Abduction in PDT Logic

Karsten Martiny[1](✉) and Ralf Möller[2]

[1] Hamburg University of Technology, Hamburg, Germany
karsten.martiny@tuhh.de
[2] Universität zu Lübeck, Lübeck, Germany
moeller@uni-luebeck.de

Abstract. Probabilistic Doxastic Temporal (PDT) Logic is a formalism to represent and reason about belief evolutions in multi–agent systems. In this work we develop a theory of abduction for PDT Logic. This gives means to novel reasoning capabilities by determining which epistemic actions can be taken in order to induce an evolution of probabilistic beliefs into a desired goal state. Next to providing a formal account of abduction in PDT Logic, we identify pruning strategies for the solution space, and give a sound and complete algorithm to find minimal solutions to the abduction problem.

1 Introduction and Related Work

Epistemic and doxastic logics are used to reason about agents' knowledge. Formalizing the analysis of knowledge and belief through such logics has been an active topic of research in diverse fields. Numerous extensions to modal epistemic logic have been made to reason about knowledge in multi–agent settings (e.g., [8]), to add probabilistic knowledge (e.g., [7]), and to analyze the dynamic evolution of knowledge (e.g., [4]).

In realistic scenarios an agent usually has only incomplete and inaccurate information about the actual state of the world, and thus considers several different situations as actually being possible. As it receives new information (e.g., it observes some facts), it has to update its beliefs about these possible worlds such that they are consistent with the new information. These updates can for example result in regarding some worlds as impossible, or judging some worlds to be more likely than before. Thus, in addition to analyzing the set of worlds an agent believes to be possible, it is also expedient to quantify these beliefs in terms of probabilities. This provides means to specify fine–grained distinctions between the range of worlds that an agent considers possible but highly unlikely, and worlds that seem to be almost certainly the actual world.

When multiple agents are involved in such a setting, an agent may not only have varying beliefs regarding the facts of the actual world, but also regarding the beliefs of other agents. In many scenarios, the actions of one agent will not only depend on its belief in facts of the actual world, but also on its beliefs in some other agent's beliefs.

B. Pfahringer and J. Renz (Eds.): AI 2015, LNAI 9457, pp. 398–410, 2015.
DOI: 10.1007/978-3-319-26350-2_35

To analyze the belief evolution of multiple agents, problem domains can be modeled using *Probabilistic Doxastic Temporal (PDT) Logic* [12]. When analyzing these problem domains, one is often interested in determining what could be done to bring about a certain belief state of some agent. To illustrate this, consider the following example:

Example 1 (Cyber security). Suppose that an adversary is trying to break into a computer system. This is usually done by using an attack graph to detect and exploit potential vulnerabilities of the system. An attack graph specifies a set of paths (i.e., sequences of actions) to carry out an attack. Several paths of the attack graph might be used in parallel, potentially by different agents (for instance, a number of infected computers controlled by a botnet). Usually, attack patterns specified by one attack graph are used multiple times. This has two important ramifications. The adversary will learn from experience which of the paths yield a high probability of successful attacks to a system. Defenders in turn will be able to gain knowledge of the attack graph through the repeated observation of certain patterns. Thus, when a system is under attack, the defender will have beliefs about both the chosen attack paths and the adversary's belief regarding the success of the respective path. Naturally, the defender's goal is to choose countermeasures such that the attacker believes that further attacks are useless.

A formal analysis of belief evolutions in such a cyber security setting using PDT Logic has been presented in [14]. However, previous work only provides means for deductive reasoning about the consequences of given events. In this paper, we show how abduction can be formalized in PDT Logic. This enables us to determine a required minimal set of actions that one has to take in order to bring about a desired goal belief state. Next to cyber security settings as in the above example, this approach may be useful in various domains. To name only a few examples, in financial markets it might be critical for a company to determine what kind of information has to be released to the public such that the shareholders' belief in a positive outlook is sufficiently high. In cooperative multi–agent scenarios, it is useful to determine minimal required communication acts among agents, such that all agents obtain all relevant information. In this work, we focus on the theoretical aspects of the abduction problem. Due to space constraints, we can only provide examples to a very limited extend. However, detailed modeling examples using PDT Logic can be found for example in [12,14].

