

An Algebraic Structuring of Epistemic States for BDI Agents in Uncertain Environments

Charles A. N. Costa
University of Brasília
Computer Science Department
Brasília, Brazil
charles.costa@aluno.unb.br

Marlo Souza
Federal University of Bahia
Computer Science Institute
Salvador, Brazil
msouza1@ufba.br

Célia Ghedini Ralha
University of Brasília
Computer Science Department
Brasília, Brazil
ghedini@unb.br

ABSTRACT

Reasoning about uncertainty is challenging for the design of Belief-Desire-Intention (BDI) agents. In dynamic, faulty, noisy, and partially observable environments, agents must frequently make decisions under uncertain beliefs. To address this ongoing challenge, the literature proposes several approaches, such as the use of stratified belief bases, as in the CAN+ framework. CAN+ provides a robust formal framework for agents to store and revise beliefs as weighted formulas. Nevertheless, reasoning about weighted formulas beyond entailment can be difficult. To better address these difficulties, we contribute an algebraic approach to reasoning about uncertain beliefs, focusing on stratified (or weighted) belief bases within the CAN+ framework and on a belief revision operator for updating such belief bases. In addition, we suggest modifications to the syntactic revision operator of CAN+ to structure an algebraic group, thus facilitating reasoning about the properties of systems implemented in CAN+. By providing a strong algebraic basis for the language, this work facilitates simplification rules and algorithms for reasoning over uncertain beliefs related to belief updating. To illustrate these concepts, we present the application of CAN+ reasoning for solving equations and supporting regression planning with the backward search algorithm in the vacuum world scenario.

KEYWORDS

Belief-Desire-Intention; epistemic states; uncertain beliefs; algebra

ACM Reference Format:

Charles A. N. Costa, Marlo Souza, and Célia Ghedini Ralha. 2026. An Algebraic Structuring of Epistemic States for BDI Agents in Uncertain Environments. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026)*, Paphos, Cyprus, May 25 – 29, 2026, IFAAMAS, 9 pages. <https://doi.org/10.65109/AWUD4334>

1 INTRODUCTION

Autonomous agents, such as surveillance robots or planetary exploration rovers, achieve their goals through plans. They go on a mission equipped with a repertoire of previously programmed plans for what they should select, combine, and execute, depending on the environmental conditions and the goals they must achieve. Those agents compare these plans by examining their preconditions and possible outcomes as units, instead of verifying each action a

plan contains. They start with the current status of the environment and reason whether taking a path where each step is a plan would result in a world state that fulfills their desires. This is the case of agents developed under Belief-Desire-Intention (BDI) model [9].

Traditional platforms for agent development, such as the Conceptual Agent Notation (CAN) [22] and AgentSpeak(L) [8], fail to help programmers deal with uncertainty. One of the promising approaches to solve this problem is representing the knowledge an agent collects about the world as graded beliefs. It consists of associating a numerical value with a possible state of the world that symbolizes how much the agents believe that the state is correct. This numerical value, a weight, is used to justify the decisions taken by an agent. A disadvantage of this approach is that it makes belief manipulation significantly more challenging than when true or false values are associated with beliefs. One of those approaches is the CAN+ language, introduced by Bauters et al. [4, 5]. CAN+ extends the CAN semantics, representing beliefs as epistemic states, to deal with uncertain beliefs.

A traditional approach to binary (true or false) beliefs is to represent a state of the world as a set of literals. If the letter a symbolizes the belief that it is sunny, the belief set of an agent must contain the literal a . On the other hand, if it is cloudy, then the agent's belief must not contain the literal a . However, if it is not the case, this literal must be removed from the set of beliefs by the revision process. This domain, provided by the usual operations of union, intersection, and complement of sets, forms an algebraic structure referred to as the Boolean algebra [15]. Many-valued algebras (MV-Algebras), as used to manipulate graded beliefs in Casali et al. [10], are defined over monoids with additional properties, a much more limited structure than the Boolean algebra [12]. However, epistemic states, as defined in Ma and Liu [18] and used in CAN+, are not MV-algebras.

In this work, we aim to help autonomous agent developers by structuring weighted beliefs as an Abelian group. Abelian group properties guarantee the existence of inverses and the identity element. An algebra endowed with these properties allows the resolution of equations using cancellation of terms, something not guaranteed with monoids. We aim to bridge the gap between expressiveness and manipulability, preserving the semantic properties of weighted beliefs while reintroducing algebraic tools. The proposed approach offers a promising path toward making autonomous agent development more robust and explainable in real-world uncertain environments.

The hypothesis we assume in this work is that structuring epistemic states as an Abelian group will facilitate the development of autonomous agents under uncertainty, providing more expressiveness and facilitating the equation of beliefs.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). <https://doi.org/10.65109/AWUD4334>

This work adopts a semi-formal, algebraic, and logical methodology based on abstract algebra and BDI agents. It begins by presenting the framework of belief revision in CAN+, and then we propose an extension of this framework to axiomatize the existence of inverses and the identity element. The introduced axioms allow us to define a representation of graded beliefs as an Abelian group, along with extra propositions to facilitate their equation. Using our proposal, we examine how regressive reasoning is employed to solve CAN+ agent problems. We then present an algorithm for regressive planning in CAN+ and describe its operation using a hypothetical scenario.

The rest of the article includes the background in Section 2, related work in Section 3, the systematic and structured algebraic proposal in Section 4, the proposal illustration using the backward search algorithm in Section 5, and finally, final considerations in Section 6.

2 BACKGROUND

This section presents an overview of fundamental concepts, including epistemic states and the CAN+ language related to BDI agents. The BDI rational model offers a robust framework for agents grounded in Bratman’s philosophical theory of practical reasoning [9]. BDI enables agents to represent their mental states—beliefs, desires, and intentions—and to update these beliefs dynamically through perception and inference, effectively handling complex and changing environments. The separation between introspection and external interaction allows agents to maintain coherent, deliberative behavior while adapting to dynamic conditions, facilitating applications across robotics, fault-tolerant systems, and simulations. Programming languages like AgentSpeak(L) [8] support BDI agent implementation by allowing flexible representations of beliefs, goals, and inference, although handling uncertainty remains a challenge for traditional frameworks.

2.1 Epistemic States Overview

Epistemic states gain different connotations depending on the context. In knowledge representation literature, it refers to a formal representation of incomplete and imperfect knowledge held by one or more agents about the world [20]. Frequently, epistemic states represent not only what an agent accepts as true but also how strongly they believe in it. This graduation allows the agent to reason about conflicting beliefs, stabilizing an order of plausibility or preference.

An influential formalization of epistemic states is by Ordinal Conditional Functions (OFCs), introduced by Spohn [26], which associates an integer value indicating a degree of plausibility to a possible world. OFCs are the basic framework structure for the representation of uncertainty in Ma and Liu [18], and they are used by Bauters et al. [5] for the definition of the CAN+ language, which we will focus on in this paper.

2.2 CAN+ Language

CAN+ is an extension of the framework CAN, designed to deal with uncertain beliefs in BDI agents. It employs syntactic representations of weighted formulas to represent an agent’s epistemic state, as well as a revision operator to update it with new incoming

information. The authors show the connections of their epistemic framework to probability and possibilistic theories, as well as providing tractable instantiations of their theory. Let us introduce the basic components of the language, which will be the basis for our discussion in Section 4.

