfrak{h}}_{comp}(o, \sim a) = \triangle_{\mathfrak{h}} f_{comp}(o, \sim a),\\
&f^{\mathfrak{h}}_{comp}\big(o, [d_1, \ldots, d_k] \rightsquigarrow d\big)\\
&\quad = \triangle_{\mathfrak{h}} \neg f_{comp}(o, d_1) \vee \ldots \vee \triangle_{\mathfrak{h}} \neg f_{comp}(o, d_k) \vee \triangle_{\mathfrak{h}} f_{comp}(o, d).
\end{aligned}
$$

The previous formulation of $f^{\mathfrak{h}}_{comp}\big(o, [d_1, \ldots, d_k] \rightsquigarrow d\big)$ presupposes an understanding of conditional (goal) sentences by agent $\mathfrak{h}$. In particular, agent $\mathfrak{m}$ does not need to provide information to agent $\mathfrak{h}$ about the antecedent of the conditional, if the consequent is true.

6.2 Agent $\mathfrak{m}$'s model of agent $\mathfrak{h}$'s mind

We assume that the artificial agent $\mathfrak{m}$ has the following pieces of information in its belief base:

$$
\begin{aligned}
\alpha_1 &\stackrel{\text{def}}{=} \bigwedge_{\substack{o \in Opt\\ x \in Var\\ v, v' \in Val_x \,:\, v \neq v'}} \Big(\mathsf{val}(o, x \mapsto v) \rightarrow \neg \mathsf{val}(o, x \mapsto v')\Big),\\
\alpha_2 &\stackrel{\text{def}}{=} \bigwedge_{\substack{o \in Opt\\ x \in Var\\ v, v' \in Val_x \,:\, v \neq v'}} \Big(\triangle_{\mathfrak{h}}\, \mathsf{val}(o, x \mapsto v) \rightarrow \triangle_{\mathfrak{h}} \neg \mathsf{val}(o, x \mapsto v')\Big),\\
\alpha_3 &\stackrel{\text{def}}{=} \bigwedge_{\Gamma, \Gamma' \in 2^{Des*} \,:\, \Gamma \neq \Gamma'} \Big(\mathsf{des}(\mathfrak{h}, \Gamma) \rightarrow \neg \mathsf{des}(\mathfrak{h}, \Gamma')\Big),\\
\alpha_4 &\stackrel{\text{def}}{=} \bigvee_{\Gamma \in 2^{Des*}} \mathsf{des}(\mathfrak{h}, \Gamma),\\
\alpha_5 &\stackrel{\text{def}}{=} \bigwedge_{o \in Opt} \Big(\mathsf{ideal}(\mathfrak{h}, o) \leftrightarrow \bigvee_{\Gamma \in 2^{Des*}} \Big(\mathsf{des}(\mathfrak{h}, \Gamma) \wedge \bigwedge_{\gamma \in \Gamma} f_{comp}(o, \gamma)\Big)\Big),\\
\alpha_6 &\stackrel{\text{def}}{=} \bigwedge_{o \in Opt} \Big(\mathsf{justif}(\mathfrak{h}, o) \leftrightarrow \bigvee_{\Gamma \in 2^{Des*}} \Big(\mathsf{des}(\mathfrak{h}, \Gamma) \wedge \bigwedge_{\gamma \in \Gamma} f^{\mathfrak{h}}_{comp}(o, \gamma)\Big)\Big).
\end{aligned}
$$

Formula α_1 captures the fact that a sport cannot have two different values for a given variable. Formula α_2 is its subjective version for agent $\mathfrak{h}$. Formulas α_3 and α_4 capture together the fact that agent $\mathfrak{h}$ has exactly one non-empty set of desires. According to formula α_5, an option o is ideal for agent $\mathfrak{h}$ if and only if it satisfies all agent $\mathfrak{h}$'s desires. Finally, according to formula α_6, agent $\mathfrak{h}$ has a reasonable justification for choosing option o if and only if she has all necessary information to conclude that option o satisfies all her desires.

Note that an alternative encoding of the problem would consist in using atomic formulas $\mathsf{des}(\mathfrak{h}, \gamma)$ with $\gamma \in Des$ standing for "agent $\mathfrak{h}$ has γ as a desire" instead of atomic formulas

$\mathsf{des}(\mathfrak{h}, \Gamma)$. This encoding would be simpler since it would allow us to remove the previous hypothesis α_3 and to replace the hypotheses $\alpha_4, \alpha_5, \alpha_6$ by the following hypotheses $\alpha'_4, \alpha'_5, \alpha'_6$:

$$\alpha'_4 \stackrel{\text{def}}{=} \bigvee_{\gamma \in Des} \mathsf{des}(\mathfrak{h}, \gamma),$$

$$\alpha'_5 \stackrel{\text{def}}{=} \bigwedge_{o \in Opt} \Big(\mathsf{ideal}(\mathfrak{h}, o) \leftrightarrow \bigwedge_{\gamma \in Des} \Big(\mathsf{des}(\mathfrak{h}, \gamma) \rightarrow f_{comp}(o, \gamma) \Big) \Big),$$

$$\alpha'_6 \stackrel{\text{def}}{=} \bigwedge_{o \in Opt} \Big(\mathsf{justif}(\mathfrak{h}, o) \leftrightarrow \bigwedge_{\gamma \in Des} \Big(\mathsf{des}(\mathfrak{h}, \gamma) \rightarrow f^{\mathfrak{h}}_{comp}(o, \gamma) \Big) \Big).$$

Notice that $\alpha_4, \alpha_5, \alpha_6$ are exponential while $\alpha'_4, \alpha'_5, \alpha'_6$ are polynomial in the size of Des.

We chose the more complex encoding with atomic formulas $\mathsf{des}(\mathfrak{h}, \Gamma)$ in the language instead of $\mathsf{des}(\mathfrak{h}, \gamma)$ for conceptual reasons. Indeed, in a realistic conversational context it is reasonable to suppose that an agent can say A!="my set of desires is Γ"! without necessarily enumerating in an explicit way all facts it desires and all facts it does not, that is, without explicitly saying that B!="I desire γ_1, ..., I desire γ_k, I don't desire γ_{k+1}, ..., I don't desire γ_m"! with $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$ and $Des \backslash \Gamma = \{\gamma_{k+1}, \ldots, \gamma_m\}$. In other words, it is reasonable to take $\mathsf{des}(\mathfrak{h}, \Gamma)$ as a primitive and leave to the hearer the deduction that B from hearing the speech act A!.

Of course, we could keep both types of atomic formulas $\mathsf{des}(\mathfrak{h}, \gamma)$ and $\mathsf{des}(\mathfrak{h}, \Gamma)$ in the language and relate them by the following additional hypothesis:

$$\alpha_{des} \stackrel{\text{def}}{=} \bigwedge_{\Gamma \in 2^{Des*}} \Big(\mathsf{des}(\mathfrak{h}, \Gamma) \leftrightarrow \big(\bigwedge_{\gamma \in \Gamma} \mathsf{des}(\mathfrak{h}, \gamma) \wedge \bigwedge_{\gamma \in Des \backslash \Gamma} \neg \mathsf{des}(\mathfrak{h}, \gamma) \big) \Big).$$

But again this would make the encoding exponential in the size of Des since α_{des} requires to quantify over 2^{Des*}.

So, to sum up, three encodings of the problem are possible: (1) only atomic formulas $\mathsf{des}(\mathfrak{h}, \Gamma)$ are in the language and we use hypotheses $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$, (2) only atomic formulas $\mathsf{des}(\mathfrak{h}, \gamma)$ are in the language and we use hypotheses $\alpha_1, \alpha_2, \alpha'_4, \alpha'_5, \alpha'_6$, (3) both types of atomic formulas are in the language and we use hypotheses $\alpha_1, \alpha_2, \alpha'_4, \alpha'_5, \alpha'_6, \alpha_{des}$. The encoding (2) is polynomial in the size of Des, while the encodings (1) and (3) are exponential. As explained above, we prefer (1) to (2) for conceptual reasons. We prefer (1) to (3) since we do not need formulas of type $\mathsf{des}(\mathfrak{h}, \gamma)$ in our formulation of the problem.

The experimental comparison of the brute force and the QBF-based approach we will present in Sect. 7.3 is relative to the encoding (1). We leave for future work an experimental comparison relative to the encodings (2) and (3).

We also assume that agent $\mathfrak{m}$ has in its belief base a complete representation of Table 1, which specifies the variable assignments for all options:

$$\alpha_7^{o,x} \stackrel{\text{def}}{=} \mathsf{val}(o, x \mapsto v_{o,x}).$$

We suppose that the pieces of information $\alpha_1, \ldots, \alpha_6$ together with $\alpha_7^{o,x}$ for every $o \in Opt$ and $x \in Var$ constitute agent $\mathfrak{m}$'s initial core belief base, that is, $\Sigma^0_{core} = \{\alpha_1, \ldots, \alpha_6\} \cup \{\alpha_7^{o,x} : o \in Opt \text{ and } x \in Var\}$. Agent $\mathfrak{m}$'s initial mutable belief base is supposed to be empty, that is, $\Sigma^0_{mut} = \emptyset$. We define $\Sigma^0_{base} = \Sigma^0_{core} \cup \Sigma^0_{mut}$.

Table 1 Variable assignments

Options	Variables					
	env	loc	soc	cost	dan	intens
sw	*water*	*mixed*	*single*	*med*	*low*	*high*
ru	*land*	*outdoor*	*single*	*low*	*med*	*high*
hr	*land*	*outdoor*	*single*	*high*	*high*	*low*
te	*land*	*mixed*	*mixed*	*high*	*med*	*med*
so	*land*	*mixed*	*team*	*med*	*med*	*med*
yo	*land*	*mixed*	*single*	*med*	*low*	*low*
di	*water*	*mixed*	*single*	*high*	*high*	*low*
sq	*land*	*indoor*	*mixed*	*high*	*med*	*med*

For every option $o \in Opt$ and variable $x \in Var$, we denote by $v_{o,x}$ the corresponding entry in the table. For instance, we have $v_{\mathrm{sw},\mathbf{env}} = water$

6.3 Instantiation of the planning problem

Let us now turn to the cognitive planning problem. We define agent $\mathfrak{m}$'s set of operators Op as follows:

$$Op = \{convince(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a)) : o \in Opt \text{ and } a \in Assign\} \cup \{convince(\mathfrak{m},\mathfrak{h},\mathsf{ideal}(\mathfrak{h},o)) : o \in Opt\}.$$

In other words, agent $\mathfrak{m}$ can only inform agent $\mathfrak{h}$ about an option's value for a certain variable or about the ideality of an option for her.

We use the speech act *convince* since we suppose agent $\mathfrak{h}$ fully trusts what agent $\mathfrak{m}$ says (i.e., $\mathfrak{h}$ believes that $\mathfrak{m}$ is both sincere and competent).

We suppose the following executability precondition for every $o \in Opt$ and $a \in Assign$ (recall that $Assign_x$ has been defined p. 40):

$$\mathcal{P}\Big(convince(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a))\Big) = \Box_{\mathfrak{m}}\Big(\mathsf{val}(o,a) \wedge \bigwedge_{v \in Val_{\mathbf{dan}}} (\mathsf{val}(o,\mathbf{dan} \mapsto v) \rightarrow \triangle_{\mathfrak{h}}\, \mathsf{val}(o,\mathbf{dan} \mapsto v))\Big) \quad \text{if } a \notin Assign_{\mathbf{dan}},$$

$$\mathcal{P}\Big(convince(\mathfrak{m},\mathfrak{h},\mathsf{val}(o,a))\Big) = \Box_{\mathfrak{m}}\mathsf{val}(o,a) \quad \text{if } a \in Assign_{\mathbf{dan}},$$

$$\mathcal{P}\Big(convince(\mathfrak{m},\mathfrak{h},\mathsf{ideal}(\mathfrak{h},o))\Big) = \Box_{\mathfrak{m}}\big(\mathsf{ideal}(\mathfrak{h},o) \wedge \mathsf{justif}(\mathfrak{h},o)\big).$$

According to the first definition, agent $\mathfrak{m}$ can inform agent $\mathfrak{h}$ about an option's value for a certain variable, if and only if this information is believed by $\mathfrak{m}$ and $\mathfrak{m}$ believes that $\mathfrak{h}$ has been already informed about the dangerousness level of the option. Indeed, we assume that, before being presented with an option's features, agent $\mathfrak{h}$ must be informed about its the dangerousness level and agent $\mathfrak{m}$ complies with this rule.[7] The second definition simply

[7] This rule arises from discussions we had with sociologists and ergonomists: if people refuse to do a dangerous activity, most of the time they prefer to be informed quickly. This also illustrates the flexibility of our language which allows, if desired, to completely order the answers given. This choice has no technical consequence.

stipulates that $\mathfrak{m}$ can inform $\mathfrak{h}$ about the dangerousness level of an option if and only if it believes what it says. Finally, according to the third definition, $\mathfrak{m}$ can inform $\mathfrak{h}$ about the ideality of an option only if it believes that $\mathfrak{h}$ has a reasonable justification for choosing it. Indeed, we assume $\mathfrak{m}$ will inform $\mathfrak{h}$ about the ideality of an option only after having explained why the option is ideal for her. The three definitions presuppose that agent $\mathfrak{m}$ cannot spread fake news (i.e., something that it does not implicitly believe).

We moreover suppose that agent $\mathfrak{m}$ has the influencing goal that $\mathfrak{h}$ will form the potential intention to practice a sport activity. In particular, agent $\mathfrak{m}$ wants to provide an effective recommendation to agent $\mathfrak{h}$ that will induce her to choose a sport activity to practice. In order to define such a goal, we must first define the concept of potential intention. We assume that, for agent $\mathfrak{h}$ to have a potential intention to choose option o, noted $\mathsf{potIntend}(\mathfrak{h}, o)$, she must have a justified belief that o is an ideal option for her[8]:

$$\mathsf{potIntend}(\mathfrak{h}, o) \stackrel{\text{def}}{=} \triangle_{\mathfrak{h}}\, \mathsf{ideal}(\mathfrak{h}, o) \wedge \mathsf{justif}(\mathfrak{h}, o).$$

This abbreviation together with the abbreviation α_6 given above relate intention with belief and desire, in line with existing theories of intention [3, 17].

Agent $\mathfrak{m}$'s influencing goal α_G in the cognitive planning problem is then defined as follows:

$$\alpha_G \stackrel{\text{def}}{=} \bigvee_{o \in Opt} \mathsf{potIntend}(\mathfrak{h}, o).$$

6.4 Example of interaction

At every step k of the interaction with agent $\mathfrak{h}$, agent $\mathfrak{m}$ tries to find a solution for the cognitive planning problem $\langle \Sigma^k_{base}, Op, \alpha_G \rangle$. This is the core aspect of the informative phase. If it can find it, it proceeds with its execution and then interaction stops. Otherwise, it enters the exploratory phase in order to gather information about agent $\mathfrak{h}$'s cognitive state. The exploratory phase is handled by the belief revision module. After having revised its belief base in the light of the information Σ^k_{input} provided by agent $\mathfrak{h}$, agent $\mathfrak{m}$ moves to step $k+1$. We suppose that

$$\begin{aligned} \Sigma^{k+1}_{core} &= Rev^{core}(\Sigma^k_{core}, \Sigma^k_{mut}, \Sigma^k_{input}), \\ \Sigma^{k+1}_{mut} &= Rev^{mut}(\Sigma^k_{core}, \Sigma^k_{mut}, \Sigma^k_{input}). \end{aligned}$$

Let us illustrate an example of interaction. It is easy to verify that the cognitive planning problem $\langle \Sigma^0_{base}, Op, \alpha_G \rangle$ has no solution. The reason is that in the initial situation agent $\mathfrak{m}$ lacks information about agent $\mathfrak{h}$'s desires. Therefore, it does not know how to influence her.