Abduction has been a subject of extensive research (e.g., [5,9]), with extensions to temporal logic (e.g., [3]) and uncertainty (e.g., [16]). However, there is little work that studies abduction in the context of both time and uncertainty. A recent study of abduction in settings involving both time and uncertainty has been introduced in [15]. This approach considers abduction for the single–agent case and uses time-invariant probabilities. By extending this work such that probabilistic multi–agent beliefs and their dynamic evolution can be represented, we develop a novel abductive formalism that is able to determine necessary actions to induce desired beliefs in a multi–agent scenario.

The remainder of this paper is structured as follows. In the next section, a brief overview of PDT Logic as introduced in [12] is given. Section 3 shows how abduction can be formalized using PDT Logic and — after deriving some conditions to prune the search space — presents an algorithm to solve the abduction problem. Finally, Sect. 4 concludes this work.

2 PDT Logic

We now briefly summarize the syntax and semantics of PDT Logic from [12]. A function–free first order logic language $\mathcal{L}$ with finite sets of constant symbols $\mathcal{L}_{cons}$ and predicate symbols $\mathcal{L}_{pred}$, and an infinite set of variable symbols $\mathcal{L}_{var}$ is given. Every predicate symbol $p \in \mathcal{L}_{pred}$ has an *arity*. A *term* is any member of the set $\mathcal{L}_{cons} \cup \mathcal{L}_{var}$. A term is called a *ground term* if it is a member of $\mathcal{L}_{cons}$. If $t_1, .., t_k$ are (ground) terms and p is a predicate symbol in $\mathcal{L}_{pred}$ with arity n, then $p(t_1, ..., t_k)$ with $k \in \{0, ..., n\}$ is a (ground) atom. If a is a (ground) atom, then a and $\neg a$ are (ground) *literals*. The set of all ground literals is denoted by $\mathcal{L}_{lit}$. As usual, $\mathcal{B}$ denotes the Herbrand Base of $\mathcal{L}$.

Time is modeled as a set τ of discrete time points $\tau = \{1, ..., t_{max}\}$. The set of agents is denoted by $\mathcal{A}$. To describe what a group of agents $\mathcal{G} \subseteq \mathcal{A}$ observes, observation atoms are defined as follows:

Definition 1. *For a non-empty group of agents $\mathcal{G} \subseteq \mathcal{A}$ and ground literal $l \in \mathcal{L}_{lit}$, $Obs_{\mathcal{G}}(l)$ is an* observation atom. *The set of all observation atoms is denoted by $\mathcal{L}_{obs}$.*

Both atoms and observation atoms are formulae. If F and G are formulae, then $F \wedge G$, $F \vee G$, and $\neg F$ are formulae.

Note that the formal concept of observations is not limited to express passive acts of observing facts, but can instead be used to model a wide range of actions: for instance, communication between agents could be modeled as group observations for the respective agents — the ramifications of the communication act are exactly the same as they would be in a shared observation (assuming that agents do not lie). In this sense, observations in PDT Logic represent the effects of epistemic actions in the line of [2] and are used to alter the belief state of agents — we will build on this below when formalizing the abduction problem.

Ontic facts and according observations form *worlds* (or *states* in the terminology of [8]). A world ω consists of a set of ground atoms and a set of observation atoms, i.e., $\omega \in 2^{\mathcal{B}} \cup 2^{\mathcal{L}_{obs}}$ With a slight abuse of notation, $a \in \omega$ and $Obs_{\mathcal{G}}(l) \in \omega$ are used to denote that an atom a (resp. observation atom $Obs_{\mathcal{G}}(l)$) holds in world ω. Since agents can only observe facts that actually hold in the respective world, admissibility conditions of worlds w.r.t. the set of observations can be defined:

Definition 2. *A world ω is admissible, iff for every observation $Obs_{\mathcal{G}}(l) \in \omega$ the observed fact holds (i.e., $x \in w$ if l is a positive literal x, and $x \notin w$ if l is a negative literal $\neg x$) and for every subgroup $\mathcal{G}' \subset \mathcal{G}$, $Obs_{\mathcal{G}'}(l) \in \omega$.*

We use Ω to denote the set of all admissible worlds. Satisfaction of a ground formula F by a world ω (denoted by $\omega \models F$) is defined the usual way [11].