Let At be a finite set of atoms, and the language \mathcal{L}^{At} constructed over At defined in Backus-Naur Form (BNF) as:

$$\varphi ::= a \mid \neg a \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2$$

Note that all $\varphi \in \mathcal{L}^{At}$ are in Negation Normal Form (NNF). We define the set Ω as all possible worlds from At , i.e., set of complete sets of literals over At . Henceforth $At = \{a, b\}$ we have $\Omega = \{\{a, b\}, \{a, \neg b\}, \{\neg a, b\}, \{\neg a, \neg b\}\}$. We say a possible world $\omega \in \Omega$ is a model for a formula $\varphi \in \mathcal{L}$ if and only if ω makes φ true, denoted as $\omega \models \varphi$. $Mod(\varphi)$ is the set of all models of a formula φ . A well-formed formula in CAN+ is defined over the extended language \mathcal{L}_{\geq}^{At} , defined in BNF as:

$$\psi ::= a \mid \neg a \mid \psi_1 \vee \psi_2 \mid \psi_1 \wedge \psi_2$$

$$\varphi ::= \psi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \psi_1 \geq \psi_2 \mid \psi_1 > \psi_2 \mid \text{not } \varphi$$

Notice that the formula $\text{not } \varphi$ is interpreted as a negation by failure in CAN+. Also, we will usually define $\neg(\psi_1 > \psi_2)$ as $\psi_2 \geq \psi_1$, and $\neg(\psi_1 \geq \psi_2)$ as $\psi_2 > \psi_1$. When a distinction is not required, we will use \mathcal{L} as a synonym for \mathcal{L}^{At} and \mathcal{L}_{\geq} as a synonym for \mathcal{L}_{\geq}^{At} .

Following Bauters et al. [5], an epistemic state denotes the credences attributed by an agent to the possible states of the world and will be represented as a mapping of possible worlds to a degree of belief in that world. The epistemic state definition is very similar to the OFCs or ranking functions by [26].

Definition 2.1 ([5]). Let Ω be a set of possible worlds. An epistemic state is a mapping $\Phi : \Omega \rightarrow \mathbb{Z}^{\infty}$, where $\mathbb{Z}^{\infty} = \mathbb{Z} \cup \{-\infty, +\infty\}$.

The order between possible worlds induced by an epistemic state is a way of organizing the possible worlds into levels according to their degrees of plausibility. The most plausible worlds occupy the highest level of plausibility, while the least plausible worlds stay in the inferior levels. This ordering can be lifted to a plausibility order on propositional formulas, which will be used to reason about the beliefs of the agent.

Definition 2.2 ([5]). Let Φ be an epistemic state, and $\alpha \in \mathcal{L}$. We call $\lambda : \mathcal{L} \mapsto \mathbb{Z}^{\infty}$ an epistemic valuation induced by Φ if

$$\lambda(\alpha) = \max\{\Phi(\omega) \mid \omega \models \alpha\}$$

Moreover, we can extend the epistemic valuation to CAN+ formulas, i.e., to formulas $\varphi \in \mathcal{L}_{\geq}$. To do that, Bauters et al. [5] provide a rewriting strategy, which transforms a formula $\varphi \in \mathcal{L}_{\geq}$ into a formula $\varphi' \in \mathcal{L}$ with equal credence. We will not present the transformation for brevity and urge the interested reader to consult the original work.

Definition 2.3 ([5]). Let Φ be an epistemic state and φ a formula in \mathcal{L}_{\geq} . We say that φ is entailed by Φ , written as $\Phi \models \varphi$, if and only if $\lambda(\varphi) > \lambda(\neg\varphi)$.

We are now set to admit representations for an epistemic state in a more syntactically convenient form.

Definition 2.4 ([5]). A weighted belief base \mathcal{B} is a set of formulas of the form (ψ, m) with $\psi \in \mathcal{L}$ and $m \in \mathbb{Z}^1$ so that there does not exist $(\psi_i, m_i), (\psi_j, m_j) \in \mathcal{B}$ for which $\psi_i \wedge \psi_j$ is satisfiable.

Importantly, weighted belief bases induce an epistemic state. Thus, an epistemic state can be represented as a weighted belief base, and each weighted belief base corresponds to a single epistemic state.

Definition 2.5. ([5]) Let \mathcal{A} be a weighted belief base. The epistemic state $\Phi_{\mathcal{A}}$ induced by \mathcal{A} is defined as

$$\Phi_{\mathcal{A}}(\omega) = \begin{cases} m & \text{if there exists a } (\psi, m) \in \mathcal{A} \text{ such that } \omega \models \psi \\ 0 & \text{otherwise} \end{cases}$$

From Definition 2.3, we can thus say that an agent represented by a weighted belief base \mathcal{A} believes in a formula φ if φ is entailed by its induced epistemic state.

Definition 2.6. Let \mathcal{A} be a weighted belief base and $\varphi \in \mathcal{L}$ be a propositional formula. We say \mathcal{A} entails φ (or the agent believes in φ) if $\Phi_{\mathcal{A}} \models \varphi$.

Notice that belief bases are syntactic representations of epistemic states. As such, different bases may induce the same epistemic state. In these cases, we will say that the bases are equivalent.

Definition 2.7. Let \mathcal{B}_1 and \mathcal{B}_2 be weighted belief bases. We say that they are equivalent, denoted $\mathcal{B}_1 \equiv \mathcal{B}_2$, if their induced epistemic states are the same.

It is trivial to see that this notion induces an equivalence relation among weighted belief bases.

LEMMA 2.8. *The relation \equiv defined in Definition 2.7 is an equivalence relation on the set of all weighted belief bases over a finite set of propositional symbols \mathcal{A} .*

The definition of a syntactic revision operator is practical since it allows agents to process and revise their beliefs efficiently. The role of a syntactic revision operator is to ensure that the framework achieves computational efficiency, which is critical for the implementation of BDI agent software and their application in real-world scenarios. Definition 2.9 brings the syntactic revision operator defined in [5].

Definition 2.9 ([5]). Let \mathcal{B} be a weighted belief base and $a = (\alpha, m)$, with $m \neq 0$. The syntactic revision \circ_s is given by $\mathcal{B} \circ_s a = \mathcal{B}' \cup ((\alpha \wedge \neg \vee(\mathcal{B}'^*))_m)^2$ with \mathcal{B}' given by:

$$\mathcal{B}' = \{(\psi \wedge \alpha, n + m), (\psi \wedge \neg \alpha, n) \mid (\psi, n) \in \mathcal{B}_\alpha\} \cup \mathcal{B} \setminus \mathcal{B}_\alpha$$

Where $\mathcal{B}_\alpha = \{(\psi, n) \mid (\psi, n) \in \mathcal{B} \text{ and } \psi \wedge \alpha \text{ is consistent}\}$ and $\mathcal{B}^* = \{\psi \mid (\psi, n) \in \mathcal{B}\}$, i.e., the set of formulas stripped of their weights. Whenever $\psi \wedge \neg \alpha$ is inconsistent, omit it, \mathcal{B}' , as it conveys no information.

The intuition behind Definition 2.9 is that when revising \mathcal{B} with the input a we update the weights with all models shared by \mathcal{B}^* and α adding m , we leave untouched the models not shared by \mathcal{B}^* and α , and we include with weight m the models of α that are not in \mathcal{B}^* .

¹Note that this definition excludes $\{\infty, -\infty\}$ from the weight domain.

² $\vee(\mathcal{B}^*) = \beta_1 \vee \dots \vee \beta_n$, and $\neg \vee(\mathcal{B}^*) = \neg \beta_1 \wedge \dots \wedge \neg \beta_n$, by De Morgan.

In practical scenarios, BDI agent designers must instantiate epistemic states using an appropriate theory to assess the uncertainty associated with states of the world. One of the advantages of the CAN+ approach is that each epistemic state can use the instantiation that better fits its model of uncertainty. Authors of [5] and [23] studied the application of probability theory and possibility theory to instantiate epistemic states. While probability theory is suitable for modelling stochastic events [27], possibility theory is adequate in situations where uncertainties come from incomplete information [6].