Thus, agent $\mathfrak{m}$ enters the exploratory phase during which agent $\mathfrak{h}$ discloses her actual desires to agent $\mathfrak{m}$. Let us suppose $\Sigma^0_{input} = \{\mathsf{des}(\mathfrak{h}, \Gamma_{\mathfrak{h}})\}$ with

[8] Our account of potential intention is reminiscent of the JTB ('justified true belief') account to knowledge [24].

$$\Gamma_{\mathfrak{h}} = \{\mathbf{env} \mapsto \mathit{land}, \mathbf{intens} \mapsto \mathit{med}, \sim\mathbf{loc} \mapsto \mathit{indoor}, \mathbf{dan} \mapsto \mathit{low}, \\ [\mathbf{cost} \mapsto \mathit{high}] \rightsquigarrow \mathbf{soc} \mapsto \mathit{mixed}\}.$$

This means that agent $\mathfrak{h}$ informs agent $\mathfrak{m}$ that she would like to practice a land activity, with medium intensity, which is not exclusively indoor, with low danger, and which can be practiced both in single and team mode, if its cost is high.

As for step 0, it is easy to verify that the cognitive planning problem $\langle \Sigma^1_{base}, Op, \alpha_G \rangle$ has still no solution. Indeed, according to agent $\mathfrak{m}$'s beliefs, there is no sport which meets all agent $\mathfrak{h}$'s desires.

Therefore, agent $\mathfrak{m}$ enters a new exploratory phase during which it asks to agent $\mathfrak{h}$ to make a concession, namely, to be less demanding by putting aside some of her desires. Let us suppose agent $\mathfrak{h}$ positively replies to agent $\mathfrak{m}$'s request by dropping the requirement that an ideal sport should have a low level of dangerousness. In other words, we have $\Sigma^1_{input} = \{\mathsf{des}(\mathfrak{h}, \Gamma'_{\mathfrak{h}})\}$ with $\Gamma'_{\mathfrak{h}} = \Gamma_{\mathfrak{h}} \backslash \{\mathbf{dan} \mapsto \mathit{low}\}$. Note that at this stage agent $\mathfrak{m}$ has to revise its mutable belief base. In particular, it has to drop the information $\mathsf{des}(\mathfrak{h}, \Gamma_{\mathfrak{h}})$ from its mutable belief base since the new information $\mathsf{des}(\mathfrak{h}, \Gamma'_{\mathfrak{h}})$ is in conflict with it, due to the presence of the information α_3 in its core belief base.

At step 2, agent $\mathfrak{m}$ can find a solution plan. In particular, it turns out that the sequence of speech acts $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6$ with

$$\begin{aligned}
\epsilon_1 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(\mathrm{te}, \mathbf{dan} \mapsto \mathit{med})\big), \\
\epsilon_2 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(\mathrm{te}, \mathbf{env} \mapsto \mathit{land})\big), \\
\epsilon_3 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(\mathrm{te}, \mathbf{intens} \mapsto \mathit{med})\big), \\
\epsilon_4 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(\mathrm{te}, \mathbf{loc} \mapsto \mathit{mixed})\big), \\
\epsilon_5 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(\mathrm{te}, \mathbf{soc} \mapsto \mathit{mixed})\big), \\
\epsilon_6 &\stackrel{\text{def}}{=} \mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{ideal}(\mathfrak{h}, \mathrm{te})\big).
\end{aligned}$$

provides a solution for the cognitive planning problem $\langle \Sigma^2_{base}, Op, \alpha_G \rangle$. This means that, by performing the sequence of operators $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6$ at step 2, agent $\mathfrak{m}$ will induce agent $\mathfrak{h}$ to form a potential intention to choose an activity. In other words, agent $\mathfrak{m}$ will provide an effective recommendation to agent $\mathfrak{h}$.

The previous interaction between agent $\mathfrak{m}$ and agent $\mathfrak{h}$ is schematically illustrated in Table 2.

We conclude this section with a general observation about the formulation of the cognitive planning problem for our example. Let $\mathfrak{m}$'s set of operators for option $o \in \mathit{Opt}$ relative to $\langle \Sigma^2_{base}, Op, \alpha_G \rangle$ be defined as follows:

$$\begin{aligned}
Op_o^{\langle \Sigma^2_{base}, Op, \alpha_G \rangle} = &\big\{\mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{val}(o, a)\big) \,:\, \mathsf{val}(o, a) \in \Sigma^2_{base}\big\} \cup \\
&\big\{\mathit{convince}\big(\mathfrak{m}, \mathfrak{h}, \mathsf{ideal}(\mathfrak{h}, o)\big)\big\}.
\end{aligned}$$

It is easy to verify that the cognitive planning problem $\langle \Sigma^2_{base}, Op, \alpha_G \rangle$ has a solution if and only if there exists $o \in \mathit{Opt}$ such that the cognitive planning problem $\langle \Sigma^2_{base}, Op_o^{\langle \Sigma, Op, \alpha_G \rangle}, \alpha_G \rangle$ has a solution. Therefore, in order to solve the cognitive planning problem $\langle \Sigma^2_{base}, Op_o^{\langle \Sigma, Op, \alpha_G \rangle}, \alpha_G \rangle$, we simply need to linearly order the options in Opt and

Table 2 Human–machine interaction

Speaker	Utterance	Phase
$\mathfrak{m}$	I cannot find any solution. Could you please tell me what you desire, to be able to find the ideal sport for you?	Informative
$\mathfrak{h}$	I would like to practice a land activity, with medium intensity, which is not exclusively indoor, with low danger, and which can be practiced both in single and team mode, if its cost is high.	Exploratory
$\mathfrak{m}$	I cannot find any solution which satisfies all your desires. Could you please make a concession by putting aside some of them?	Informative
$\mathfrak{h}$	I would like to practice a land activity, with medium intensity, which is not exclusively indoor, and which can be practiced both in single and team mode, if its cost is high.	Exploratory
$\mathfrak{h}$	Tennis is an activity with medium danger (ϵ_1), whose environment is land (ϵ_2), with medium intensity (ϵ_3), which can be practiced both indoor and outdoor (ϵ_4), and both in single and team mode (ϵ_5). For all these reasons, it is the ideal sport for you (ϵ_6).	Informative

solve the cognitive planning problems $\langle \Sigma, Op_o^{\langle \Sigma, Op, \alpha_G \rangle}, \alpha_G \rangle$ in sequence one after the other according to the ordering.

The scenario presented in this section does not exhaust the application potential of our approach. In a related work [43] we have designed an artificial agent aimed at motivating the user to practice a physical activity regularly. The logical specification of the agent was made in conformity with motivational interviewing, a counseling method used in clinical psychology for eliciting behavior change [47].

6.5 Implementation

This part of the document illustrates the implementation of our artificial assistant. The implemented system[9] allows us to represent and reason about other agents' beliefs, desires,

[9] https://github.com/iritlab/artificial_agent.

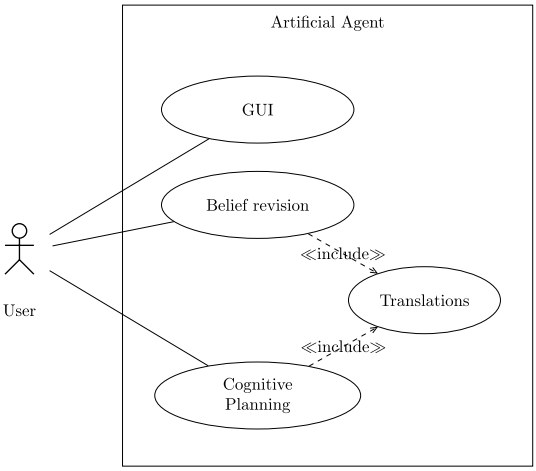


Fig. 4 Articifial agent use case diagram

and intentions using our NP-complete fragment. The use case diagram in Fig. 4 represents our system functionality. It considers four use cases: the graphical user interface (GUI), the belief revision, the cognitive planning, and the translator module.

The GUI[10] is used by the artificial agent to interact with the human agent in order to collect information about her desires and preferences and to perform the utterances cor-responding to the sequence of speech acts resulting from the planning process. The *trans-lations* module encapsulates the set of reductions showed in Fig. 3. This set of reductions is used by both the belief revision and the cognitive planning module. The *belief revision* module revises the belief base with the information provided by the human agent, as speci-fied in Sect. 5.2.

The *cognitive planning* module reads the initial state, the set of actions, and the goal and starts to generate candidate plans. We show the detailed system architecture in Fig. 5.

The main implementation of the cognitive planning functionality relies on a brute force algorithm based on SAT. It starts with plans of length 1, and enters in a loop. At each interaction, the planning module asks the SAT solver to verify whether the plan allows to achieve the goal. If no plan of length k is found, the program will increase the counter in one and look for a plan of length k+1. However, it is possible to switch to a QBF-based algorithm for cognitive planning that we detail in the next section.

The SAT-based brute force algorithm relies on two facts. First of all, thanks to Proposition 4, verifying whether a plan is a solution plan can be formulated as a satisfiability checking problem of a $\mathcal{L}^{+}_{\mathsf{Frag}}$-formula φ . Secondly, thanks to Propositions 2 and 3, Theorems 2 and 3, the fact that the size of $red(\varphi)$ is polynomial in the size of φ and that translations tr_1 and tr_2 are polynomial, we have a polysize reduction of satisfiability checking of φ to SAT.

[10] Note that a 3-D avatar web interface was implemented as the GUI module (see https://cognitive-planning.schm.fr/). It has been developed by the DAVI company (https://davi.ai/en/home/). It behaves in conformity with the formal specification given in Sect. 6. In the exploratory phase, it interacts with the human user in order to collect information about her desires. Then, in the informative phase, it computes a plans consisting of a sequence of assertions aimed at persuading the human to practice a sport in line with her preferences.

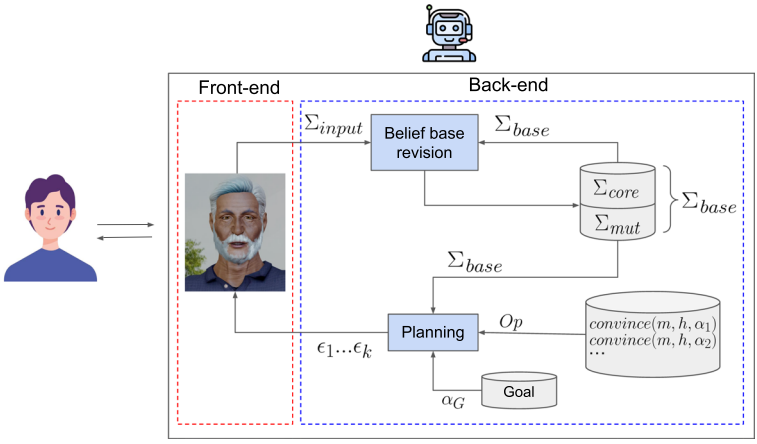


Fig. 5 System architecture

7 A QBF-based approach

We present now a QBF encoding for checking plan existence in a cognitive planning problem with prefix $\exists\forall$. Intuitively, a solution plan candidate is non-deterministically chosen ($\exists$) and the validity of this plan is checked ($\forall$). Moreover, we will empirically compare this QBF-based approach with the brute force SAT-based approach in terms of time needed for computing a solution.

7.1 Extension with selectors

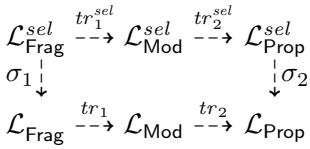
We first introduce the new language $\mathcal{L}^{sel}_{\mathsf{Frag}}$ that extends the language $\mathcal{L}_{\mathsf{Frag}}$ with *selector* propositional variables in order to represent, within a single formula, different formulas of $\mathcal{L}_{\mathsf{Frag}}$ depending on the truth value of these *selector* variables. In other words, for any valuation of selectors we can syntactically simplify the formula of $\mathcal{L}^{sel}_{\mathsf{Frag}}$ substituting selectors by $\top$ or $\bot$ to obtain a formula of $\mathcal{L}_{\mathsf{Frag}}$. The idea behind the encoding of checking plan existence into QBF is to adapt the reduction process from $\mathcal{L}^{+}_{\mathsf{Frag}}$ to $\mathcal{L}_{\mathsf{Prop}}$ using selectors so that each valuation of the selectors designates a specific plan candidate among all the possible plans for the problem. So we define the following extension of the language $\mathcal{L}_{\mathsf{Frag}}$ called $\mathcal{L}^{sel}_{\mathsf{Frag}}$ as followed:

$$\begin{array}{lll} \beta & ::= & \alpha \mid \neg\beta \mid \beta \wedge \beta \mid s, \\ \varphi & ::= & \beta \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_{\mathfrak{m}}\beta, \end{array}$$

where α ranges over $\mathcal{L}_0$, $\mathfrak{m}$ is the special agent in Agt called the 'machine', and s ranges over Atm^{sel} a countable set of selectors distinct from all other atoms defined in this paper. Note that selector variables can be used in implicit beliefs but not in explicit beliefs. We also extend the modal language $\mathcal{L}_{\mathsf{Mod}}$ to the language $\mathcal{L}^{sel}_{\mathsf{Mod}}$:

$$\begin{array}{lll} \omega & ::= & q \mid \neg\omega \mid \omega \wedge \omega \mid s, \\ \varphi & ::= & q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \blacksquare\omega, \end{array}$$

Fig. 6 Summary of reduction processes with selectors



where q ranges over Atm^+ and s ranges again over Atm^{sel}. And finally, to add selectors to $\mathcal{L}_{\mathsf{Prop}}$ we define the propositional logic language $\mathcal{L}^{sel}_{\mathsf{Prop}}$ built from the set of atomic propositions $Atm^{++} \cup Atm^{sel}$. In order to propagate the selector variables the right way, we extend on $\mathcal{L}^{sel}_{\mathsf{Frag}}$ and $\mathcal{L}^{sel}_{\mathsf{Mod}}$ the definitions of translations tr_1 and tr_2 respectively to tr_1^{sel} and tr_2^{sel} by adding to their inductive definitions the two rules:

$$tr_1^{sel}(s) = s$$
$$tr_2^{sel}(s, x, y) = s$$

and we extend the definition of the size of a formula from $\mathcal{L}_{\mathsf{Mod}}$ to a formula from $\mathcal{L}^{sel}_{\mathsf{Mod}}$ by adding $size(s) = 1$ for $s \in Atm^{sel}$.