Discrete probabilistic distributions can represent beliefs by mapping each possible world to a probability of it being true, i.e., $P : \Omega \rightarrow [0, 1]$, such that $\sum_{\omega \in \Omega} P(\omega) = 1$. When we say that $P(\omega) = 1$, it means that ω is true for sure, and $P(\omega) = 0$ means that ω is not possible. The work in [23] extended the AgentSpeak(L) language to support CAN+ definitions and implemented an interpreter for the extended language.

3 RELATED WORK

The related work focuses on contributions to the formal understanding and computational treatment of rational multi-agent systems, integrating belief revision, dynamic preference reasoning, and goal-oriented deliberation for complex, uncertain environments. Table 1 summarizes graded beliefs works dealing with uncertainty.

Casali et al. [10] approaches uncertainties in BDI agents using graded beliefs, desires, and intentions, but details about semantic and operational belief revision are out of scope. Although graduation on desires and intentions represents the agent's preferences on what it wants to bring about, regarding beliefs, it represents a degree of plausibility. They named their framework g-BDI. The formalization applies Łukasiewicz logic. Later, Casali et al. [11] extended the g-BDI framework, where agents could have multiple mental contexts, each one with their particular uncertain interpretation, and defined bridge rules to allow interaction between contexts. Casali et al. [11] extended the original framework with rational Pavelka logic [14] and propositional dynamic logic [28] to deal with the effects of action execution and to represent probabilistic beliefs. Biga and Casali [7] presented the g-BDI implementation (g-Jason), which takes advantage of Jason annotations to represent degrees.

Bauters et al. [4] extended CAN and CAN(Plan) to include graded beliefs managed by epistemic states, named CAN+ and CAN(Plan)+, respectively. An epistemic state represents a particular manner of interpreting the uncertainty associated with a specific aspect of the agent's concerns. The joining of all epistemic states of an agent forms a Global Uncertain Beliefs (GUB) set. The authors presented a way of representing graded beliefs, as well as the semantics of belief revision and the justification of implications. The formal basis is [18], where epistemic states take into account the strength of the evidence before changing an agent's beliefs. In [5], the proposal is extended with a focus on tractability. The work of Scannell [23] introduces an environment capable of running agents developed using the CAN+ proposed in Bauters et al. [4, 5].

Maintaining the consistency of graded belief bases is the aim of Zein and Pereira [29]. They adapt the algorithm in Alechina et al. [1] for contraction (exclusion) of inconsistent beliefs by removing from the chain of dependency the beliefs with a lower degree of

plausibility. In their proposal, the input sources must inform the agent of the degree of plausibility of the belief, which corresponds to the degree of preference. If revising the belief base with an input results in an inconsistency, the belief with the lowest preference is removed. While [7] focuses on goal and plan selection, [29] focuses on belief base maintenance. Thus, we consider [7] and [29] complementary works. However, other approaches allow the existence of inconsistent beliefs without harm to the agent.

Archibald et al. [2] extended CAN to allow that actions have probabilistic outcomes. Each action outcome $\mu(\phi^+, \phi^-)$ has a probability p_i of happening, with the sum equal to 1. But they do not propose treatment for uncertain beliefs. The authors elaborate strategies for plan and intention selection under adequate probability distributions. In those strategies, they use bigraphs of Milner [19] to formalize state transition. The evaluation uses the BigraphER tool [24] and the PRISM model checker [16] in an intelligent manufacturing scenario. The experimental insight is that intention selection has more influence than plan selection on the agent’s reward. In [3], CAN+ is extended to enable the rationalization of BDI agents’ plan execution as Markov Decision Processes (MDP). Their proposal employs epistemic state concepts to represent uncertain belief, considering uncertain inputs of one literal for tractability reasons. The proposed CAN_+^M language combines CAN+ and MDP. The authors provide a tool named CAN-Verify for translating CAN programs into bigraphs. Agents programmed in CAN_+^M can be executed using BigraphER and verified using PRISM.

Lorini and Schwarzenruber [17] extend belief base revision to multi-agent systems, introducing a framework where agents hold belief bases comprising both propositional and higher-order beliefs about themselves and others, partitioned into mutable and core components. Building on this foundation, they propose a belief revision operator inspired by screened revision, analyze its computational complexity, and develop an optimal model checking algorithm. Furthermore, we examine the complexity of epistemic planning within this framework, thus demonstrating its applicability to multi-agent scenarios. Souza et al. [25] investigate the application of dynamic preference logic to encode the mental attitudes of BDI agents. They focus on representing preferences over possible worlds through priority graphs. Their approach also interprets BDI programs within this logic. It leverages the connection between dynamic operations on preference models and transformations on priority graphs. This foundation facilitates reasoning about agent programs and serves as a tool for specifying reasoning mechanisms that ensure desirable properties in rationality theories.

As presented in Table 1, this work uses epistemic states as a formal framework to deal with uncertainty, proposing an operator to construct the negation of a belief expressed as a weighted formula. We suggest changes in the syntactic revision operator of CAN+ to structure an algebraic group, facilitating reasoning about the properties of systems implemented in CAN+.

4 ABELIAN GROUP EPISTEMIC STATES

This section presents an adaptation of the formal epistemic framework of [5], casting weighted belief bases as algebraic structures, namely an abelian group. With this, we employ standard group and Boolean equational theories to derive simplification rules, which

Table 1: Related work overview

Reference	Formal framework	Belief update	Algebraic equation
Casali et al. [10]	Lukasiewicz logic	–	–
Casali et al. [11]	Pavelka rational logic, dynamic logic	–	–
Bauters et al. [4, 5]	Epistemic states	✓	–
Zein and Pereira [29]	AGM postulates, belief justification	✓	–
Archibald et al. [2]	Bigraphs	–	–
Archibald et al. [3]	Epistemic states, Bigraphs	✓	–
Lorini and Schwarzenruber [17]	Propositional, higher-order beliefs	✓	–
Souza et al. [25]	Dynamic preference logic, priority graphs	✓	–
This work	Epistemic states, abstract algebra	✓	✓

can be used to lower the practical complexity of computing update of belief bases throughout the agent execution.

As stated, we will structure the domain of weighted belief bases, with a generalized syntactic revision \oplus_s , based on the operation presented in Definition 2.9, as an abelian group and explore its properties to provide simplification rules for reasoning.

We call a structure $\mathcal{G} = \langle G, * \rangle$ an abelian group if its operation $*$ satisfies the properties of closure, associativity, existence of unary and inverse elements, and commutativity. A binary operator $*$ satisfies closure (or is closed in a set G) if $\forall a, b \in G$ we have $a * b \in G$.

4.1 Belief revision operator

We will define our set of elements G as the set of all weighted belief bases, for a given set of atomic symbols At . This will be the basis for our algebraic structure.

Definition 4.1. Let At be a finite set of propositional symbols and $\mathcal{L}_{\mathbb{Z}} = \{(\varphi, m) \mid \varphi \in \mathcal{L}, m \in \mathbb{Z}\}$ be the set of all possible weighted formulas formed over the language \mathcal{L} . We define the set $G \subseteq 2^{\mathcal{L}_{\mathbb{Z}}}$ as the set of all weighted belief bases over At .

Notice that, the arguments of the syntactic revision operator \circ_s in Definition 2.9 are a weighted belief base \mathcal{B} and a weighted formula (φ, m) . Although we can interpret a weighted formula (φ, m) as a unitary belief set $\{(\varphi, m)\}$, the operator \circ_s is not defined if we take the second operand as an arbitrary weighted belief base and, as such, it is not closed on the set of weighted belief bases G . To circumvent this problem, we redefine the syntactic revision operator to take as an argument two weighted belief bases of arbitrary length.

Definition 4.2. Let $\mathcal{A}, \mathcal{B} \in G$ be weighted belief bases. We define the operator $\oplus_s : G \times G \mapsto G$ as $\mathcal{A} \oplus_s \mathcal{B} = \mathcal{A}_B^+ \cup \mathcal{A}_B^- \cup \mathcal{B}_A^-$, where:

$$\begin{aligned}\mathcal{A}_B^+ &= \{(\alpha \wedge \beta, m+n) \mid (\alpha, m) \in \mathcal{A} \text{ and } (\beta, n) \in \mathcal{B}\} \\ \mathcal{A}_B^- &= \{(\alpha \wedge \neg \bigvee (\mathcal{B}^*), m) \mid (\alpha, m) \in \mathcal{A}\} \\ \mathcal{B}_A^- &= \{(\beta \wedge \neg \bigvee (\mathcal{A}^*), n) \mid (\beta, n) \in \mathcal{B}\}\end{aligned}$$

Being that $C^* = \{\varphi \mid (\varphi, k) \in C\}$, the set of formulas without their weights. The sets above defined all disconsider pairs (φ, k) s.t. φ is inconsistent.

Definition 4.2 generalizes the syntactic revision operator of [5], considering the intuition behind it, i.e., the result of the revision of a weighted belief base by another consists of updating the credence of all shared models between them, while maintaining the other models untouched. Figure 1 presents this idea, where Ω is the set of all possible worlds, \mathcal{A}, \mathcal{B} are weighted belief bases, and the gray circles represent the models of the formulas in \mathcal{A} and \mathcal{B} . We use \oplus_s to represent our adapted operator in Definition 4.2 to make it clear its difference from that of Definition 2.9.

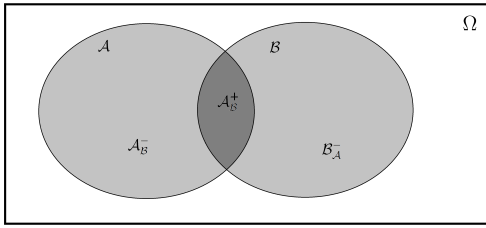


Figure 1: Venn diagram representing the models of two weighted belief bases, where $A = \text{Mod}(\mathcal{A})$ and $B = \text{Mod}(\mathcal{B})$. Models inside the dark gray area are those updated by the revision of \mathcal{A} by \mathcal{B}

In fact, it is easy to see that, in the case where the input information $\mathcal{B} = \{(\beta, n)\}$, a weighted belief base with only one weighted formula, the result of applying Definition 2.9 is equivalent to the application of Definition 4.2, since all elements of $\mathcal{A} \circ_s (\beta, n)$ are in $\mathcal{A} \oplus_s \mathcal{B}$.

LEMMA 4.3. If \mathcal{A} is a weighted belief base and $\mathcal{B} = \{b\}$ such that $b = (\beta, n)$, then $\mathcal{A} \circ_s b = \mathcal{A} \oplus_s \mathcal{B}$.

As a generalization of \circ_s , \oplus_s is also NP-complete. It follows from the fact that deciding satisfiability when applying \circ_s or \oplus_s reduces from SAT. Fortunately, the problem becomes tractable if additional restrictions are considered, as noted in [5]. Notice that this result is not surprising, since any revision with an arbitrary weighted belief base can be reduced to a sequence of single-input revisions.

PROPOSITION 4.4. Let $\mathcal{B} = \{(\varphi_i, m_i) \mid i \in n\}$ be a weighted belief base, then $\mathcal{B} \equiv \emptyset \circ_s (\varphi_0, m_0) \circ_s \dots \circ_s (\varphi_{n-1}, m_{n-1})$

PROOF. It suffices to remember that, by Definition 2.4, if \mathcal{B} is a weighted belief base, than for any $i, j \in n$, $\varphi_i \wedge \varphi_j \equiv \perp$. As such, the result of the sequence of revision is exactly \mathcal{B} . \square

From these results, it is immediate that the operation \oplus_s is closed on the set G .

PROPOSITION 4.5. The revision operator \oplus_s is closed in G .

Moreover, the results above tell us that in most cases, we need only to consider the simple case with unitary inputs to investigate this structure. Also, from Lemma 4.3, Proposition 4.4, and Bauters et al's [5, pg. 745] Proposition 4 about the semantic consistency of the simple input revision operation, we also obtain the following.

THEOREM 4.6. Let \mathcal{A}_0 be a weighted belief base, $s = \langle \mathcal{B}_1, \dots, \mathcal{B}_n \rangle$ be a sequence of weighted belief bases, and $\mathcal{A}_{i+1} = \mathcal{A}_i \oplus_s \mathcal{B}_{i+1}$. Let Φ_i be the epistemic state induced by \mathcal{A}_i . Then, for any $\omega \in \Omega$, it holds that $(\varphi, m) \in \mathcal{A}_n$ iff $\Phi_n(\omega) = m$ for every $\omega \in \text{Mod}(\varphi)$.

Notice that Definitions 2.9 and 4.2 seem different, in the sense that the first explicitly preserves the elements of the first argument (a weighted belief base) that are inconsistent with the second (an input), i.e., the element $\mathcal{B} \setminus \mathcal{B}_\alpha$ in Definition 2.9. The equivalent set in Definition 4.2 would be a set of all $(\alpha, m) \in \mathcal{A}$ that are inconsistent with all elements in \mathcal{B} . However, if such an element exists, it would be in \mathcal{A}_B^- , thus this difference is only apparent.

LEMMA 4.7. If there is $(\alpha, m) \in \mathcal{A}$ such that $(\alpha \wedge \bigvee (\mathcal{B}^*)) \equiv \perp$, then $(\alpha, m) \in \mathcal{A}_B^-$.

By Definition 4.2, the revision of an empty belief base by a non-empty weighted belief base results in the non-empty base itself.

LEMMA 4.8. Take $\mathcal{A} = \emptyset$ and $\mathcal{B} \neq \emptyset$, then $\mathcal{A} \oplus_s \mathcal{B} = \mathcal{B}$.

We can also see that there is an identity element with respect to the operation \oplus_s in G .

PROPOSITION 4.9. There is a weighted belief base $\mathcal{E} = \emptyset$, s.t. for any weighted belief base \mathcal{A} , it holds that $\mathcal{A} \oplus_s \mathcal{E} = \mathcal{E} \oplus_s \mathcal{A} = \mathcal{A}$.

Notice that the empty base attributes zero plausibility to all formulas, there are an infinite number of equivalent bases to it, and all of them can be taken as an identity element.

LEMMA 4.10. Let $\Delta \subset \mathcal{L}$ be a finite set of propositional formulas, s.t. if $\varphi, \psi \in \Delta$, then $\varphi \wedge \psi \equiv \perp$. The weighted belief base $\mathcal{A} = \{(\varphi, 0) \mid \varphi \in \Delta\} \equiv \emptyset$.

Similarly, for any weighted belief base, there is an inverse element³. w.r.t. the revision operation.

PROPOSITION 4.11. Let \mathcal{A} be a weighted belief base. For all $\mathcal{A} \in G$ exists an $-\mathcal{A} \in G$ such that $\mathcal{A} \oplus_s -\mathcal{A} \equiv \emptyset$.