As we will only use selectors as a technical trick over the reduction process, it is not useful to define semantics for the languages $\mathcal{L}^{sel}_{\mathsf{Frag}}$ and $\mathcal{L}^{sel}_{\mathsf{Mod}}$. It is sufficient to define, in order to designate a particular formula of $\mathcal{L}_{\mathsf{Frag}}$ given a fixed valuation $V_{sel} \subseteq Atm^{sel}$ of selectors, a syntactical simplification function $\sigma_1 : 2^{Atm^{sel}} \times \mathcal{L}^{sel}_{\mathsf{Frag}} \to \mathcal{L}_{\mathsf{Frag}}$. The formula $\sigma_1(V_{sel}, \varphi)$ is the result of the successive applications of all following uniform substitutions on φ: $[\top/s]$ for all $s \in V_{sel}$ and $[\bot/s]$ for all $s \in Atm^{sel} \backslash V_{sel}$. Moreover, in order to designate a particular formula of $\mathcal{L}_{\mathsf{Prop}}$ given a fixed valuation $V_{sel} \subseteq Atm^{sel}$ of selectors, we define exactly the same way another syntactical simplification $\sigma_2 : 2^{Atm^{sel}} \times \mathcal{L}^{sel}_{\mathsf{Prop}} \to \mathcal{L}_{\mathsf{Prop}}$. We then have the relations between the reduction processes over languages with or without selectors given in Fig. 6.

The following proposition states that the two ways to reduce and simplify a formula from $\mathcal{L}^{sel}_{\mathsf{Frag}}$ to $\mathcal{L}_{\mathsf{Prop}}$, given a valuation of selectors, lead to equisatisfiable propositional formulas.

Proposition 6 *Given a formula $\varphi \in \mathcal{L}^{sel}_{\mathsf{Frag}}$ and a valuation $V_{sel} \subseteq Atm^{sel}$ of selectors of φ, the following two propositional formulas in $\mathcal{L}_{\mathsf{Prop}}$ are equisatisfiable*:

$$\sigma_2\Big(V_{sel}, tr_2^{sel}\Big(tr_1^{sel}(\varphi), 0, size\Big(tr_1^{sel}(\varphi)\Big)\Big)\Big),$$
$$tr_2\Big(tr_1\big(\sigma_1(V_{sel}, \varphi)\big), 0, size\Big(tr_1\big(\sigma_1(V_{sel}, \varphi)\big)\Big)\Big).$$

(The proof is provided in "Appendix")

7.2 Encoding

For a given cognitive planning problem $\langle \Sigma, Op, \alpha_G \rangle$, by Proposition 5 and without loss of generality, we can only consider poly-size solution plan candidates of the form $\epsilon_1, \ldots, \epsilon_k$

with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$. We define the set of selector variables $Sel_{Op} = \{s_{\epsilon \preceq \epsilon'} : \epsilon, \epsilon' \in Op\} \subseteq Atm^{sel}$ and the formula $\varphi_{Sel_{Op}}$ as the conjunction of the following axioms:

$$\bigwedge_{\epsilon \in Op} \left(\neg s_{\epsilon \preceq \epsilon} \rightarrow \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \neg s_{\epsilon \preceq \epsilon'} \wedge \neg s_{\epsilon' \preceq \epsilon} \right) \tag{S1}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \left(\neg s_{\epsilon \preceq \epsilon'} \vee \neg s_{\epsilon' \preceq \epsilon} \right) \tag{S2}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \bigwedge_{\substack{\epsilon'' \in Op \\ \epsilon \neq \epsilon'' \\ \epsilon' \neq \epsilon''}} \left(s_{\epsilon \preceq \epsilon'} \wedge s_{\epsilon' \preceq \epsilon''} \rightarrow s_{\epsilon \preceq \epsilon''} \right) \tag{S3}$$

$$\bigwedge_{\epsilon \in Op} \bigwedge_{\substack{\epsilon' \in Op \\ \epsilon \neq \epsilon'}} \left(s_{\epsilon \preceq \epsilon} \wedge s_{\epsilon' \preceq \epsilon'} \rightarrow s_{\epsilon \preceq \epsilon'} \vee s_{\epsilon' \preceq \epsilon} \right) \tag{S4}$$

These axioms are constructed in order to set a bijective function between the models of $\varphi_{Sel_{Op}}$ and the solution plan candidates for $\langle \Sigma, Op, \alpha_G \rangle$. In the sequel, for a given model of $\varphi_{Sel_{Op}}$, the corresponding plan will be called the designated plan. Intuitively, we define a total order $\preceq$ between the elements of the designated plan, and other actions from Op are not ordered at all. Note that this assumption implies the reflexivity of $\preceq$ on elements of the designated plan and only on these elements of Op. Axiom S1 states that if an action $\epsilon \in Op$ is not $\preceq$-reflexive, i.e. is not selected in the designated plan, then no other action $\epsilon' \in Op$ is $\preceq$-related with ϵ. The following axioms define asymmetry (S2) and transitivity (S3) of $\preceq$ on elements of the designated plan. Finally, axiom S4 states that $\preceq$ is total on elements of the designated plan.

A plan candidate $\epsilon_1, \ldots, \epsilon_k$ is a solution plan for $\langle \Sigma, Op, \alpha_G \rangle$ if and only if the following formula is valid in the class $\mathbf{M}$:

$$\left(\bigwedge_{\alpha \in \Sigma} \Box_{\mathrm{m}} \alpha \right) \rightarrow \langle\!\langle \epsilon_1 \rangle\!\rangle \ldots \langle\!\langle \epsilon_{k-1} \rangle\!\rangle \langle\!\langle \epsilon_k \rangle\!\rangle \Box_{\mathrm{m}} \alpha_G$$

which can be also written as:

$$\left(\bigwedge_{\alpha \in \Sigma} \Box_{\mathrm{m}} \alpha \right) \rightarrow \left(\bigwedge_{i \in \{1, \ldots, k\}} [+_{\mathrm{m}} \alpha_{\epsilon_1}] \ldots [+_{\mathrm{m}} \alpha_{\epsilon_{i-1}}] \mathcal{P}(\epsilon_i) \right) \wedge [+_{\mathrm{m}} \alpha_{\epsilon_1}] \ldots [+_{\mathrm{m}} \alpha_{\epsilon_{k-1}}] [+_{\mathrm{m}} \alpha_{\epsilon_k}] \Box_{\mathrm{m}} \alpha_G$$

Note that for a given $i \in \{1, \ldots, k\}$, we have

$$red\big([+_{\mathfrak{m}}\alpha_{\epsilon_1}]\dots[+_{\mathfrak{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\big) = \begin{cases} \top & \text{if } \mathcal{P}(\epsilon_i) = \bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon_j} \text{ for some } j < i, \\ red\big(\mathcal{P}(\epsilon_i)\big) & \text{otherwise.} \end{cases}$$

In words, if the precondition $\mathcal{P}(\epsilon_i)$ of an operator ϵ_i is an explicit belief $\bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon_j}$ of agent $\mathfrak{m}$ which is added by another operator ϵ_j which precedes ϵ_i in the plan candidate, then the reduction of the test of the precondition of ϵ_i into $\mathcal{L}_{\mathsf{Frag}}$ is set to $\top$. Hence, to focus on a particular designated plan, we can use a selector variable $s_{\epsilon_j \leq \epsilon_i}$ to generate a reduction into $\mathcal{L}^{sel}_{\mathsf{Frag}}$ depending on the value of this selector by replacing $\top$ by $s_{\epsilon_j \leq \epsilon_i} \vee \bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon_j}$. Indeed, if ϵ_j is selected and precedes ϵ_i in a designated plan then $[+_{\mathfrak{m}}\alpha_{\epsilon_j}]$ should be taken into account in the reduction of the precondition, and $s_{\epsilon_j \leq \epsilon_i}$ and the reduction are equivalent to $\top$. Else, if ϵ_j is not selected or doesn't precede ϵ_i in a designated plan then $[+_{\mathfrak{m}}\alpha_{\epsilon_j}]$ shouldn't be taken into account in the reduction of the precondition, and $s_{\epsilon_j \leq \epsilon_i}$ is equivalent to $\bot$ and the reduction is equivalent to $\bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon_j}$.

Moreover, if $\mathcal{P}(\epsilon_i)$ contains implicit belief subformulas of the form $\Box_{\mathfrak{m}}\alpha$, the reduction of such a subformula is given by:

$$\begin{aligned} red\big([+_{\mathfrak{m}}\alpha_{\epsilon_1}]\dots[+_{\mathfrak{m}}\alpha_{\epsilon_{i-1}}]\Box_{\mathfrak{m}}\alpha\big) &= \Box_{\mathfrak{m}}\Big(\alpha_{\epsilon_1} \to \big(\dots \to (\alpha_{\epsilon_{i-1}} \to \alpha)\dots\big)\Big) \\ &= \Box_{\mathfrak{m}}\Big(\Big(\bigvee_{j\in\{1,\dots,i-1\}} \neg\alpha_{\epsilon_j}\Big) \vee \alpha\Big) \end{aligned}$$

As previously, we can use a selector variable $s_{\epsilon_j \leq \epsilon_i}$ to generate a reduction into $\mathcal{L}^{sel}_{\mathsf{Frag}}$ depending on the value of this selector by replacing $\neg\alpha_{\epsilon_j}$ by $s_{\epsilon_j \leq \epsilon_i} \wedge \neg\alpha_{\epsilon_j}$. Indeed, if ϵ_j is selected and precedes ϵ_i in a designated plan then $s_{\epsilon_j \leq \epsilon_i}$ is equivalent to $\top$ and the disjunct $\neg\alpha_{\epsilon_j}$ is present in the reduction, else $s_{\epsilon_j \leq \epsilon_i}$ is equivalent to $\bot$ and the disjunct $\neg\alpha_{\epsilon_j}$ is absent from the reduction.

Finally, we can then define a function generating the reduction of the precondition $\mathcal{P}(\epsilon)$ of any action ϵ from a designated plan, depending on all the actions ϵ' that precede ϵ in the designated plan given by a model of $\varphi_{Sel_{Op}}$. For $\epsilon \in Op$, we define such a function $\Pi_{\leq\epsilon} : \mathcal{L}_{\mathsf{Frag}} \longrightarrow \mathcal{L}^{sel}_{\mathsf{Frag}}$ such that:

$$\begin{aligned} \Pi_{\leq\epsilon}(p) &= p \\ \Pi_{\leq\epsilon}(\alpha) &= \begin{cases} s_{\epsilon'\leq\epsilon} \vee \bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon'} & \text{if } \exists\epsilon' \neq \epsilon : \alpha = \bigtriangleup_{\mathfrak{m}}\alpha_{\epsilon'} \\ \alpha & \text{otherwise} \end{cases} \\ \Pi_{\leq\epsilon}(\neg\varphi) &= \neg\Pi_{\leq\epsilon}(\varphi) \\ \Pi_{\leq\epsilon}(\varphi_1 \wedge \varphi_2) &= \Pi_{\leq\epsilon}(\varphi_1) \wedge \Pi_{\leq\epsilon}(\varphi_2) \\ \Pi_{\leq\epsilon}(\Box_{\mathfrak{m}}\alpha) &= \Box_{\mathfrak{m}}\Big(\Big(\bigvee_{\substack{\epsilon'\in Op \\ \epsilon\neq\epsilon'}} s_{\epsilon'\leq\epsilon} \wedge \neg\alpha_{\epsilon'}\Big) \vee \alpha\Big) \end{aligned}$$

For the goal, we proceed in a similar manner to calculate the reduction depending on the selector variables, and we obtain the formula of $\mathcal{L}^{sel}_{\mathsf{Frag}}$:

$$\varphi_P = \Box_{\mathfrak{m}}\left(\left(\bigvee_{\epsilon\in Op} s_{\epsilon\preceq\epsilon}\wedge\neg\alpha_\epsilon\right)\vee\alpha_G\right)$$

It is now easily seen that, depending on the values of selector variables, the following formula of $\mathcal{L}^{sel}_{\mathsf{Frag}}$ represents all possible formulas of $\mathcal{L}_{\mathsf{Frag}}$ that allows us to check the validity of all plan candidates for the planning problem $\langle\Sigma, Op, \alpha_G\rangle$:

$$\varphi_{\langle\Sigma,Op,\alpha_G\rangle} = \varphi_{Sel_{Op}}\wedge\left(\left(\bigwedge_{\alpha\in\Sigma}\Box_{\mathfrak{m}}\alpha\right)\rightarrow\bigwedge_{\epsilon\in Op}\left(s_{\epsilon\preceq\epsilon}\rightarrow\Pi_{\preceq\epsilon}(\mathcal{P}(\epsilon))\right)\wedge\varphi_P\right)$$

Proposition 7 *Given a plan candidate $P = \epsilon_1, \ldots, \epsilon_k$ with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$, if we consider the valuation V_{sel} of selectors in Sel_{Op} such that P is the corresponding designated plan, then $\sigma_1(V_{sel}, \varphi_{\langle\Sigma,Op,\alpha_G\rangle}) \equiv red\left(\left(\bigwedge_{\alpha\in\Sigma}\Box_{\mathfrak{m}}\alpha\right)\rightarrow\langle\!\langle\epsilon_1\rangle\!\rangle\ldots\langle\!\langle\epsilon_{k-1}\rangle\!\rangle\langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathfrak{m}}\alpha_G\right)$.*

(The proof is provided in "Appendix")

Using respectively functions tr^{sel}_1 and tr^{sel}_2, we can calculate the reduction of $\varphi_{\langle\Sigma,Op,\alpha_G\rangle}$ into propositional logic as $F_{\langle\Sigma,Op,\alpha_G\rangle} = tr^{sel}_2\left(tr^{sel}_1(\varphi_{\langle\Sigma,Op,\alpha_G\rangle}), 0, size\left(tr^{sel}_1(\varphi_{\langle\Sigma,Op,\alpha_G\rangle})\right)\right)$ and we denote by $Atm^{++}(F_{\langle\Sigma,Op,\alpha_G\rangle})$ the set of all propositional variables from Atm^{++} occurring in the formula $F_{\langle\Sigma,Op,\alpha_G\rangle}$. We then have the following theorem.

Theorem 8 *The cognitive planning problem $\langle\Sigma, Op, \alpha_G\rangle$ has a solution plan iff the following quantified boolean formula is true*:

Table 4 Translator module and TouIST solver processing times (in s)

Plan size	Translator module		TouIST module	
	Prop. logic	QBF	MiniSat	RAReQS
4	0.004	4.340	0.006	15.890
5	0.003	4.663	0.018	14.819
6	0.005	4.983	0.016	16.428
7	0.013	5.660	0.016	18.472
8	0.012	6.171	0.029	18.912
9	0.013	6.944	0.034	23.712

$$Q = \exists_{s_{\epsilon\preceq\epsilon'}\in Sel_{Op}} s_{\epsilon\preceq\epsilon'} \forall_{q\in Atm^{++}(F_{\langle\Sigma,Op,\alpha_G\rangle})} q \quad F_{\langle\Sigma,Op,\alpha_G\rangle}$$

(The proof is provided in "Appendix")

This theorem allows us to check plan existence and to compute, when exists, a solution plan from the truth values of selectors given by a QBF solver.

7.3 Experimental comparison

In this section, we present the experimental trials conducted in order to test our implemented cognitive planning system on the artificial assistant application that we introduced.