PROOF. Let us take $-\mathcal{A} = \{(\alpha, -m) \mid (\alpha, m) \in \mathcal{A}\}$. Since \mathcal{G} is the set of all the weighted formulas, then $-\mathcal{A} \in G$. Furthermore, all models in \mathcal{A} are also models in $-\mathcal{A}$, then $\mathcal{A}_{-\mathcal{A}}^-$ and $-\mathcal{A}_{\mathcal{A}}^-$ are equal to \emptyset . Remains $\mathcal{A}_{-\mathcal{A}}^+ = \{(\alpha \wedge \beta, m+n) \mid (\alpha, m) \in \mathcal{A} \text{ and } (\beta, n) \in -\mathcal{A}\}$. Since \mathcal{A} is a weighted belief base, then $\alpha \wedge \beta \equiv \perp$ if $\alpha \neq \beta$. Thus, $\mathcal{A}_{-\mathcal{A}}^+ = \{(\alpha \wedge \alpha, m + (-m)) \mid (\alpha, m) \in \mathcal{A}\} = \{(\alpha, 0) \mid (\alpha, m) \in \mathcal{A}\}$. Performing the simplification, we have $\mathcal{A}_{-\mathcal{A}}^+ \equiv \emptyset$. Thus, $\mathcal{A} \oplus_s -\mathcal{A} \equiv \emptyset$. \square

³Notice here that we are talking about equivalence, not equality, as syntactically the results may differ. Later, we will use the equivalence relation \equiv to provide a set, namely the quotient set, for which the resulting structure forms an abelian group in the proper sense.

Notice that, thanks to Proposition 4.4, to prove associativity, we need only to consider the case of unary belief bases.

PROPOSITION 4.12. *Let \mathcal{A} , \mathcal{B} , and \mathcal{C} weighted belief bases. It holds that $(\mathcal{A} \oplus_s \mathcal{B}) \oplus_s \mathcal{C} \equiv \mathcal{A} \oplus_s (\mathcal{B} \oplus_s \mathcal{C})$.*

PROOF. We need only to consider the case $a = (\alpha, m)$, $b = (\beta, n)$, and $c = (\gamma, p)$

Step 1: Find $(a \oplus_s b) \oplus_s c$.

Expanding $(a \oplus_s b)$ using Definition 4.2:

$$(a \oplus_s b) \oplus_s c = [\{(\alpha \wedge \beta, m+n), (\alpha \wedge \neg\beta, m), (\neg\alpha \wedge \beta, n)\}] \oplus_s c$$

Applying Definition 4.2, we have

$$(a \oplus_s b) \oplus_s c = \{(\alpha \wedge \beta \wedge \gamma, m+n+p), (\alpha \wedge \neg\beta \wedge \gamma, m+p), (\neg\alpha \wedge \beta \wedge \gamma, n+p)\} \cup \{(\alpha \wedge \beta \wedge \neg\gamma, m+n), (\alpha \wedge \neg\beta \wedge \neg\gamma, m), (\neg\alpha \wedge \beta \wedge \neg\gamma, n)\} \cup ((\gamma \wedge \neg \bigvee \mathcal{B}_\gamma), p)$$

However $(\gamma \wedge \neg \bigvee \mathcal{B}_\gamma), p) = (\gamma \wedge (\neg\alpha \wedge \neg\beta), p) \equiv (\neg\alpha \wedge \neg\beta \wedge \gamma, p)$, since $\neg \bigvee \mathcal{B}_\gamma = \neg\alpha \wedge \neg\beta$. Thus, substituting:

$$(a \oplus_s b) \oplus_s c = \{(\alpha \wedge \beta \wedge \gamma, m+n+p), (\alpha \wedge \neg\beta \wedge \gamma, m+p), (\neg\alpha \wedge \beta \wedge \gamma, n+p), (\alpha \wedge \beta \wedge \neg\gamma, m+n), (\alpha \wedge \neg\beta \wedge \neg\gamma, m), (\neg\alpha \wedge \beta \wedge \neg\gamma, n), (\neg\alpha \wedge \neg\beta \wedge \gamma, p)\}$$

Step 2: Find $a \oplus_s (b \oplus_s c)$.

Since \oplus_s is commutative, $a \oplus_s (b \oplus_s c) = (b \oplus_s c) \circ a$. Thus, we know that

$$(b \oplus_s c) \oplus_s a = \{(\beta \wedge \gamma \wedge \alpha, n+p+m), (\beta \wedge \neg\gamma \wedge \alpha, n+m), (\neg\beta \wedge \gamma \wedge \alpha, p+m), (\beta \wedge \gamma \wedge \neg\alpha, n+p), (\beta \wedge \neg\gamma \wedge \neg\alpha, n), (\neg\beta \wedge \gamma \wedge \neg\alpha, p), (\neg\beta \wedge \neg\gamma \wedge \alpha, m)\}$$

Rearranging the terms, we have

$$(b \oplus_s c) \oplus_s a = \{(\alpha \wedge \beta \wedge \gamma, m+n+p), (\alpha \wedge \beta \wedge \neg\gamma, m+n), (\alpha \wedge \neg\beta \wedge \gamma, m+p), (\neg\alpha \wedge \beta \wedge \gamma, n+p), (\neg\alpha \wedge \beta \wedge \neg\gamma, n), (\neg\alpha \wedge \neg\beta \wedge \gamma, p), (\alpha \wedge \neg\beta \wedge \neg\gamma, m)\}$$

Finally, we can see that $(a \oplus_s b) \oplus_s c = a \oplus_s (b \oplus_s c)$. \square

Since the revision operator only changes the credence of the models that satisfy both the initial base and the input by addition of the plausibility values associated with each information, preserving the credence of all other models, it is easy to see that the operation is commutative.

PROPOSITION 4.13. *The revision operator is commutative, i.e., for any $\mathcal{A}, \mathcal{B} \in G$, it holds that $\mathcal{A} \oplus_s \mathcal{B} \equiv \mathcal{B} \oplus_s \mathcal{A}$.*

From Propositions 4.5, 4.12, 4.9, 4.11, and 4.13, we can conclude the following.

THEOREM 4.14. $\mathcal{G} = \langle G_{\oplus_s}, \oplus_s \rangle$, in which $[\mathcal{A}] \oplus_s [\mathcal{B}] = [\mathcal{A} \oplus_s \mathcal{B}]$, is an abelian group.

4.2 Simplification rules

Now, we will focus on exploring simple equivalences in the structure of G to provide simplification rules for reasoning on weighted belief bases. These simplification rules are important as reasoning about the agent's belief and plan execution under uncertainty can be reduced to computing the result of successive belief revision

operations, which can be stated as an equation on the elements of G . As such, these simplification rules can be used to simplify reasoning, reducing the formula complexity of the agents' belief bases during their execution, as well as reducing the complexity of reasoning by eliminating or simplifying the operations (and satisfiability checks) to be performed.

Let us begin with some basic simplification rules.

PROPOSITION 4.15. *Let $\mathcal{A}, \mathcal{B} \in G$ be weighted belief bases. It holds that:*

- if $(\varphi, 0) \in \mathcal{A}$, then $\mathcal{A} \equiv (\mathcal{A} \setminus \{(\varphi, 0)\})$
- if $(\varphi, m), (\varphi, -m) \in \mathcal{A}$, then $\mathcal{A} \equiv (\mathcal{A} \setminus \{(\varphi, m), (\varphi, -m)\})$
- for any $0 < k \in \mathbb{N}$, $\underbrace{\mathcal{A} \oplus_s \dots \oplus_s \mathcal{A}}_{k \text{ times}} = \{(\varphi, k \cdot m) \mid (\varphi, m) \in \mathcal{A}\}$

Since, by Proposition 4.4, weighted belief bases can be rewritten as sequences of unitary weighted formulas, these simplification rules above, together with well-known Boolean equivalences, will be the basis to construct an equational theory which will allow us to reason about an agent's beliefs. In order to do that, however, we will focus the presentation on the following sequences of weighted formulas. As such, as previously done by Bauters et al. [5], given a sequence $\langle s \rangle = \langle s_1, \dots, s_n \rangle$ of weighted formulas, we will refer to the sequential operation $\mathcal{A} \circ_s s_1 \circ_s \dots \circ_s s_n$ simply as $\mathcal{A} \oplus_s \langle s \rangle$.