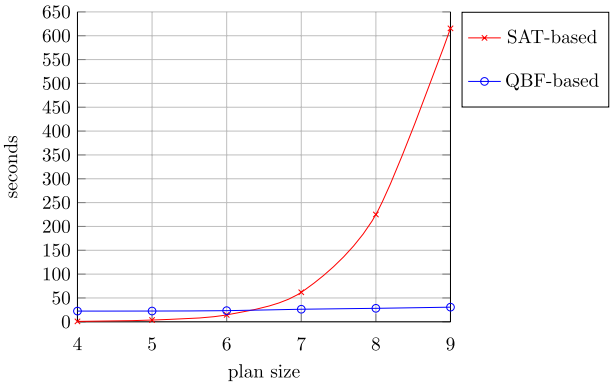


Fig. 7 Computation times for the SAT-based and the QBF-based approach

These experiments evaluate the performance of the SAT-based and the QBF-based approaches of the planning module, in integration with the belief revision module. The GUI was not used during the test; therefore, the procedure was carried out on command line mode.

In order to perform the test we generate first a set of desires of the human in different input files. These input files are processed by the belief revision module sequentially in order to generate the mutable part of the belief base. Second, the translator module is called to generate the initial state and the goal. Finally, the planning module is called, using MiniSat[11] when SAT-based approach is selected, or RAReQS[12] when QBF-approach is selected.

In the experimental analysis the bound for the length of a plan was arbitrarily set to 9. Nonetheless, in a dialogical context like the one investigated in Sect. 6, we do not expect a human to be able to process sequences of speech acts of much greater length due to her cognitive limitations. The comparison of computation times for the SAT-based and the QBF-based approach given in Fig. 7 shows that each approach should be preferred depending on the length of the cognitive plan. When the size of the computed plan is strictly shorter than 6 than the brute force approach outperforms the QBF-based approach. For plans strictly larger than 6 the QBF-based approach outperforms the brute force approach, and the performance of the former is exponentially better than the performance of the latter. Since we do no know the length of the cognitive plan before, the parallel execution of both approaches could be used to generate efficiently such a plan.[13]

To identify possible bottlenecks during the process, we decomposed the amount of execution time spent by each module in the system. Table 3 indicates, for different maximum lengths of the plan sought (column 1), the time taken to find this plan with the SAT approach (column 2) and with the QBF approach (column 3). We can observe a very clear increase in the difference between the times taken by each approach in

[11] http://minisat.se/.

[12] http://sat.inesc-id.pt/~mikolas/sw/areqs/.

[13] Of course, this threshold 6 on the length of the plan depends strongly on the test conditions, in particular the machine used. The experiments tabulated in Tables 3, 4 and plotted in Fig. 7 were conducted using an Ubuntu 64-bit Linux virtual machine running on an Intel Core i7 processor with 16 gigabytes RAM. However, to evaluate the variations in performance depending on the hardware in which we run the system, we also decided to run the experiments in a secondary machine that uses an Intel Core i5 processor with 16 gigabytes RAM. The results showed that the former outperformed the latest in performance, ranging between 60 and 70% faster.

Table 3 Planning module processing time (in s) for the SAT-based and the QBF-based approach

Plan size	Brute force	QBF
4	0.884	22.475
5	3.719	22.546
6	14.561	23.363
7	61.920	26.337
8	225.077	28.245
9	615.434	30.697

favor of the QBF approach. (Note that the times themselves don't matter much here, and it's the difference that interests us.) In Table 4, we show the times taken by the translation module between (that is, the set of transformations shown in Fig. 3 p. 22) our modal logic on the one hand and the SAT or QBF approach on the other hand (columns 2 and 3). We also present in this table the time used by TouIST to solve the propositional logic formula using MiniSat on the one hand, and to solve the QBF formula using RAReQS on the other hand (columns 4 and 5). Finally, it is important to note that the calculation times presented are the result of the average between 3 successive tests and the difference between tests did not exceed 10% of the average time retained. We are aware of the limited value of these results, but the implementation is for this work just a proof of concept of its feasibility, and we did not seek to optimize the results (which was outside the scope of this work).

8 Conclusion

We have presented a simple logic-based framework for cognitive planning which can be used to endow an artificial agent with the capability of influencing a human agent's beliefs and intentions. We have studied complexity of both satisfiability checking for the logic and the cognitive planning problem. We have shown its potential for application in the HMI domain by formalizing and implementing a HMI scenario in which a human agent and an artificial agent interact through dialogue. We have illustrated the interrelations between the cognitive planning and the belief revision module in this scenario. Our implementation relies on SAT techniques, given the NP-completeness of the satisfiability problem for the epistemic language we consider. We compared the brute force approach to cognitive planning based on a SAT-solver with a competing implementation using a reduction to QBF, relative to the time needed to find a valid plan. According to our tests and their implementation conditions, we observed that the latter beats the former when computing plans with length above a given threshold and, moreover, the improvement in performance is exponential in the length of the plan. Nonetheless, when computing plans of length below the threshold, the brute force approach turns out to be more efficient than the QBF approach. In particular, we recall that the present work does not claim to be an exhaustive comparison between the SAT and the QBF approach.

Future work will be devoted to complement our reasoning and planning model with a machine learning component. We intend to combine our cognitive planning approach with inductive logic programming (ILP) [48], in order to construct an agent's prior

information Σ, as used in the formulation of the cognitive planning problem, through inductive methods. This will allow the persuader to predict the persuadee's beliefs, like in models of theory of mind based on neural networks [57].

Another direction we intend to explore is an extension of our framework by a normative component. The formal language and agent architecture we have presented so far do not include this component. We plan to endow our conversational agent with the capacity to reason about the normative consequences of its behavior and to comply with a presupposed set of ethical and legal norms. To this aim, we will extend our epistemic language by a deontic component. Examples of ethical and legal norms that are relevant for our dialogue scenario are the ethical norm of sincerity (i.e., don't lie, don't assert facts that you do not believe) or the ethical norm of refraining from manipulating or deceiving others. The latter requires a proper understanding by the agent of the subtle distinction between benevolent, harmless forms of persuasion and malevolent, harmful forms that are investigated in some related work on the logical theory of persuasion and manipulation [8, 35].

Appendix: Proofs

Proof of Theorem 2

Proof The proof relies on a previous result proved in [39, Theorem 1]. The result shows that the set of validities for the full epistemic language $\mathcal{L}$ relative to the belief base semantics given in Sect. 4.1 is the same as the set of validities relative to a Kripke-style semantics in which an agent's set of epistemically accessible states at a given state s is a subset of, and not necessarily equal to, the set of states that satisfy all formulas in the agent's belief base at s.

From that result, it is straightforward to show that the belief base semantics for the language $\mathcal{L}_{\mathsf{Frag}}$ is equivalent to a "weaker" semantics exploiting Kripke-like pointed structures of the form (N, s) with $N = (S, \mathcal{B}, \rightsquigarrow, \tau)$ and $s \in S$, where

- S is a non-empty set of states,
- $\mathcal{B} : Agt \times S \longrightarrow 2^{\mathcal{L}_0}$ is a belief base function,
- $\rightsquigarrow \subseteq S \times S$ is agent $\mathfrak{m}$'s epistemic accessibility relation,
- $\tau : Atm \longrightarrow 2^S$ is valuation function,

and with respect to which $\mathcal{L}_{\mathsf{Frag}}$-formulas are interpreted as follows (boolean cases are omitted for simplicity):

$$
\begin{aligned}
(N, s) \vDash p &\Longleftrightarrow s \in \tau(p), \\
(N, s) \vDash \bigtriangleup_i \alpha &\Longleftrightarrow \alpha \in \mathcal{B}(i, s), \\
(N, s) \vDash \Box_{\mathfrak{m}} \alpha &\Longleftrightarrow \forall s' \in S, \text{ if } s \rightsquigarrow s' \text{ then } (N, s') \vDash \alpha.
\end{aligned}
$$

In particular, for every $\varphi \in \mathcal{L}_{\mathsf{Frag}}$, we have that φ is satisfiable in $\mathbf{M}$ iff there exists a pointed structure (N, s) with $N = (S, \mathcal{B}, \rightsquigarrow, \tau)$ and $s \in S$ such that $(N, s) \vDash \varphi$ and

$$
\rightsquigarrow(s) \subseteq \bigcap_{\alpha \in \mathcal{B}(\mathfrak{m}, s)} \| \alpha \|_N, \tag{BG}
$$

with $||\alpha||_N = \{s' \in S : (N, s') \vDash \alpha\}$ and $\rightsquigarrow(s) = \{s' \in S : s \rightsquigarrow s'\}$. We denote by $\mathbf{N}$ the subclass of pointed structures (N, s) satisfying the previous Constraint (BG) and define satisfiability of a formula $\varphi \in \mathcal{L}_{\mathsf{Frag}}$ relative to $\mathbf{N}$ in the usual way.[14]

In the rest of the proof, we are going to show that φ is satisfiable in the class $\mathbf{N}$ if and only if $tr_1(\varphi)$ is satisfiable in the class $\mathbf{K}$.

We first prove the left-to-right direction: if φ is satisfiable in the class $\mathbf{N}$ then $tr_1(\varphi)$ is satisfiable in the class $\mathbf{K}$.

Let $(N, s_0) \in \mathbf{N}$ with $N = (S, \mathcal{B}, \rightsquigarrow, \tau)$ and $s_0 \in S$ such that $(N, s_0) \vDash \varphi$. We build the Kripke model $M = (W, \Rightarrow, \pi)$, as follows:

- $W = S$,
- $\Rightarrow = \rightsquigarrow$,
- $\pi(p) = \tau(p)$ for $p \in Atm$,
- $\pi\left(p_{\triangle_i\alpha}\right) = \{s \in W : \alpha \in \mathcal{B}(i, s)\}$ for $p_{\triangle_i\alpha} \in Atm^+ \setminus Atm$.

By induction on the structure of $\alpha \in \mathcal{L}_0$, it is straightforward to show that

$$\text{for every } s \in S, (N, s) \vDash \alpha \text{ if and only if } (M, s) \vDash tr_0(\alpha). \tag{i}$$

The boolean cases are trivial. As for the case $\alpha = \triangle_i\beta$, we have $(N, s) \vDash \triangle_i\beta$ iff $\beta \in \mathcal{B}(i, s)$ iff $s \in \pi\left(p_{\triangle_i\beta}\right)$ iff $(M, s) \vDash p_{\triangle_i\beta}$ iff $(M, s) \vDash tr_0(\triangle_i\beta)$.

By induction on the structure of φ, we are going to show that $(N, s_0) \vDash \varphi$ if and only if $(M, s_0) \vDash tr_1(\varphi)$. The atomic case and the boolean cases are straightforward and we do not need to prove them. Let us prove the case $\varphi = \triangle_{\mathfrak{m}}\alpha$.

($\Rightarrow$) $(N, s_0) \vDash \triangle_{\mathfrak{m}}\alpha$ means $\alpha \in \mathcal{B}(\mathfrak{m}, s_0)$. By definition of M, the latter implies

$$s_0 \in \pi\left(p_{\triangle_{\mathfrak{m}}\alpha}\right). \tag{ii}$$

Moreover, $\alpha \in \mathcal{B}(\mathfrak{m}, s_0)$ implies $\rightsquigarrow(s_0) \subseteq ||\alpha||_N$ since (N, s_0) satisfies the previous Constraint (BG). By the previous item (i) and the construction of $\Rightarrow$, we have $||\alpha||_N = ||tr_0(\alpha)||_M$ with $||tr_0(\alpha)||_M = \{s' \in W : (M, s') \vDash tr_0(\alpha)\}$. Therefore, by the definition of $\Rightarrow$, it follows that

$$\Rightarrow(s_0) \subseteq ||tr_0(\alpha)||_M. \tag{iii}$$

From the previous items (ii) and (iii), we obtain $(M, s_0) \vDash p_{\triangle_{\mathfrak{m}}\alpha} \wedge \blacksquare tr_0(\alpha)$. Hence, $(M, s_0) \vDash tr_1(\triangle_{\mathfrak{m}}\alpha)$.

($\Leftarrow$) Suppose $(M, s_0) \vDash tr_1(\triangle_{\mathfrak{m}}\alpha)$. The latter means $(M, s_0) \vDash p_{\triangle_{\mathfrak{m}}\alpha} \wedge \blacksquare tr_0(\alpha)$. Hence, by construction of M from N, we have $\alpha \in \mathcal{B}(\mathfrak{m}, s_0)$. The latter means that $(N, s_0) \vDash \triangle_{\mathfrak{m}}\alpha$.

It is just routine to prove the case $\varphi = \triangle_i\alpha$ with $i \neq \mathfrak{m}$.

Finally, let us prove the case $\varphi = \Box_{\mathfrak{m}}\alpha$. $(N, s_0) \vDash \Box_{\mathfrak{m}}\alpha$ iff $\rightsquigarrow(s_0) \subseteq ||\alpha||_N$ iff $\Rightarrow(s_0) \subseteq ||tr_0(\alpha)||_M$ — by the previous item (i) and the construction of $\Rightarrow$ — iff, $(M, s_0) \vDash \blacksquare tr_0(\alpha)$ iff $(M, s_0) \vDash tr_1(\Box_{\mathfrak{m}}\alpha)$.

Thus, $(M, s_0) \vDash tr_1(\varphi)$, since we supposed $(N, s_0) \vDash \varphi$.

We are going to prove the right-to-left direction: if $tr_1(\varphi)$ is satisfiable in the class $\mathbf{K}$ then φ is satisfiable in the class $\mathbf{N}$.

[14] (BG) stands here for "belief groundedness" in the sense that agent $\mathfrak{m}$'s epistemic state $\rightsquigarrow(s)$ at the actual state s is defined from and grounded on its actual belief base $\mathcal{B}(\mathfrak{m}, s)$.

Let $M = (W, \Rightarrow, \pi)$ be a Kripke model and $w_0 \in W$ such that $(M, w_0) \vDash tr_1(\varphi)$. We build the structure $N = (S, \mathcal{B}, \rightsquigarrow, \tau)$ as follows:

- $S = \{w^A : w \in W\} \cup \{w^B : w \in W\}$,
- $\mathcal{B}(i, v^x) = \left\{\alpha \in \mathcal{L}_0 : v \in \pi(p_{\triangle_i \alpha})\right\}$ for $i \neq \mathfrak{m}$ or $v \neq w_0$ or $x \neq A$,
- $\mathcal{B}(\mathfrak{m}, w_0^A) = \left\{\alpha \in \mathcal{L}_0 : w_0 \in \pi(p_{\triangle_{\mathfrak{m}} \alpha}) \text{ and } \Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M\right\}$,
- $\rightsquigarrow = \{(w^A, v^B) : w \Rightarrow v\} \cup \{(w^B, v^B) : w \Rightarrow v\}$,
- $\tau(p) = \{w^A : w \in \pi(p)\} \cup \{w^B : w \in \pi(p)\}$ for $p \in Atm$.

It is easy to verify that, for every $\alpha \in \mathcal{L}_0$, $\left\{v^B \in S : (M, v) \vDash tr_0(\alpha)\right\} = \left\{v^B \in S : (N, v^B) \vDash \alpha\right\}$. Therefore, by construction of $\rightsquigarrow$, $\Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M$ is equivalent to $\rightsquigarrow(w_0^A) \subseteq ||\, tr_0(\alpha) \,||_N$. Again by construction of $\rightsquigarrow$, the latter is equivalent to $\rightsquigarrow(w_0^A) \subseteq ||\, tr_0(\alpha) \,||_N$. Thus,

$$\text{(iv) for every } \alpha \in \mathcal{L}_0, \Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M \text{iff } \rightsquigarrow(w_0^A) \subseteq ||\, tr_0(\alpha) \,||_N.$$

The previous item (iv) guarantees that $(N, w_0^A) \in \mathbf{N}$.

By induction on the structure of φ, we are going to show that $(M, w_0) \vDash tr_1(\varphi)$ if and only if $(N, w_0^A) \vDash \varphi$. The atomic case and the boolean cases are straightforward and we do not need to prove them. Let us prove the case $\varphi = \triangle_{\mathfrak{m}} \alpha$.

$(M, w_0) \vDash tr_1(\triangle_{\mathfrak{m}} \alpha)$ is equivalent to $(M, w_0) \vDash p_{\triangle_{\mathfrak{m}} \alpha} \wedge \blacksquare tr_0(\alpha)$. The latter is equivalent to $w_0 \in \pi(p_{\triangle_{\mathfrak{m}} \alpha})$ and $\Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M$. By definition of $\mathcal{B}(\mathfrak{m}, w_0^A)$, the latter is equivalent to $\alpha \in \mathcal{B}(\mathfrak{m}, w_0^A)$ which in turn is equivalent to $(N, w_0^A) \vDash \triangle_{\mathfrak{m}} \alpha$.

Let us prove the case $\varphi = \triangle_i \alpha$ with $i \neq \mathfrak{m}$.

Suppose $i \neq \mathfrak{m}$. Then, $(M, w_0) \vDash tr_1(\triangle_i \alpha)$ is equivalent to $(M, w_0) \vDash p_{\triangle_i \alpha}$. The latter is equivalent to $w_0 \in \pi(p_{\triangle_i \alpha})$. By definition of $\mathcal{B}(i, w_0^A)$ for $i \neq \mathfrak{m}$, the latter is equivalent to $\alpha \in \mathcal{B}(i, w_0^A)$ which in turn is equivalent to $(N, w_0^A) \vDash \triangle_i \alpha$.

Finally, let us prove the case $\varphi = \Box_{\mathfrak{m}} \alpha$. $(M, w_0) \vDash tr_1(\Box_{\mathfrak{m}} \alpha)$ is equivalent to $(M, w_0) \vDash \blacksquare tr_0(\alpha)$. The latter is in turn equivalent to $\Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M$.

Therefore, by construction of $\rightsquigarrow$, $\Rightarrow(w_0) \subseteq ||\, tr_0(\alpha) \,||_M$ is equivalent to $\rightsquigarrow(w_0^A) \subseteq ||\, tr_0(\alpha) \,||_N$. Again by construction of $\rightsquigarrow$, the latter is equivalent to $\rightsquigarrow(w_0^A) \subseteq ||\, tr_0(\alpha) \,||_N$. The latter means that $(N, w_0^A) \vDash \Box_{\mathfrak{m}} \alpha$.

Thus, $(N, w_0^A) \vDash \varphi$, since we supposed $(M, w_0) \vDash tr_1(\varphi)$. □

Proof of Theorem 4

Proof Suppose we want to check whether a formula $\varphi \in \mathcal{L}_{\mathsf{Frag}}$ is satisfiable. Thanks to Theorems 2 and 3, we just need to check whether $tr_2\Big(tr_1(\varphi), 0, size\big(tr_1(\varphi)\big)\Big) \in \mathcal{L}_{\mathsf{Prop}}$ is satisfiable in propositional logic. The size of $tr_2\Big(tr_1(\varphi), 0, size\big(tr_1(\varphi)\big)\Big)$ is clearly polynomial in the size of φ since, as we have shown above, tr_1 and tr_2 are polynomial translations. We know that satisfiability checking in propositional logic (SAT problem) is NP-complete [62, Theorem 7.37]. Thus, we can conclude that checking satisfiability of formulas in $\mathcal{L}_{\mathsf{Frag}}$ is in NP.

NP-hardness follows from the evident fact that there exists a polysize reduction of SAT to satisfiability checking of $\mathcal{L}_{\mathsf{Frag}}$-formulas. □

Proof of Proposition 1

Proof The second and third equivalences are valid since the binary relation $\rightsquigarrow_{+_i\alpha} \subseteq \mathbf{M} \times \mathbf{M}$ such that $(B, Cxt) \rightsquigarrow_{+_i\alpha} (B', Cxt')$ iff $B' = B^{+_i\alpha}$ and $Cxt' = Cxt$ is functional (i.e., serial and deterministic). We prove the first and fourth equivalence.

Let us start with the first equivalence, case $\alpha' = \bigtriangleup_i\alpha$:

$$\begin{aligned}
(B, Cxt) \vDash [+_i\alpha] \bigtriangleup_i \alpha \quad &\Longleftrightarrow \quad (B^{+_i\alpha}, Cxt) \vDash \bigtriangleup_i\alpha, \\
&\Longleftrightarrow \quad \alpha \in B_i^{+_i\alpha}, \\
&\Longleftrightarrow \quad \alpha \in B_i \cup \{\alpha\}, \\
&\Longleftrightarrow \quad (B, Cxt) \vDash \top.
\end{aligned}$$

The case $\alpha' \neq \bigtriangleup_i\alpha$ is provable by induction on the structure of α' in a straightforward manner.

Let us move to the fourth equivalence, case $i = \mathfrak{m}$:

$$\begin{aligned}
(B, Cxt) \vDash [+_{\mathfrak{m}}\alpha] \Box_{\mathfrak{m}}\alpha' \quad &\Longleftrightarrow \quad (B^{+_{\mathfrak{m}}\alpha}, Cxt) \vDash \Box_{\mathfrak{m}}\alpha', \\
&\Longleftrightarrow \quad \forall B' \in Cxt, \text{ if } B^{+_{\mathfrak{m}}\alpha}\mathcal{R}_{\mathfrak{m}}B' \text{ then } (B', Cxt) \vDash \alpha', \\
&\Longleftrightarrow \quad \forall B' \in Cxt, \text{ if } \big(\forall \beta \in B_{\mathfrak{m}} \cup \{\alpha\}, B' \vDash \beta\big) \text{ then } \\
&\qquad\qquad\qquad\qquad (B', Cxt) \vDash \alpha', \\
&\Longleftrightarrow \quad \forall B' \in Cxt, \text{ if } \big(B\mathcal{R}_{\mathfrak{m}}B' \text{ and } B' \vDash \alpha\big) \text{ then } \\
&\qquad\qquad\qquad\qquad (B', Cxt) \vDash \alpha', \\
&\Longleftrightarrow \quad (B, Cxt) \vDash \Box_{\mathfrak{m}}(\alpha \to \alpha').
\end{aligned}$$

The case case $i \neq \mathfrak{m}$ is analogous:

$$\begin{aligned}
(B, Cxt) \vDash [+_i\alpha] \Box_{\mathfrak{m}}\alpha' \quad &\Longleftrightarrow \quad (B^{+_i\alpha}, Cxt) \vDash \Box_{\mathfrak{m}}\alpha', \\
&\Longleftrightarrow \quad \forall B' \in Cxt, \text{ if } B^{+_i\alpha}\mathcal{R}_{\mathfrak{m}}B' \text{ then } (B', Cxt) \vDash \alpha', \\
&\Longleftrightarrow \quad \forall B' \in Cxt, \text{ if } B\mathcal{R}_{\mathfrak{m}}B' \text{ then } (B', Cxt) \vDash \alpha' \\
&\text{(since } i \neq \mathfrak{m}\text{)}, \\
&\Longleftrightarrow \quad (B, Cxt) \vDash \Box_{\mathfrak{m}}\alpha'.
\end{aligned}$$

This concludes the proof. □

Proof of Proposition 2

Proof The proof is by induction on the structure of the formula φ.

Cases $\varphi = p$ and $\varphi = \bigtriangleup_i\alpha$ are evident.

Case $\varphi = \neg\psi$. $\neg\psi \leftrightarrow red(\neg\psi)$ is equal to $\neg\psi \leftrightarrow \neg red(\psi)$. The latter is logically equivalent to $\psi \leftrightarrow red(\psi)$, i.e., for every $(B, Cxt) \in \mathbf{M}$, $(B, Cxt) \models \neg\psi \leftrightarrow \neg red(\psi)$ iff $(B, Cxt) \models \psi \leftrightarrow red(\psi)$. By induction hypothesis, the latter is valid. Therefore, $\neg\psi \leftrightarrow red(\neg\psi)$ is valid. Furthermore, $red(\neg\psi)$ is equal to $\neg red(\psi)$ and, by induction hypothesis $red(\psi) \in \mathcal{L}_{\mathsf{Frag}}$. Thus, $red(\neg\psi) \in \mathcal{L}_{\mathsf{Frag}}$.

Case $\varphi = \psi_1 \wedge \psi_2$ can be proved in analogous way.

Case $\varphi = \Box_{\mathfrak{m}}\alpha$. $\Box_{\mathfrak{m}}\alpha \leftrightarrow red(\Box_{\mathfrak{m}}\alpha)$ is equal to $\Box_{\mathfrak{m}}\alpha \leftrightarrow \Box_{\mathfrak{m}} red(\alpha)$. By structural induction, it is easy to prove the following useful proposition.

Proposition 8 *For every* $\alpha \in \mathcal{L}_0$, $red(\alpha) = \alpha$.

Thus, by Proposition 8, $\Box_{\mathfrak{m}}\alpha \leftrightarrow \Box_{\mathfrak{m}} red(\alpha)$ is equal to $\Box_{\mathfrak{m}}\alpha \leftrightarrow \Box_{\mathfrak{m}}\alpha$ which is valid. Hence, $\Box_{\mathfrak{m}}\alpha \leftrightarrow red(\Box_{\mathfrak{m}}\alpha)$ is valid. Clearly, $red(\Box_{\mathfrak{m}}\alpha) \in \mathcal{L}_{\mathsf{Frag}}$.

Case $\varphi = [+_i\alpha]\psi$. We prove this case by induction on the structure of ψ.

Subcase $\psi = \bigtriangleup_i\alpha$. $[+_i\alpha]\bigtriangleup_i\alpha \leftrightarrow red\big([+_i\alpha]\bigtriangleup_i\alpha\big)$ is equal to $[+_i\alpha]\bigtriangleup_i\alpha \leftrightarrow \top$. By Proposition 1 (first equivalence, first case), the latter is valid. Hence, $[+_i\alpha]\bigtriangleup_i\alpha \leftrightarrow red\big([+_i\alpha]\bigtriangleup_i\alpha\big)$ is valid. Clearly, $red([+_i\alpha]\bigtriangleup_i\alpha) \in \mathcal{L}_{\mathsf{Frag}}$ since $red([+_i\alpha]\bigtriangleup_i\alpha) = \top$ and $\top \in \mathcal{L}_{\mathsf{Frag}}$.

Subcase $\psi = \alpha'$ with $\alpha' \neq \bigtriangleup_i\alpha$. $[+_i\alpha]\alpha' \leftrightarrow red\big([+_i\alpha]\alpha'\big)$ is equal to $[+_i\alpha]\alpha' \leftrightarrow red\big(\alpha'\big)$ which is equal to $[+_i\alpha]\alpha' \leftrightarrow \alpha'$, by Proposition 8. By Proposition 1 (first equivalence, second case), the latter is valid. Hence, $[+_i\alpha]\alpha' \leftrightarrow red\big([+_i\alpha]\alpha'\big)$ with $\alpha' \neq \bigtriangleup_i\alpha$ is valid. Clearly, $red([+_i\alpha]\alpha') \in \mathcal{L}_{\mathsf{Frag}}$ since, by Proposition 8, $red([+_i\alpha]\alpha') = [+_i\alpha]\alpha'$ and $[+_i\alpha]\alpha' \in \mathcal{L}_{\mathsf{Frag}}$.

Subcase $\psi = \neg\chi$. $[+_i\alpha]\neg\chi \leftrightarrow red\big([+_i\alpha]\neg\chi\big)$ is equal to $[+_i\alpha]\neg\chi \leftrightarrow red\big(\neg[+_i\alpha]\chi\big)$ which is equal to $[+_i\alpha]\neg\chi \leftrightarrow \neg red\big([+_i\alpha]\chi\big)$. The latter is logically equivalent to $\neg[+_i\alpha]\neg\chi \leftrightarrow red\big([+_i\alpha]\chi\big)$. By Proposition 1 (second equivalence), $\neg[+_i\alpha]\neg\chi \leftrightarrow [+_i\alpha]\chi$ is valid. Therefore, $\neg[+_i\alpha]\neg\chi \leftrightarrow red\big([+_i\alpha]\chi\big)$ is logically equivalent to $[+_i\alpha]\psi \leftrightarrow red\big([+_i\alpha]\psi\big)$. By induction hypothesis, the latter is valid. It follows that $[+_i\alpha]\neg\chi \leftrightarrow red\big([+_i\alpha]\neg\chi\big)$ is valid too. Clearly, $red\big([+_i\alpha]\neg\chi\big) \in \mathcal{L}_{\mathsf{Frag}}$ since $red\big([+_i\alpha]\neg\chi\big) = red\big(\neg[+_i\alpha]\chi\big) = \neg red\big([+_i\alpha]\chi\big)$ and, by induction hypothesis $red\big([+_i\alpha]\chi\big) \in \mathcal{L}_{\mathsf{Frag}}$.

Subcase $\psi = \chi_1 \wedge \chi_2$ is proved in an analogous way.

Subcase $\psi = \Box_{\mathfrak{m}}\alpha'$ and $i = \mathfrak{m}$. $[+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow red\big([+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha'\big)$ is equal to $[+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow red\big(\Box_{\mathfrak{m}}(\alpha \to \alpha')\big)$ which is equal to $[+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow \Box_{\mathfrak{m}} red(\alpha \to \alpha')$. By Proposition 8, the latter is equal to $[+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow \Box_{\mathfrak{m}}(\alpha \to \alpha')$ which in turn, by Proposition 1 (fourth equivalence), is valid. Thus, $[+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha' \leftrightarrow red\big([+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha'\big)$ is valid too. Clearly, $red\big([+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha'\big) \in \mathcal{L}_{\mathsf{Frag}}$ since $red\big([+_{\mathfrak{m}}\alpha]\Box_{\mathfrak{m}}\alpha'\big) = red\big(\Box_{\mathfrak{m}}(\alpha \to \alpha')\big) = \Box_{\mathfrak{m}} red(\alpha \to \alpha')$ and, by Proposition 8, $\Box_{\mathfrak{m}} red(\alpha \to \alpha') \in \mathcal{L}_{\mathsf{Frag}}$.

Subcase $\psi = \Box_{\mathfrak{m}}\alpha'$ and $i \neq \mathfrak{m}$ can be proved in an analogous way.

Subcase $\psi = [+_j\beta]\chi$. By induction hypothesis, we have that $[+_j\beta]\chi \leftrightarrow red([+_j\beta]\chi)$ is valid and $red([+_j\beta]\chi) \in \mathcal{L}_{\mathsf{Frag}}$. Thus, $[+_j\beta]\chi$ and $red([+_j\beta]\chi)$ are logically equivalent. Hence, $[+_i\alpha][+_j\beta]\chi$ and $[+_i\alpha]red([+_j\beta]\chi)$ are logically equivalent too.