Definition 4.16. Let $\langle s_1 \rangle : m \mapsto \mathcal{L}_{\mathbb{Z}}$, $\langle s_2 \rangle : n \mapsto \mathcal{L}_{\mathbb{Z}}$ be finite sequences of weighted formulas. We say $\langle s_1 \rangle$ and $\langle s_2 \rangle$ are operationally equivalent, or simply equivalent, denoted by $\langle s_1 \rangle \equiv \langle s_2 \rangle$, if for any weighted belief base $\mathcal{A} \in G$, it holds that $\mathcal{A} \oplus_s s_1 \equiv \mathcal{A} \oplus_s s_2$.

As a corollary of Proposition 4.4, Theorem 4.6, and Theorem 4.14, we can verify operational equivalences between sequences simply by verifying their effect on the empty belief base.

LEMMA 4.17. *Let $s_1 : m \mapsto \mathcal{L}_{\mathbb{Z}}$, $s_2 : n \mapsto \mathcal{L}_{\mathbb{Z}}$ be finite sequences of weighted formulas.*

$$s_1 \equiv s_2 \quad \text{iff} \quad \emptyset \oplus_s s_1 \equiv \emptyset \oplus_s s_2$$

Now, we can state the simplifying rules in terms of sequences.

LEMMA 4.18. *The following equivalences hold:*

- (1) $\langle (\varphi, 0) \rangle \equiv \langle \rangle$
- (2) $\langle (\varphi, m), (\varphi, -m) \rangle \equiv \langle \rangle$
- (3) $\langle (\varphi, m_0), \dots, (\varphi, m_{n-1}) \rangle \equiv \langle (\varphi, \sum_{i < n} m_i) \rangle$
- (4) $\langle (\varphi, m), (\varphi \vee \psi, n) \rangle \equiv \langle (\varphi, m+n), (\neg\varphi \wedge \psi, m) \rangle$
- (5) if $\varphi \equiv \psi$, then $\langle (\varphi, m) \rangle \equiv \langle (\psi, m) \rangle$
- (6) $\langle (\varphi \vee \psi, m) \rangle \equiv \langle (\varphi \wedge \psi, m), (\varphi \wedge \neg\psi, m), (\neg\varphi \wedge \psi, m) \rangle$

Notice that rule 6 above is important since we can simplify bases to remove disjunctions, reducing complexity for the satisfiability check - this will be important in Theorem 4.21 later. Finally, to show how these rules can be used, we will need a form of reduction rule. For that, let us introduce some operations on sequences, which will be necessary.

Definition 4.19. Let $n \in \mathbb{N}$ be a natural number, $\langle s \rangle : n \mapsto \mathcal{L}_{\mathbb{Z}}$ be a finite sequence of weighted formulas of size n , i.e., $\langle s \rangle = \langle s_0, s_1, \dots, s_{n-1} \rangle$. We define

- if $i < n$, then $\langle\langle s \rangle - i\rangle$ is the sequence of $n - 1$ elements s.t. $s - i(k) = s(k)$ if $k < i$ and $s - i(k) = s(k - 1)$ otherwise. In other words, $\langle\langle s \rangle - i\rangle = \langle s_0, \dots, s_{i-1}, s_{i+1}, \dots, s_{n-1} \rangle$ if the removal of the i -th element of the sequence.
- if $I \subseteq n$, $\langle\langle s \rangle - I\rangle$ is the $n - |I|$ sequence obtained by removing all elements with index $i \in I$ of the sequence s .
- if $\langle t \rangle = \langle t_0, \dots, t_{k-1} \rangle$ is a sequence of weighted formulas of size k , then $\langle\langle s \rangle + \langle t \rangle\rangle$ is the $n + k$ sequence $\langle\langle s \rangle + \langle t \rangle\rangle(k) = \langle s \rangle(k)$ if $k < n$ and $\langle\langle s \rangle + \langle t \rangle\rangle(k) = \langle t \rangle(k - n)$, otherwise. In other words, $\langle\langle s \rangle + \langle t \rangle\rangle = \langle s_0, \dots, s_{n-1}, t_0, \dots, t_{k-1} \rangle$.

Finally, we can show that for any sequences of weighted formulas, i.e., for any sequences of updates in a belief base, we can simplify them by removing unnecessary information and rewriting them in a more convenient form

PROPOSITION 4.20. *Let $\langle s \rangle$, $\langle r \rangle$, and $\langle t \rangle$ be finite sequences of weighted formulas, s.t. $\langle s \rangle = \langle s_1, \dots, s_n \rangle$ and $\langle r \rangle \equiv \langle t \rangle$. If there is $I = \{i_1, \dots, i_k\}$ s.t. $r = \langle s_{i_1}, s_{i_2}, \dots, s_{i_k} \rangle$, then $\langle s \rangle \equiv \langle\langle s \rangle - I\rangle + \langle t \rangle$*

With this, we can provide a general result on how to compute the beliefs of an agent after an update.

THEOREM 4.21. *Let \mathcal{A} be a weighted belief base and $\langle t \rangle$ be a finite sequence of weighted formulas of size n . For any $\varphi \in \mathcal{L}$, it holds that $\mathcal{A} \oplus \langle t \rangle \models \varphi$ iff $\Delta \models \varphi$, where $\Delta = \{\psi_i \mid \langle t \rangle(i) = (\psi_i, k_i) \text{ for } i \in I\} \cup \{Y\}$ is a consistent set of propositional formulas, for some $I \subseteq n$ and $(Y, k) \in \mathcal{A}$ satisfying $m = k + \sum_{i \in I} \text{val}(i)$, with $\text{val}(i) = k_i$, for $i \in I$ and $\langle t \rangle(i) = (\psi_i, k_i)$, s.t. m is maximal.*

PROOF. To construct the proof, it suffices for us to remember that i) the explicit beliefs of an agent are described by the formulas with highest plausibility values in the belief base, ii) by definition of \oplus_s (and \circ_s), the operator only changes the plausibility of formulas when they are jointly consistent, and iii) all formulas in a weighted belief are inconsistent with each other. Thus, to compute the agents' beliefs, it suffices to search for the subset of formulas in the sequence $\langle t \rangle$ which are consistent with some formula on \mathcal{A} , and that maximizes the sum of plausibility values. \square

5 APPLICATIONS OF ABELIAN GROUP EPISTEMIC STATES

In this section, we discuss the applications of our proposal. First, we demonstrate how to apply our structure to solve equations over weighted belief domains. Then, we apply the same ideas to demonstrate that our approach is also useful for more abstract reasoning, such as in automated planning. We present a variation of the backward search algorithm for CAN+ using the vacuum world scenario.

5.1 Solving equations in CAN+

A direct application of an abelian group epistemic states is using cancellation of terms to solve equations, like in Example 5.1.

Example 5.1. The weighted belief base of an agent is $\mathcal{B}_0 = (\alpha, 10)$. We have $\lambda(\alpha) = 10$ and $\lambda(-\alpha) = 0$. Thus, since $\lambda(\alpha) > \lambda(-\alpha)$, $\mathcal{B} \models \alpha$. The agent desires to bring about a possible world \mathcal{B}_1 such that $\mathcal{B}_1 \models -\alpha$. What would be the input that results in

this state of the world? First, let us say that the goal is to make $\mathcal{B}_1 = \{(\alpha, 10), (-\alpha, 20)\}$. The agent can use our proposal to reason:

$$\begin{aligned} \mathcal{B} \oplus_s x &= \mathcal{B}_1 \\ \{(\alpha, 10)\} \oplus_s x &= \{(\alpha, 10), (-\alpha, 20)\} \\ \{(\alpha, 10)\} \oplus_s x \oplus_s \{(\alpha, -10)\} &= \{(\alpha, 10), (-\alpha, 20)\} \oplus_s \{(\alpha, -10)\} \\ x &= \{(-\alpha, 20)\} \end{aligned}$$

We can see that $(\alpha, 10) \circ_s (-\alpha, 20)$ is, indeed, $\{(\alpha, 10), (-\alpha, 20)\}$. The agent can now use this information to select an appropriate plan to bring about \mathcal{B}_1 .