By induction hypothesis, since $red([+_j\beta]\chi) \in \mathcal{L}_{\mathsf{Frag}}$, we also have that $[+_i\alpha]red([+_j\beta]\chi) \leftrightarrow red\big([+_i\alpha]red([+_j\beta]\chi)\big)$ is valid. Thus, $[+_i\alpha]red([+_j\beta]\chi)$ and $red\big([+_i\alpha]red([+_j\beta]\chi)\big)$ are logically equivalent.

Therefore, we can conclude that $[+_i\alpha][+_j\beta]\chi$ and $red\big([+_i\alpha]red([+_j\beta]\chi)\big)$ are logically equivalent too and, moreover, $[+_i\alpha][+_j\beta]\chi \leftrightarrow red\big([+_i\alpha]red([+_j\beta]\chi)\big)$ is valid. Hence, $[+_i\alpha][+_j\beta]\chi \leftrightarrow red([+_i\alpha][+_j\beta]\chi)$ is also valid since it is equal to the latter.

Checking that $red([+_i\alpha][+_j\beta]\chi) \in \mathcal{L}_{\mathsf{Frag}}$ is easy. We have $red([+_i\alpha][+_j\beta]\chi) = red\big([+_i\alpha]red([+_j\beta]\chi)\big)$ and, by induction hypothesis, $red([+_j\beta]\chi) \in \mathcal{L}_{\mathsf{Frag}}$. Thus, again by induction hypothesis, $red\big([+_i\alpha]red([+_j\beta]\chi)\big) \in \mathcal{L}_{\mathsf{Frag}}$. Hence, $red([+_i\alpha][+_j\beta]\chi) \in \mathcal{L}_{\mathsf{Frag}}$. □

Proof of Proposition 4

Proof We first prove the left-to-right direction, namely, that $\Sigma \models_{\mathbf{M}} \varphi$ implies $\models_{\mathbf{M}} \big(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\big) \rightarrow \varphi$, with $\varphi \in \mathcal{L}^{+}_{\mathsf{Frag}}$ and $\Sigma \subset \mathcal{L}_0$. Suppose $\Sigma \models_{\mathbf{M}} \varphi$ and, towards a contradiction, $\not\models_{\mathbf{M}} \big(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\big) \rightarrow \varphi$. The latter means that we can find $(B, Cxt) \in \mathbf{M}$ with $B = (B_1, \ldots, B_n, V)$ such that $(B, Cxt) \models \bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha$ and $(B, Cxt) \not\models \varphi$. We define $Cxt' = \big(Cxt \cap \mathcal{R}_i(B)\big)$, with $\mathcal{R}_i(B) = \{B' \in \mathbf{S} : B\mathcal{R}_iB'\}$. Since $(B, Cxt) \models \bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha$, it is straightforward to verify that $Cxt' \subseteq \mathbf{S}(\Sigma)$. Moreover, by induction on the structure of φ and the fact that $(B, Cxt) \not\models \varphi$, it is routine to verify that $(B, Cxt') \not\models \varphi$ which contradicts the initial assumption $\Sigma \models_{\mathbf{M}} \varphi$.

Let us prove the right-to-left direction. Suppose $\models_{\mathbf{M}} \big(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\big) \rightarrow \varphi$ and, towards a contradiction, $\Sigma \not\models_{\mathbf{M}} \varphi$. The latter means that that we can find $(B, Cxt) \in \mathbf{M}$ with $B = (B_1, \ldots, B_n, V)$ such that $Cxt \subseteq \mathbf{S}(\Sigma)$ and $(B, Cxt) \not\models \varphi$. Since $Cxt \subseteq \mathbf{S}(\Sigma)$, clearly $(B, Cxt) \models \bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha$. Thus, $(B, Cxt) \models \bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha \wedge \neg\varphi$ which contradicts the initial assumption. □

Proof of Theorem 6

Proof By Proposition 5, a cognitive planning problem $\langle \Sigma, Op, \alpha_G\rangle$ has a solution plan if and only if it has a poly-size solution plan. Consider a poly-time non-deterministic Turing machine with an NP-oracle (Σ_2^{P}-Turing machine). It begins with an empty plan and branches over all poly-size plans of length $k \leq |Op|$ choosing non deterministically operators to add to the plan. It accepts if $\Sigma \models_{\mathbf{M}} \langle\!\langle \epsilon_1 \rangle\!\rangle \ldots \langle\!\langle \epsilon_k \rangle\!\rangle \Box_{\mathrm{m}}\alpha_G$ i.e., using Proposition 4, if $\neg\big(\big(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\big) \rightarrow \langle\!\langle \epsilon_1 \rangle\!\rangle \ldots \langle\!\langle \epsilon_k \rangle\!\rangle \Box_{\mathrm{m}}\alpha_G\big)$ is unsatisfiable in the class $\mathbf{M}$. Thanks to Theorem 5, unsatisfiability of this $\mathcal{L}^{+}_{\mathsf{Frag}}$ formula can be checked by the NP-oracle. When $k = |Op|$ and the formula is satisfiable, the Turing machine rejects. □

Proof of Theorem 7

Proof It is well known that checking satisfiability of a $\exists\forall$ QBF is Σ_2^{P}-hard as it is possible to simulate an alternating Turing machine in polynomial time with 2 alternations

and starting in an existential state, that decides all the problems in the class Σ_2^{P}. Let $\psi = \exists x_1 \dots \exists x_n \forall y_1 \dots \forall y_m \varphi(x_1, \dots, x_n, y_1, \dots, y_m)$ be a quantified boolean formula (QBF) in prenex normal form. We consider the cognitive planning problem $\langle \Sigma, Op, \alpha_G \rangle$ where:

$$\begin{aligned}
\Sigma &= \{\neg \triangle_{\mathfrak{h}} x_i \vee \neg \triangle_{\mathfrak{h}} \neg x_i : i \in \{1, \dots, n\}\} \\
Op &= \{+_{\mathfrak{m}} \triangle_{\mathfrak{h}} x_i, +_{\mathfrak{m}} \triangle_{\mathfrak{h}} \neg x_i : i \in \{1, \dots, n\}\} \\
\mathcal{P}(+_{\mathfrak{m}} \triangle_{\mathfrak{h}} x_i) &= \top \text{ for all } i \in \{1, \dots, n\} \\
\mathcal{P}(+_{\mathfrak{m}} \triangle_{\mathfrak{h}} \neg x_i) &= \top \text{ for all } i \in \{1, \dots, n\} \\
\alpha_G &= \bigwedge_{i \in \{1,\dots,n\}} \big(\triangle_{\mathfrak{h}} x_i \vee \triangle_{\mathfrak{h}} \neg x_i\big) \wedge \mathit{encode}\big(\varphi(x_1, \dots, x_n, y_1, \dots, y_m)\big)
\end{aligned}$$

where

$$\begin{aligned}
\mathit{encode}(x_i) &= \triangle_{\mathfrak{h}} x_i \\
\mathit{encode}(\neg x_i) &= \triangle_{\mathfrak{h}} \neg x_i \\
\mathit{encode}(y_i) &= \triangle_{\mathfrak{h}} y_i \\
\mathit{encode}(\neg y_i) &= \neg \triangle_{\mathfrak{h}} y_i \\
\mathit{encode}(\varphi_1 \wedge \varphi_2) &= \mathit{encode}(\varphi_1) \wedge \mathit{encode}(\varphi_2) \\
\mathit{encode}(\varphi_1 \vee \varphi_2) &= \mathit{encode}(\varphi_1) \vee \mathit{encode}(\varphi_2)
\end{aligned}$$

We want to prove that $\langle \Sigma, Op, \alpha_G \rangle$ has a solution plan if and only if ψ is true.

If $\langle \Sigma, Op, \alpha_G \rangle$ has a solution plan, then by Proposition 5, it has a poly-size solution plan $P = \epsilon_1, \dots, \epsilon_k$ with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$. It is easily seen that for each $i \in \{1, \dots, n\}$, exactly one action of either $+_{\mathfrak{m}} \triangle_{\mathfrak{h}} x_i$ or $+_{\mathfrak{m}} \triangle_{\mathfrak{h}} \neg x_i$ is in the plan P. Indeed on the one hand, at most one of these actions is in the plan because $\neg \triangle_{\mathfrak{h}} x_i \vee \neg \triangle_{\mathfrak{h}} \neg x_i \in \Sigma$. And on the other hand, at least one is in the plan because of the goal $\triangle_{\mathfrak{h}} x_i \vee \triangle_{\mathfrak{h}} \neg x_i$.

Then, we can match each solution plan candidate for $\langle \Sigma, Op, \alpha_G \rangle$ to a valuation v of propositional variables in $\{x_1, \dots, x_n\}$. Hence, a solution plan candidate for $\langle \Sigma, Op, \alpha_G \rangle$ is given by $\langle\!\langle \epsilon_1 \rangle\!\rangle \dots \langle\!\langle \epsilon_n \rangle\!\rangle$, with for all $i \in \{1, \dots, n\}$:

$$\epsilon_i = \begin{cases} +_{\mathfrak{m}} \triangle_{\mathfrak{h}} \neg x_i & \text{if } v \nvDash x_i \\ +_{\mathfrak{m}} \triangle_{\mathfrak{h}} x_i & \text{if } v \vDash x_i \end{cases}$$

Such a plan is a solution plan if and only if $\Sigma \vDash_{\mathbf{M}} \langle\!\langle \epsilon_1 \rangle\!\rangle \dots \langle\!\langle \epsilon_n \rangle\!\rangle \Box_{\mathfrak{m}} \alpha_G$ which can be also written as:

$$\Sigma \vDash_{\mathbf{M}} \left(\bigwedge_{i \in \{1,\dots,n\}} [+_{\mathfrak{m}} \alpha_{\epsilon_1}] \dots [+_{\mathfrak{m}} \alpha_{\epsilon_{i-1}}] \mathcal{P}(\epsilon_i) \right) \wedge \left([+_{\mathfrak{m}} \alpha_{\epsilon_1}] \dots [+_{\mathfrak{m}} \alpha_{\epsilon_{n-1}}] [+_{\mathfrak{m}} \alpha_{\epsilon_n}] \Box_{\mathfrak{m}} \alpha_G \right)$$

Given that, on the one hand $\mathcal{P}(\epsilon_i) = \top$ for all $i \in \{1, \dots, n\}$ and $\mathit{red}([+_{\mathfrak{m}} \alpha] \top) = \top$, and on the other hand $\mathit{red}([+_{\mathfrak{m}} \alpha] \Box_{\mathfrak{m}} \alpha') = \mathit{red}\big(\Box_{\mathfrak{m}} (\alpha \to \alpha')\big) = \Box_{\mathfrak{m}} \big(\mathit{red}(\neg \alpha \vee \alpha')\big)$, when applying recursively the reduction from $\mathcal{L}^{+}_{\mathsf{Frag}}$ to $\mathcal{L}_{\mathsf{Frag}}$ we obtain:

$$red\big(\langle\!\langle \epsilon_1 \rangle\!\rangle \dots \langle\!\langle \epsilon_n \rangle\!\rangle \Box_{\mathfrak{m}} \alpha_G\big) = \Box_{\mathfrak{m}} \Bigg(\Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \not\Vdash x_i}} \neg \triangle_{\mathfrak{h}} \neg x_i \Big) \vee \Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \Vdash x_i}} \neg \triangle_{\mathfrak{h}} x_i \Big) \vee \alpha_G \Bigg)$$

Then, by Propositions 2 and 4, $\Sigma \models_{\mathbf{M}} \langle\!\langle \epsilon_1 \rangle\!\rangle \dots \langle\!\langle \epsilon_n \rangle\!\rangle \Box_{\mathfrak{m}} \alpha_G$ if and only if the following formula $\Phi \in \mathcal{L}_{\mathsf{Frag}}$ is valid in class $\mathbf{M}$:

$$\Phi = \Big(\bigwedge_{\alpha \in \Sigma} \Box_{\mathfrak{m}} \alpha \Big) \to \Box_{\mathfrak{m}} \Bigg(\Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \not\Vdash x_i}} \neg \triangle_{\mathfrak{h}} \neg x_i \Big) \vee \Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \Vdash x_i}} \neg \triangle_{\mathfrak{h}} x_i \Big) \vee \alpha_G \Bigg)$$

That is verified, by Theorems 2 and 3, if and only if the following formula $F \in \mathcal{L}_{\mathsf{Prop}}$, resulting from the successive applications of tr_1 then tr_2 to the previous one, is valid in propositional logic:

$$\begin{aligned} F = & \Big(\bigwedge_{i \in \{1,\dots,n\}} \bigwedge_{0 \le z \le y} \big(r_{0,z} \to (\neg p_{\triangle_{\mathfrak{h}} x_{i(z)}} \vee \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z)}}) \big) \Big) \to \\ & \bigwedge_{0 \le z \le y} \Bigg[r_{0,z} \to \Bigg[\Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \not\Vdash x_i}} \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z)}} \Big) \vee \Big(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \Vdash x_i}} \neg p_{\triangle_{\mathfrak{h}} x_{i(z)}} \Big) \vee \dots \\ & \dots \Big(\Big(\bigwedge_{i \in \{1,\dots,n\}} \big(p_{\triangle_{\mathfrak{h}} x_{i(z)}} \vee p_{\triangle_{\mathfrak{h}} \neg x_{i(z)}} \big) \Big) \wedge encode_2\big(\varphi(x_1,\dots,x_n,y_1,\dots,y_m), z\big) \Big) \Bigg] \Bigg] \end{aligned}$$

where $y = size(tr_1(\Phi))$, and

$$\begin{aligned} encode_2(x_i, z) &= p_{\triangle_{\mathfrak{h}} x_{i(z)}} \\ encode_2(\neg x_i, z) &= p_{\triangle_{\mathfrak{h}} \neg x_{i(z)}} \\ encode_2(y_i, z) &= p_{\triangle_{\mathfrak{h}} y_{i(z)}} \\ encode_2(\neg y_i, z) &= \neg p_{\triangle_{\mathfrak{h}} y_{i(z)}} \\ encode_2(\varphi_1 \wedge \varphi_2, z) &= encode_2(\varphi_1, z) \wedge encode_2(\varphi_2, z) \\ encode_2(\varphi_1 \vee \varphi_2, z) &= encode_2(\varphi_1, z) \vee encode_2(\varphi_2, z) \end{aligned}$$

We consider partial valuations of only propositional variables $\{r_{0,z} : 0 \le z \le y\}$ in F with the k variables $\{r_{0,z_i} : 0 \le i < k\}$ being true, the other ones being false. Without loss of generality, we can choose any $z_k \in \{0,\dots,y\} \backslash \{z_i : 0 \le i < k\}$ to obtain a new partial valuation $\{r_{0,z_i} : 0 \le i < k+1\}$ from the latter by switching the truth value of one propositional variable r_{0,z_k} from false to true, and repeat this process until $k = y + 1$.