5.2 Inverses in backward search

In traditional automated planning, an action scheme a is the triple $\langle \text{Pre}(a), \text{Add}(a), \text{Del}(a) \rangle$. $\text{Pre}(a)$ are the preconditions that must be met to execute the action. $\text{Add}(a)$ and $\text{Del}(a)$ are the effects that set the literals true or false, respectively. This representation supports regressive reasoning, i.e., computing the state that must precede the execution of an action. This kind of reasoning is applied in the backward search algorithm for automated planning [21].

The backward search is a classical automated planning algorithm. It starts from the goal G and iteratively regresses to the initial state. At each step, the algorithm selects a relevant action from the agent's repertoire of actions. An action is relevant if $G \cap \text{Add}(a) \neq \emptyset$ and $G \cap \text{Del}(a) = \emptyset$. It then computes a new regressed goal G' by Equation (1), representing the conditions required before executing the selected action to ensure that the goal holds afterward [13]. This process continues until the initial state is reached or no application actions remain.

$$G' = (G \setminus \text{Add}(a)) \cup \text{Pre}(a) \quad (1)$$

In CAN+ [5, 23], the action's preconditions are formulas in \mathcal{L}_{\geq} . The effects are sequences of inputs since actions can only change the state of the world by revision. CAN+, thus, defines an action scheme a as $\langle \text{Pre}(a), \text{Pos}(a) \rangle$. Regarding a goal G , relevant actions are those in which, for all literals $G \models l$, we have $\text{Pos}(a) \models l$. Furthermore, for no l should we have $\text{Pos}(a) \models \neg l$. Applying our algebraic approach, we calculate the regression goal G' using Equation (2), which is based on the inverse of $\text{Pos}(a)$.

$$G' = G \oplus_s (-\text{Pos}(a)) \oplus_s \text{Pre}(a) \quad (2)$$

The Backward-CAN+ in Algorithm 1 implements the backward search strategy, using our approach for the inverse of a weighted belief base. It takes the initial state s_0 , the target goal G , and the action set A as input. The procedure starts by initializing the plan π to \emptyset (line 1), and immediately returns π if the base case $s_0 \models G$ is met (line 2). Otherwise, it enters an iterative phase (lines 4-8), considering only relevant actions $a \in A$ that contribute to G without contradicting it. For each such action, the regressed goal G' is computed (line 5), and a recursive invocation on $\langle s_0, G', A \rangle$ (line 6). If the recursive call returns a valid plan π , the current action a is appended to the sequence, returning $a \cdot \pi$ (lines 7-8). If no action leads to a valid plan, it terminates by returning *Failure* (line 9).

As an illustration of how Algorithm 1 works on CAN+ algebraic structure, take the vacuum world scenario described in Figure 2. In

Algorithm 1: BackwardSearch_CAN+(s_0, G, A)

Input: Initial state s_0 , goal G , actions A
Output: Plan π achieving G or *Failure*

```

1  $\pi \leftarrow \{\}$ 
2 if  $G \models s_0$  then
3   return  $\pi$ 
4 foreach  $a \in A$  such that  $\exists l(G \models l \implies Pos(a) \models l)$  and
    $\nexists l(G \models l \implies Pos(a) \models \neg l)$  do
5    $G' = G \oplus_s (-Pos(a)) \oplus_s Pre(a)$ 
6    $\pi \leftarrow BackwardSearch\_CAN+(s_0, G', A)$ 
7   if  $\pi \neq Failure$  then
8     return  $a \cdot \pi$ 
9 return Failure

```

this scenario, when the robot detects dirt on Square 3, it must head there (if not), move, and vacuum it. Then go to Square 1 to deploy garbage into the trash bin. We divide the robot's beliefs into three epistemic states: Φ_1 are the beliefs about Square 1 being dirty or not, represented by the literal *hasDirty*; Φ_2 is whether the robot is already carrying dirt or not, represented by the literal *loaded*; and Φ_3 is the beliefs about the position of the robot on the board, literals $\{At(1), At(2), At(3)\}$.

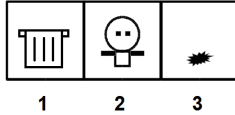


Figure 2: Vacuum world scenario with three squares

In the vacuum world domain, three actions define the agent's behavior: *Clean*, *Deploy*, and *Travel*. The *Clean* is applicable when the agent is in Square 3 ($At(3)$), can detect dirt (*HasDirt*), and the vacuum is empty ($\neg Loaded$), thus $Pre(Clean) = Dirt \wedge \neg Loaded \wedge At(3)$. Its execution removes dirt and loads the vacuum, represented by the post effects $Pos_{\Phi_1}(Clean) = \{(HasDirt, -1), (\neg HasDirt, 1)\}$ over Φ_1 , and $Pos_{\Phi_2} = \{(Loaded, 1), (\neg Loaded, -1)\}$. *Travel* enables movement between locations, requiring $At(X) \wedge \neg At(Y)$ and producing $Pos(Travel(X, Y))_{\Phi_3} = \{(At(X), -1), (At(Y), 1)\}$. *Deploy* is applicable when $Loaded \wedge At(1)$ holds, and unloads the vacuum producing $Pos(Deploy)_{\Phi_2} = \{(Loaded, -1), (\neg Loaded, 1)\}$. Together, these action specifications formally capture the domain's transition dynamics through logical preconditions and symbolic effects.

When the agent believes it is in square 2, the belief base can be $\mathcal{B}_{\Phi_3} = (At(2), 1)$. If the agent wants to go to square 3, it executes $Travel(At(2), At(3))$, possibly causing the following revision:

$$\begin{aligned} \mathcal{B}'_{\Phi_3} &= \mathcal{B}_{\Phi_3} \oplus Pos_{\Phi_3}(Travel(At(2), At(3))) \\ \mathcal{B}'_{\Phi_3} &= (At(2), 1) \oplus \{(At(2), -1), (At(3), 1)\} \\ \mathcal{B}'_{\Phi_3} &= (At(3), 1) \end{aligned}$$

Applying the inverse of the *Travel* action post effects causes the agent to obtain the inverse route.

$$-Pos_{\Phi_3}(Travel(At(X), At(Y))) = Pos_{\Phi_3}(Travel(At(Y), At(X)))$$

Figure 3 represents an execution of Algorithm 1, from the initial state in node 8, represented in Figure 2, to the goal state in node 1. The path after node 3 was not represented because from then on, the subsequent states would be equal to nodes 2 and 4, which were already visited.

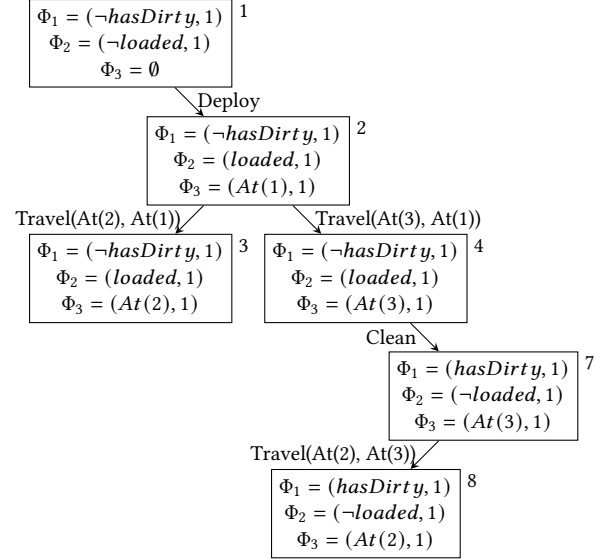


Figure 3: Plan generated by Algorithm 1 for vacuum world scenario. Nodes are states, and the edges are the regression of actions

6 CONCLUSION

The article presents an algebraic approach to reasoning with uncertainty for BDI agents using weighted beliefs. The CAN+ approach for weighted beliefs of BDI agents extends CAN to support tractable epistemic states. However, it still lacked structural analysis that favored problem-solving and regression reasoning. Our unified approach axiomatizes representation and operations of epistemic states using an abelian group structure to express revision and regression. The approach embodies an expressive perspective of how actions interact with belief changes, preserving compatibility with agent reasoning frameworks. The results indicate that algebraic reasoning can support planning tasks by making explicit the relationships between states, actions, and goals. Thus, a systematic treatment of belief update and plan generation does not harm the symbolic and interpretable nature of BDI programming, warranting further investigation. In addition, analyzing the computational cost of the proposed operator in simulated or real environments is desirable. Analyzing how the approach can be applied to traditional BDI programming problems, such as plan execution scheduling, provides broader insights into the application.

ACKNOWLEDGMENTS

Prof. Célia Ghedini Ralha thanks for the research productivity grant 309688/2021-3 from the Brazilian National Council for Scientific and Technological Development (CNPq).

REFERENCES

- [1] Natasha Alechina, Mark Jago, and Brian Logan. 2005. Resource-bounded belief revision and contraction. In *International Workshop on Declarative Agent Languages and Technologies*. Springer, Utrecht, The Netherlands, 141–154.
- [2] Blair Archibald, Muffy Calder, Michele Sevegnani, and Mengwei Xu. 2021. Probabilistic BDI agents: actions, plans, and intentions. In *Software Engineering and Formal Methods: 19th International Conference, SEFM 2021, Virtual Event, December 6–10, Proceedings*. Springer, 262–281.
- [3] Blair Archibald, Michele Sevegnani, and Mengwei Xu. 2025. Modelling and verifying BDI agents under uncertainty. *Science of Computer Programming* 242 (2025), 103254.
- [4] Kim Bauters, Weiru Liu, Jun Hong, Carles Sierra, and Lluís Godo. 2014. CAN(PLAN)+: extending the operational semantics of the BDI architecture to deal with uncertain information. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (Quebec City, Quebec, Canada) (UAI'14)*. AUAI Press, Arlington, Virginia, USA, 52–61.
- [5] Kim Bauters, Kevin McAreevey, Weiru Liu, Jun Hong, Lluís Godo, and Carles Sierra. 2017. Managing different sources of uncertainty in a BDI framework in a principled way with tractable fragments. *Journal of Artificial Intelligence Research* 58 (2017), 731–775.
- [6] Salem Benferhat, Didier Dubois, Henri Prade, and Mary-Anne Williams. 2002. A practical approach to revising prioritized knowledge bases. *Studia Logica* 70, 1 (2002), 105–130.
- [7] Adrián Biga and Ana Casali. 2013. G-jason: An extension of jason to engineer agents capable to reason under uncertainty. In *Proc. of 14th Intelligent Agent and Systems Workshop*. EDULP, La Plata, Argentina, 17.
- [8] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. *Programming multi-agent systems in AgentSpeak using Jason*. Vol. 15. John Wiley & Sons, Chichester, UK.
- [9] Michael Bratman. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, Cambridge.
- [10] Ana Casali, Lluís Godo, and Carles Sierra. 2004. Graded BDI models for agent architectures. In *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, Lisbon, Portugal, 126–143.
- [11] Ana Casali, Lluís Godo, and Carles Sierra. 2011. A graded BDI agent model to represent and reason about preferences. *Artificial Intelligence* 175, 7–8 (2011), 1468–1478.
- [12] Roberto L Cignoli, Itala M d'Ottaviano, and Daniele Mundici. 2013. *Algebraic foundations of many-valued reasoning*. Vol. 7. Springer Science & Business Media, Heidelberg, Germany.
- [13] Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated planning and acting*. Cambridge University Press, Cambridge, UK.
- [14] Zuzana Haniková. 2023. Rational Pavelka logic: The best among three worlds? *Fuzzy Sets and Systems* 456 (2023), 92–106. <https://doi.org/10.1016/j.fss.2022.08.010> Logic (214 p.).
- [15] Thomas W Hungerford. 2012. *Algebra*. Vol. 73. Springer Science & Business Media, Heidelberg, Germany.
- [16] Marta Kwiatkowska, Gethin Norman, and David Parker. 2002. PRISM: Probabilistic Symbolic Model Checker. In *Computer Performance Evaluation: Modelling Techniques and Tools*, Tony Field, Peter G. Harrison, Jeremy Bradley, and Uli Harder (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 200–204.
- [17] Emiliano Lorini and Francois Schwarzentruber. 2021. Multi-Agent Belief Base Revision. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Zhi-Hua Zhou (Ed.)*. International Joint Conferences on Artificial Intelligence Organization, Montreal, Canada, 1959–1965. <https://doi.org/10.24963/ijcai.2021/270> Main Track.
- [18] Jianbing Ma and Weiru Liu. 2011. A framework for managing uncertain inputs: An axiomization of rewarding. *International Journal of Approximate Reasoning* 52, 7 (2011), 917–934.
- [19] Robin Milner. 2008. Bigraphs and their algebra. *Electronic Notes in Theoretical Computer Science* 209 (2008), 5–19.
- [20] Pavlos Peppas and Mary-Anne Williams. 1995. Constructive modelings for theory change. *Notre Dame Journal of Formal Logic* 36, 1 (1995), 120–133.
- [21] Stuart J. Russell and Peter Norvig. 2009. *Artificial Intelligence: a modern approach* (3 ed.). Pearson, London, England.
- [22] Sebastian Sardina, Lavindra De Silva, and Lin Padgham. 2006. Hierarchical planning in BDI agent programming languages: A formal approach. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. IFAAMAS, Japan, 1001–1008.
- [23] Aidan Scannell. 2018. *Extending BDI agents to model and reason with uncertainty*. Master of Research in Robotics and Autonomous Systems. University of Bristol and University of the West of England.
- [24] Michele Sevegnani and Muffy Calder. 2016. BigraphER: Rewriting and Analysis Engine for Bigraphs. In *Computer Aided Verification - 28th International Conference, CAV July 17-23, Proceedings, Part II*. Springer, Toronto, Canada, 494–501. https://doi.org/10.1007/978-3-319-41540-6_27
- [25] Marlo Souza, Alvaro Moreira, Renata Vieira, and John-Jules Ch. Meyer. 2017. A Dynamic Preference Logic for Reasoning about Agent Programming. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE Computer Society, Los Alamitos, CA, USA, 133–138. <https://doi.org/10.1109/BRACIS.2017.43>
- [26] Wolfgang Spohn. 1988. Ordinal conditional functions: A dynamic theory of epistemic states. In *Causation in decision, belief change, and statistics: Proceedings of the Irvine Conference on Probability and Causation*. Springer, Heidelberg, Germany, 105–134.
- [27] HM Taylor and S Karlin. 1998. *An introduction to stochastic modelling*. Academic Press, New York, USA.
- [28] Nicolas Troquard and Philippe Balbiani. 2023. Propositional Dynamic Logic. *The Stanford Encyclopedia of Philosophy (Fall 2023 Edition)* (2023). Edward N. Zalta & Uri Nodelman (eds.), URL = <<https://plato.stanford.edu/archives/fall2023/entries/logic-dynamic/>>.
- [29] Dima El Zein and Célia da Costa Pereira. 2020. Graded belief revision for jason: A rule-based approach. In *2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE, New York, USA, 211–218.