Let $F_k = A_k \to B_k$ where $A_0 = B_0 = \top$ and

$$A_{k+1} = A_k \wedge \bigwedge_{i\in\{1,\dots,n\}} \left(\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}} \vee \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\right)$$

$$B_{k+1} = B_k \wedge \Bigg[\Bigg(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v(x_i)=0}} \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\Bigg) \vee \dots$$

$$\dots \Bigg(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v(x_i)=1}} \neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}}\Bigg) \vee \dots$$

$$\dots \Bigg(\Big(\bigwedge_{i\in\{1,\dots,n\}} \left(p_{\triangle_{\mathfrak{h}} x_{i(z_k)}} \vee p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\right)\Big) \wedge \dots$$

$$\dots encode_2\big(\varphi(x_1,\dots,x_n,y_1,\dots,y_m), z_k\big)\Bigg)\Bigg]$$

It is easily seen that for a given partial valuation $\{r_{0,z_i} : 0 \le i < k\}$ we can simplify conjunctions on the left part and the right part of the implication in F to obtain F_k. Hence, the truth value of F and F_k is the same for this partial valuation.

Let us now prove that any complete valuation V is a model of F (i.e. F is valid in propositional logic) if and only if given the partial valuation v of propositional variables in $\{x_1,\dots,x_n\}$ and for any valuation of propositional variables in $\{y_1,\dots,y_m\}$ we have $\varphi(x_1,\dots,x_n,y_1,\dots,y_m)$ is true (i.e. the QBF ψ is true). The proof is by induction on the number k of true propositional variables from $\{r_{0,z} : 0 \le z \le y\}$ in the valuation V.

Case $k = 0$ is evident as $F_0 = \top \to \top$.

Case $k + 1$: Assume that F_k is true for a given partial valuation with k true propositional variables $\{r_{0,z_i} : 0 \le i < k\}$.

On the one hand, if A_k is false, then A_{k+1} is also false and F_{k+1} is true.

On the other hand, if A_k is true, then B_k is true. In this case, if A_{k+1} is false, F_{k+1} is true again. Suppose that A_{k+1} is true, then we have $\bigwedge_{i\in\{1,\dots,n\}} \left(\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}} \vee \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\right)$ is true. Hence, we have $\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}} \vee \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}$ is true for any $i \in \{1,\dots,n\}$. Then, at least one of the three following cases holds:

(1) $\neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}$ is true for at least one x_i such that $v \nvDash x_i$, then the disjunct $\left(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \nvDash x_i}} \neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\right)$ is true and B_{k+1} and F_{k+1} are true.

(2) $\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}}$ is true for at least one x_i such that $v \vDash x_i$, then the disjunct $\left(\bigvee_{\substack{i \in \{1,\dots,n\} \\ v \vDash x_i}} \neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}}\right)$ is true and B_{k+1} and F_{k+1} are true.

(3) $\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}}$ is true and $\neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}$ is false for all x_i such that $v \nvDash x_i$, and $\neg p_{\triangle_{\mathfrak{h}} x_{i(z_k)}}$ is false and $\neg p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}$ is true for all x_i such that $v \vDash x_i$, then the disjunct $\left(\left(\bigwedge_{i\in\{1,\dots,n\}} \left(p_{\triangle_{\mathfrak{h}} x_{i(z_k)}} \vee p_{\triangle_{\mathfrak{h}} \neg x_{i(z_k)}}\right)\right) \wedge encode_2\big(\varphi(x_1,\dots,x_n,y_1,\dots,y_m), z_k\big)\right)$ is true and B_{k+1} and F_{k+1} are true if and only if $\varphi(x_1,\dots,x_n,y_1,\dots,y_m)$ is true for the partial valuation v of propositional variables in $\{x_1,\dots,x_n\}$ completed by any valuation of propositional variables in $\{y_1,\dots,y_m\}$.

Indeed, $p_{\triangle_{\mathfrak{h}}\neg x_{i(z_k)}}$ is true for all x_i such that $v \nvDash x_i$ and $p_{\triangle_{\mathfrak{h}}x_{i(z_k)}}$ is true for all x_i such that $v \vDash x_i$. On the one hand, $\bigwedge_{i\in\{1,\dots,n\}} \left(p_{\triangle_{\mathfrak{h}}x_{i(z_k)}} \vee p_{\triangle_{\mathfrak{h}}\neg x_{i(z_k)}}\right)$ is true. On the other hand, we compare the truth values of $encode_2\left(\varphi(x_1,\dots,x_n,y_1,\dots,y_m),z_k\right)$ and $\varphi(x_1,\dots,x_n,y_1,\dots,y_m)$. For all $i \in \{1,\dots,n\}$ the truth values of propositional variables $p_{\triangle_{\mathfrak{h}}x_{i(z_k)}}$ and $p_{\triangle_{\mathfrak{h}}\neg x_{i(z_k)}}$ match respectively with the truth values of litterals x_i and $\neg x_i$ given by valuation v. Moreover, for all $i \in \{1,\dots,m\}$ the truth value of propositional variable $p_{\triangle_{\mathfrak{h}}y_{i(z_k)}}$ matches with the truth value of propositional variable y_i. Then $encode_2\left(\varphi(x_1,\dots,x_n,y_1,\dots,y_m),z_k\right)$ is true for any complete valuation of variables $\{p_{\triangle_{\mathfrak{h}}y_{i(z_k)}} : 1 \le i \le m\}$ if and only if $\varphi(x_1,\dots,x_n,y_1,\dots,y_m)$ is true for the partial valuation v of propositional variables in $\{x_1,\dots,x_n\}$ completed by any valuation of propositional variables in $\{y_1,\dots,y_m\}$.

We can conclude that F is true for any valuation, and then F is valid in propositional logic, if and only if v is a valuation of existentially quantified variables for which ψ is true. Then $\langle \Sigma, Op, \alpha_G\rangle$ has a solution plan if and only if ψ is true. This proves that deciding plan existence for $\langle \Sigma, Op, \alpha_G\rangle$ is Σ_2^{P}-hard. □

Proof of Proposition 6

Proof Let a formula $\varphi \in \mathcal{L}^{sel}_{\mathsf{Frag}}$ and a valuation $V_{sel} \subseteq Atm^{sel}$ of selectors of φ. The constants $\bot$ and $\top$ are respectively represented in $\mathcal{L}^{sel}_{\mathsf{Frag}}$ by $(p \wedge \neg p)$ and $\neg(p \wedge \neg p)$ for one propositional variable p. Without loss of generality, we can consider that $size(\bot) = size(\top) = 1$, because these constants can be represented instead by fresh propositional variables $p_\bot$ and $p_\top$ adding two conjuncts to formula $\varphi' = \varphi[p_\bot/\bot][p_\top/\top]$ such that $\psi = \varphi' \wedge \left(p_\bot \leftrightarrow (p \wedge \neg p)\right) \wedge \left(p_\top \leftrightarrow \neg(p \wedge \neg p)\right)$ and φ are equisatisfiable. Under this assumption, we prove by induction on the structure of φ' that $A_{\varphi'} = B_{\varphi'}$,

$$\text{where} \quad \begin{cases} A_{\varphi'} = size\Big(tr_1\big(\sigma_1(V_{sel},\varphi')\big)\Big) \\ B_{\varphi'} = size\big(tr_1^{sel}(\varphi')\big) \end{cases}$$

Indeed, in the case $\varphi' = s$, on the one hand $B_s = size\big(tr_1^{sel}(s)\big) = 1$. On the other hand, when $s \in V_{sel}$ we have $A_s = size\Big(tr_1\big(\sigma_1(V_{sel},s)\big)\Big) = size(\top) = 1$, and when $s \notin V_{sel}$ we have $A_s = size\Big(tr_1\big(\sigma_1(V_{sel},s)\big)\Big) = size(\bot) = 1$. All other induction cases are evident. Given this result for φ', is it easily seen that $A_\psi = B_\psi$.

Let now prove that the following syntactic equality between formulas:

$$tr_2\Big(tr_1\big(\sigma_1(V_{sel},\psi)\big),0,A_\psi\Big) = \sigma_2\Big(V_{sel}, tr_2^{sel}\Big(tr_1^{sel}(\psi),0,B_\psi\Big)\Big)$$

Again, it is easily seen that it suffices to prove this result for the formula φ'. The proof is by induction on the structure of φ'.

Cases $\varphi' = p$, $\varphi' = \bigtriangleup_i \alpha$, $\varphi' = s$, $\varphi' = \neg\varphi_0$ and $\varphi' = \varphi_1 \wedge \varphi_2$ are evident.

Case $\varphi' = \Box_{\mathfrak{m}}\beta$ is verified because $A_\psi = B_\psi$. Indeed, we have the following syntactic equality between formulas:

$$\bigwedge_{0\leq z\leq A_\psi}\Big(r_{x,z}\rightarrow tr_2(\omega,z,A_\psi)\Big)=\sigma_2\Big(V_{sel},\bigwedge_{0\leq z\leq B_\psi}\Big(r_{x,z}\rightarrow tr_2^{sel}(\omega,z,B_\psi)\Big)\Big)$$

In conclusion, as we have the result of syntactical equality for ψ, we have the expected result of equisatisfiability for φ. □

Proof of Proposition 7

Proof Let $P=\epsilon_1,\ldots,\epsilon_k$ a sequence of actions such that $k\leq|Op|$ and $\epsilon_i\neq\epsilon_j$ for all $i<j$. We consider the valuation V_{sel} of selectors in Sel_{Op} such that P is the corresponding designated plan. First, we are going to prove the three following properties:

a. $\sigma_1(V_{sel},\varphi_{Sel_{Op}})\equiv\top$

b. $\forall i\in\{1,\ldots,k\}:\sigma_1\Big(V_{sel},\Pi_{\preceq\epsilon_i}\big(\mathcal{P}(\epsilon_i)\big)\Big)\equiv red\Big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\Big)$

c. $\sigma_1(V_{sel},\varphi_P)\equiv red\big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{k-1}}][+_{\mathrm{m}}\alpha_{\epsilon_k}]\Box_{\mathrm{m}}\alpha_G\big)$

Property (a.) is trivially verified because the designated plan is a particular plan candidate, and then it corresponds to a model of $\varphi_{Sel_{Op}}$.

We prove property (b.) by induction on the structure of $\mathcal{P}(\epsilon_i)\in\mathcal{L}_{\mathsf{Frag}}$. We give here only the explicit and implicit belief cases. Other cases propagate the property trivially following the inductive definitions of σ_1, $\Pi_{\preceq\epsilon}$ and *red*.

- if $\mathcal{P}(\epsilon_i)=\bigtriangleup_{\mathrm{m}}\alpha_{\epsilon'}$ then
  - when $\epsilon'=\epsilon_j$ with $j<i$ we have $\sigma_1(V_{sel},s_{\epsilon'\preceq\epsilon_i})=\top$ and then $\sigma_1\Big(V_{sel},\Pi_{\preceq\epsilon_i}\big(\mathcal{P}(\epsilon_i)\big)\Big)$ $=\sigma_1\Big(V_{sel},s_{\epsilon'\preceq\epsilon_i}\vee\bigtriangleup_{\mathrm{m}}\alpha_{\epsilon'}\Big)\equiv\top=red\Big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\Big)$
  - when $\epsilon'=\epsilon_j$ with $j\geq i$, or $\epsilon'\in Op\backslash P$, we have $\sigma_1(V_{sel},s_{\epsilon'\preceq\epsilon_i})=\bot$ and then $\sigma_1\Big(V_{sel},\Pi_{\preceq\epsilon_i}\big(\mathcal{P}(\epsilon_i)\big)\Big)=\sigma_1\Big(V_{sel},s_{\epsilon'\preceq\epsilon_i}\vee\bigtriangleup_{\mathrm{m}}\alpha_{\epsilon'}\Big)\equiv\bigtriangleup_{\mathrm{m}}\alpha_{\epsilon'}=red\Big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\Big)$
- if $\mathcal{P}(\epsilon_i)=\Box_{\mathrm{m}}\alpha$ then $\sigma_1\Big(V_{sel},\Pi_{\preceq\epsilon_i}\big(\mathcal{P}(\epsilon_i)\big)\Big)=\sigma_1\Big(V_{sel},\Box_{\mathrm{m}}\Big(\Big(\bigvee_{\substack{\epsilon'\in Op\\ \epsilon_i\neq\epsilon'}}s_{\epsilon'\preceq\epsilon_i}\wedge\neg\alpha_{\epsilon'}\Big)\vee\alpha\Big)\Big)$

  $\equiv\Box_{\mathrm{m}}\Big(\bigvee_{j\in\{1,\ldots,i-1\}}\neg\alpha_{\epsilon_j}\vee\alpha\Big)=red\Big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\Big)$

Property (c.) is proved as followed:

$$\begin{aligned}\sigma_1(V_{sel},\varphi_P)&=\sigma_1\Big(V_{sel},\Box_{\mathrm{m}}\Big(\Big(\bigvee_{\epsilon\in Op}s_{\epsilon\preceq\epsilon}\wedge\neg\alpha_\epsilon\Big)\vee\alpha_G\Big)\Big)\\&\equiv\Box_{\mathrm{m}}\Big(\Big(\bigvee_{i\in\{1,\ldots,k\}}\neg\alpha_{\epsilon_i}\Big)\vee\alpha_G\Big)\\&=red\big([+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{k-1}}][+_{\mathrm{m}}\alpha_{\epsilon_k}]\Box_{\mathrm{m}}\alpha_G\big)\end{aligned}$$

Finally, from the properties (a.), (b.) and (c.), and the inductive definitions of σ_1 and *red*, we have:

$$\sigma_1(V_{sel}, \varphi_{\langle\Sigma,Op,\alpha_G\rangle}) = \left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \sigma_1\left(V_{sel}, \bigwedge_{\epsilon\in Op}\left(s_{\epsilon\preceq\epsilon} \rightarrow \Pi_{\preceq\epsilon}(\mathcal{P}(\epsilon))\right) \wedge \varphi_P\right)$$

$$\equiv \left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \bigwedge_{i\in\{1,\ldots,k\}} \sigma_1\left(V_{sel}, \Pi_{\preceq\epsilon_i}(\mathcal{P}(\epsilon_i))\right) \wedge \sigma_1\left(V_{sel}, \varphi_P\right)$$

$$\equiv red\left(\left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \left(\bigwedge_{i\in\{1,\ldots,k\}} [+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{i-1}}]\mathcal{P}(\epsilon_i)\right)\ldots\right.$$

$$\left.\ldots \wedge [+_{\mathrm{m}}\alpha_{\epsilon_1}]\ldots[+_{\mathrm{m}}\alpha_{\epsilon_{k-1}}][+_{\mathrm{m}}\alpha_{\epsilon_k}]\Box_{\mathrm{m}}\alpha_G\right)$$

$$= red\left(\left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \langle\!\langle\epsilon_1\rangle\!\rangle \ldots \langle\!\langle\epsilon_{k-1}\rangle\!\rangle\langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathrm{m}}\alpha_G\right)$$

□

Proof of Theorem 8

Proof ($\Rightarrow$) Suppose that the cognitive planning problem $\langle\Sigma, Op, \alpha_G\rangle$ has a solution plan. Then, by Proposition 5, we know that there is a solution plan $P = \epsilon_1, \ldots, \epsilon_k$ with $k \leq |Op|$ and $\epsilon_i \neq \epsilon_j$ for all $i < j$. By definition of solution plan, we have $\Sigma \vDash_{\mathbf{M}} \langle\!\langle\epsilon_1\rangle\!\rangle \ldots \langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathrm{m}}\alpha_G$, and then using respectively Propositions 4 and 2, $red\left(\left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \langle\!\langle\epsilon_1\rangle\!\rangle \ldots \langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathrm{m}}\alpha_G\right)$ is valid in the class $\mathbf{M}$. We consider the valuation V_{sel} of selectors in Sel_{Op} such that P is the corresponding designated plan (i.e. $\forall i, j$ such that $1 \leq i \leq j \leq k$, we have $\sigma_1(v, s_{\epsilon_i\preceq\epsilon_j}) = \top$ and $\forall\epsilon, \epsilon' \in Op\backslash P$ we have $\sigma_1(v, s_{\epsilon\preceq\epsilon'}) = \bot$). Then, by Proposition 7, $\sigma_1(V_{sel}, \varphi_{\langle\Sigma,Op,\alpha_G\rangle})$ is valid in the class $\mathbf{M}$, by Theorem 2,$tr_1\left(\sigma_1(V_{sel}, \varphi_{\langle\Sigma,Op,\alpha_G\rangle})\right)$ is valid in the class $\mathbf{K}$, and by Theorem 3 and Proposition 6, $\sigma_2\left(V_{sel}, F_{\langle\Sigma,Op,\alpha_G\rangle}\right)$ is valid in propositional logic. Hence, V_{sel} gives a valuation of selectors such that for all truth values of other variables in $F_{\langle\Sigma,Op,\alpha_G\rangle}$ this latter formula is true. Then the quantified boolean formula Q is true.

($\Leftarrow$) Let v a valuation of variables in Sel_{Op} for which Q is true. Note that, in this case, $\varphi_{Sel_{Op}}$ is evaluated to $\top$ as it remains as a conjunct of $F_{\langle\Sigma,Op,\alpha_G\rangle}$ after application of tr_1^{sel} and tr_2^{sel} on $\varphi_{\langle\Sigma,Op,\alpha_G\rangle}$. We have to prove that the corresponding designated plan $P = \epsilon_1, \ldots, \epsilon_k$ with $k \leq |Op|$ is a solution plan. As the quantified boolean formula Q is true in particular for the valuation v of existentially quantified variables, $\sigma_2\left(V_{sel}, F_{\langle\Sigma,Op,\alpha_G\rangle}\right)$ is valid in propositional logic. Moreover, by Propositions 6 and 7, we have $\sigma_2\left(V_{sel}, F_{\langle\Sigma,Op,\alpha_G\rangle}\right)$ is equisatisfiable to the reduction into propositional logic of $\left(\bigwedge_{\alpha\in\Sigma} \Box_{\mathrm{m}}\alpha\right) \rightarrow \langle\!\langle\epsilon_1\rangle\!\rangle \ldots \langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathrm{m}}\alpha_G$ using respectively functions red, tr_1 and tr_2. Then, using respectively Theorems 2, 3, Propositions 2 and 4 we can deduce that $\Sigma \vDash_{\mathbf{M}} \langle\!\langle\epsilon_1\rangle\!\rangle \ldots \langle\!\langle\epsilon_k\rangle\!\rangle\Box_{\mathrm{m}}\alpha_G$ (i.e. P is a solution plan).

□

Acknowledgements This work is supported by the ANR project CoPains: "Cognitive Planning in Persuasive Multimodal Communication", PRCE-Défi 7-Axe 3-2018 project:https: www.irit.fr/CoPains.

References

1. Amgoud, L., Maudet, N., & Parsons, S. (2000). Modelling dialogues using argumentation. In *Proceedings of the fourth international conference on multiagent systems* (pp. 31–38). IEEE.
2. Aucher, G., & Bolander, T. (2013). Undecidability in epistemic planning. In *Proceedings of the 23rd international joint conference on artificial intelligence (IJCAI 2013)* (pp. 27–33). AAAI Press.
3. Audi, R. (1973). Intending. *The Journal of Philosophy, 70*(13), 387–403.
4. Bench-Capon, T. J. M. (2003). Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation, 13*(3), 429–448.
5. Black, E., Coles, A. J., & Hampson, C. (2017). Planning for persuasion. In *Proceedings of the 16th international conference on autonomous agents and multiagent systems (AAMAS 2017)* (Vol. 2, pp. 933–942). IFAAMAS.
6. Bolander, T., & Andersen, M. B. (2011). Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics, 21*(1), 9–34.
7. Bolander, T., Holm Jensen, M., & Schwarzentruber, F. (2015). Complexity results in epistemic planning. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI 2015)* (pp. 2791–2797). AAAI Press.
8. Bonnet, G., Leturc, C., & Lorini, E., et al. (2021). Influencing choices by changing beliefs: A logical theory of influence, persuasion, and deception. In *Proceedings of the second international workshop on deceptive AI (DeceptAI 2021), communications in computer and information science (CCIS)* (Vol. 1296, pp. 302–321). Springer.
9. Bonzon, E., & Maudet, N. (2011). On the outcomes of multiparty persuasion. In *Proceedings of the 8th international conference on argumentation in multi-agent systems (ArgMAS 2011)* (pp. 86–101). Springer.
10. Budzyńska, K., & Kacprzak, M. (2008). A logic for reasoning about persuasion. *Fundamenta Informaticae, 85*(1–4), 51–65.
11. Caridroit, T., Lagniez, J., & Le Berre, D., et al. (2017). A sat-based approach for solving the modal logic s5-satisfiability problem. In *Proceedings of the thirty-first AAAI conference on artificial intelligence (AAAI-17)* (pp. 3864–3870). AAAI Press.
12. Cialdini, R. B. (2001). *Influence: Science and practice*. Boston: Allyn & Bacon.
13. Cooper, M. C., Herzig, A., & Maffre, F., et al. (2016). A simple account of multi-agent epistemic planning. In *Proceedings of the 22nd European conference on artificial intelligence (ECAI 2016)* (pp. 193–201).
14. Cooper, M. C., Herzig, A., Maffre, F., et al. (2021). A lightweight epistemic logic and its application to planning. *Artificial Intelligence, 298*(103), 437.
15. Cosmides, L., & Tooby, J. (1992). Cognitive adaptions for social exchange. In Barkow, J., Cosmides, L., & Tooby, J. (Eds.), *The adapted mind: Evolutionary psychology and the generation of culture* (pp. 163–228). Communications in Computer and Information Science (CCIS): Oxford University Press.
16. Da Costa Pereira, C., Tettamanzi, A., & Villata, S. (2011). Changing one's mind: Erase or rewind? Possibilistic belief revision with fuzzy argumentation based on trust. In *Proceedings of the twenty-second international joint conference on artificial intelligence (IJCAI 2011)* (pp. 164–171). AAAI Press.
17. Davidson, D. (1980). *Essays on actions and events*. Oxford: Clarendon Press.
18. Davila, J. L. F., Longin, D., & Lorini, E., et al. (2021). A simple framework for cognitive planning. In *Proceedings of the thirty-fifth AAAI conference on artificial intelligence (AAAI 2021)* (pp. 6331–6339). AAAI Press. https://ojs.aaai.org/index.php/AAAI/article/view/16786.
19. Demolombe, R. (2004). Reasoning about trust: A formal logical framework. In *Proceedings of the second international conference on trust management (iTrust 2004), LNCS* (Vol. 2995, pp. 291–303). Springer.
20. Dignum, F., Dunin-Keplicz, B., & Verbrugge, R. (2001). Creating collective intention through dialogue. *Logic Journal of the IGPL, 9*(2), 289–304. https://doi.org/10.1093/jigpal/9.2.289
21. Fagin, R., Halpern, J., Moses, Y., et al. (1995). *Reasoning about knowledge*. Cambridge: MIT Press.
22. Ghallab, M., Howe, A., Knoblock, C., et al. (1998). PDDL—The planning domain definition language. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.212.
23. Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice*. Burlington: Morgan Kaufmann.
24. Goldman, A. (1979). What is justified belief? In Pappas, G. (Ed.), *Justification and knowledge* (pp. 1–25). D. Reidel.
25. Goldman, A. I. (2006). *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford: Oxford University Press.
26. Halpern, J. Y., & Moses, Y. (1992). A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence, 54*(3), 319–379.

27. Hintikka, J. (1962). *Knowledge and belief*. New York: Cornell University Press.
28. Hunter, A. (2015). Modelling the persuadee in asymmetric argumentation dialogues for persuasion. In *Proceedings of the 24th international conference on artificial intelligence (IJCAI 2015)* (pp. 3055–3061). AAAI Press.
29. Hunter, A. (2018). Towards a framework for computational persuasion with applications in behaviour change. *Argument & Computation, 9*(1), 15–40.
30. Kominis, F., & Geffner, H. (2015). Beliefs in multiagent planning: From one agent to many. In Brafman, R. I., Domshlak, C., Haslum, P., et al. (Eds.), Proceedings of the 25th international conference on automated planning and scheduling (ICAPS 2015) (pp. 147–155). AAAI Press.
31. Kominis, F., & Geffner, H. (2017). Multiagent online planning with nested beliefs and dialogue. In *Proceedings of the twenty-seventh international conference on automated planning and scheduling (ICAPS 2017)* (pp. 186–194). AAAI Press.
32. Ladner, R. E. (1977). The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing, 6*(3), 467–480.
33. Lakemeyer, G., Lespérance, Y. (2012). Efficient reasoning in multiagent epistemic logics. In *Proceedings of the 20th European conference on artificial intelligence (ECAI 2012), frontiers in artificial intelligence and applications* (Vol. 242, pp 498–503). IOS Press.
34. Lê Cong, S., Pinchinat, S., & Schwarzentruber, F. (2018). Small undecidable problems in epistemic planning. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden* (pp. 4780–4786). http://jijcai.org.
35. Leturc, C., & Bonnet, G. (2022). Reasoning about manipulation in multi-agent systems. *Journal of Applied Non Classical Logics, 32*(2–3), 89–155. https://doi.org/10.1080/11663081.2022.2124067
36. Lismont, L., & Mongin, P. (1994). On the logic of common belief and common knowledge. *Theory and Decision, 37*, 75–106.
37. Lomuscio, A., Qu, H., & Raimondi, F. (2017). MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer, 19*, 9–30.
38. Lorini, E. (2018). In praise of belief bases: Doing epistemic logic without possible worlds. In *Proceedings of the thirty-Second AAAI conference on artificial intelligence (AAAI-18)* (pp. 1915–1922). AAAI Press.
39. Lorini, E. (2020). Rethinking epistemic logic with belief bases. *Artificial Intelligence, 282*, 103233.
40. Lorini, E. (2021). A qualitative theory of cognitive attitudes and their change. *Theory and Practice of Logic Programming, 21*(4), 428–458.
41. Lorini, E., & Demolombe, R. (2008). From binary trust to graded trust in information sources: A logical perspective. In *Proceedings of the 11th international workshop on trust in agent societies (TRUST 2008). Revised selected and invited papers, LNCS* (Vol. 5396, pp. 205–225). Springer.
42. Lorini, E., & Schwarzentruber, F. (2021). Multi-agent belief base revision. In *Proceedings of the 30th international joint conference on artificial intelligence (IJCAI 2021)*. http://ijcai.org.
43. Lorini, E., Sabouret, N., & Ravenet, B., et al. (2022). Cognitive planning in motivational interviewing. In *Proceedings of the 14th international conference on agents and artificial intelligence (ICAART 2022)* (pp. 508–517). SCITEPRESS.
44. Löwe, B., Pacuit, E., & Witzel, A. (2011). DEL planning and some tractable cases. In *Proceedings of the 3rd international international workshop on logic, rationality and interaction (LORI 2011)* (pp. 179–192). Berlin, Heidelberg: Springer.
45. Makinson, D. (1997). Screened revision. *Theoria, 63*, 14–23.
46. Meyer, J. J., & van der Hoek, W. (1995). *Epistemic logic for AI and computer science*. Cambridge: Cambridge University Press.
47. Miller, W. R., & Rollnick, S. (2012). *Motivational interviewing: Helping people change*. New York: Guilford Press.
48. Muggleton, S., & de Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming, 19–20*, 629–679.
49. Muise, C., Belle, V., & Felli, P., et al. (2015a). Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of the 29th AAAI conference on artificial intelligence (AAAI 2015)* (pp. 3327–3334). AAAI Press.
50. Muise, C., Chakraborti, T., & Agarwal, S., et al. (2019). Planning for goal-oriented dialogue systems. CoRR abs/1910.08137. arXiv:1910.08137.
51. Muise, C., Belle, V., Felli, P., et al. (2021). Efficient multi-agent epistemic planning: Teaching planners about nested belief. *Artificial Intelligence, 302*, 103605.
52. Muise, C. J., Dignum, F., & Felli, P., et al. (2015b). Towards team formation via automated planning. In *Proceedings of the 2015 international workshop on coordination, organizations, institutions, and normes in agent systems (COIN XI), lecture notes in computer science* (Vol. 9628, pp. 282–299). Springer.

53. Muise, C. J., Miller, T., & Felli, P. et al. (2015c). Efficient reasoning with consistent proper epistemic knowledge bases. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems (AAMAS 2015)* (pp. 1461–1469). ACM.
54. Perloff, R. M. (2003). *The dynamics of persuasion: Communication and attitudes in the 21st century*. Mahwah: L Erlbaum.
55. Prakken, H. (2006). Formal systems for persuasion dialogue. *The Knowledge Engineering Review, 21*(2), 163–188.
56. Proietti, C., & Yuste-Ginel, A. (2019). Persuasive argumentation and epistemic attitudes. In *Proceedings of the second international workshop on dynamic logic. New trends and applications (DALI 2019), LNCS* (Vol. 12005, pp. 104–123). Springer.
57. Rabinowitz, N. C., Perbet, F., & Song, H. F., et al. (2018). Machine theory of mind. In *Proceedings of the 35th international conference on machine learning (ICML 2018), proceedings of machine learning research* (Vol. 80, pp. 4215–4224). PMLR.
58. Rashotte, L. (2009). Social influence. In G. Ritzer & J. M. Ryan (Eds.), *Concise Blackwell encyclopedia of sociology*. Oxford: Blackwell.
59. Salhi, Y. (2019). On an argument-centric persuasion framework. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems (AAMAS 2019)* (pp. 1279–1287). International Foundation for Autonomous Agents and Multiagent Systems.
60. Searle, J. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.
61. Shams, Z., Vos, M. D., & Oren, N., et al (2016). Normative practical reasoning via argumentation and dialogue. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence (IJCAI 2016)* (pp. 1244–1250). IJCAI/AAAI Press.
62. Sipser, M. (2013). *Introduction to the theory of computation* (3rd ed.). Delhi: Cengage Learning.
63. Stalnaker, R. (2002). Common ground. *Linguistics and Philosophy, 25*(5–6), 701–721.
64. Teixeira, M. S., & Dragoni, M. (2022). A review of plan-based approaches for dialogue management. *Cognitive Computation, 14*(3), 1019–1038.
65. van Ditmarsch, H., van der Hoek, W., Kooi, B. (2007). Dynamic epistemic logic. Synthese Library, Springer, Netherlands. https://books.google.fr/books?id=dKRQPHvOIGQC.
66. Walton, D., & Krabbe, E. (1995). *Commitment in dialogue: Basic concepts of interpersonal reasoning. SUNY series in logic and language*. State University of New York Press.
67. Weber, K., Janowski, K., & Rach, N., et al. (2020). Predicting persuasive effectiveness for multimodal behavior adaptation using bipolar weighted argument graphs. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems (AAMAS 2020)* (pp. 1476–1484). International Foundation for Autonomous Agents and Multiagent Systems.