Example 2. In a (highly simplified) formalization of 1, we could have a set of two agents $\mathcal{A} = A, D$, representing an attacker and defender, respectively. We assume that we have two different computer systems C_1, C_2, and actions $att(c), def(c)$ to represent that system c is attacked resp. defended. In this scenario, possible ontic facts could for example be $att(C_1)$ and $def(C_1)$. Corresponding observations to this could for example be $Obs_{\{D\}}(att(C_1))$, representing that the defender observes an attack on system C_1, or $Obs_{\{A\}}(def(C_1))$, representing that the attacker observes a defensive action to protect system C_1. In this scenario, the following could be examples of possible worlds:

$$\begin{aligned}
\omega_1 &= \{att(C_1), Obs_{\{A\}}(att(C_1))\}\},\\
\omega_2 &= \{att(C_1), Obs_{\{A\}}(att(C_1)), Obs_{\{D\}}(att(C_1))\},\\
\omega_3 &= \{att(C_1), Obs_{\{A,D\}}(att(C_1))\},
\end{aligned}$$

Naturally, if the attacker carries out an attack, she observes this attack, represented by $Obs_{\{A\}}(att(C_1))$ in all worlds. In ω_1, her attack is undetected, as the defender does not observe it. In worlds ω_2 and ω_3, the defender detects this attack. The difference between the latter two worlds is that in ω_3, there is a *shared* observation about the attack, i.e., both agents know that the respective opponent has the same observation, while in ω_2, the defender observes the attack, but the attacker is unaware of this.

Using the concept of admissible worlds, we can represent the evolution of time as sequences of worlds:

Definition 3. *A* thread *is a mapping* $Th : \tau \to \Omega$

Thus, a thread is a sequence of worlds and $Th(i)$ identifies the actual world at time i according to thread Th. The set of all possible threads (i.e., all possible sequences constructible from τ and Ω) is denoted by $\mathcal{T}$. A method of constructing such a set of threads induced by a set of PDT Logic belief formulae $\mathfrak{B}$ is described in [13]. Due to space constraints, we do not discuss this method here and instead simply assume that a specification of possible threads $\mathcal{T}$ induced by a set of PDT Logic belief formulae $\mathfrak{B}$ is given. For notational convenience, we assume that there is an additional prior world $Th(0)$ for every thread.

Temporal relationships between events can be expressed through temporal rules:[1]

Definition 4. *Let F, G be two formulae, and Δt a time interval. Then $r_{\Delta t}(F, G)$ is called a temporal rule.*

The meaning of such an expression is to be understood as "F is followed by G after exactly Δt time units".

[1] The introduction of PDT Logic in [12] enables the expression of a variety of temporal relationships through an axiomatic definition. Due to space constraints, we present an adapted simplified version that suffices for the purpose of this work.

2.1 Kripke Structures

With the definition of threads, a slightly modified version of Kripke structures [10] can be adopted. For a set $\mathcal{A}$ of n agents, a Kripke structure is defined as a tuple $\langle \Omega, \mathcal{K}_1(t), ..., \mathcal{K}_n(t) \rangle$, with the set of admissible worlds Ω and binary relations $\mathcal{K}_i$ on Ω for every agent $i \in \mathcal{A}$ and every time point $t \in \tau$. Intuitively, $(\omega, \omega') \in \mathcal{K}_i$ specifies that at time t in world ω, agent i considers ω' as a possible world as well.

These Kripke structures are initialized for each agent such all worlds that occur at time $t = 1$ in some thread Th' are considered possible.

$$\forall Th \in \mathcal{T}: \ \mathcal{K}_i(Th(0)) = \bigcup_{Th' \in \mathcal{T}} \{Th'(1)\}, \ i = 1, ..., n \tag{1}$$

Note that the set of time points τ ranges over $1, ..., t_{max}$. We use the auxiliary time point $t = 0$ only to simplify the subsequent presentation: by initializing the Kripke structures as specified above, we can express the Kripke structures for all time points $t \in \tau$ as results of successive updates to the respective $\mathcal{K}_i$.

With the evolution of time, each agent can eliminate the worlds that do not comply with its respective observations. Through the elimination of worlds, an agent will also reduce the set of threads it considers possible (if — due to some observation — a world ω is considered impossible at a time point t, then all threads Th with $Th(t) = \omega$ are considered impossible). It is assumed that agents have perfect recall and therefore will not consider some thread possible again if it was considered impossible at one point. Thus, $\mathcal{K}_i$ is updated w.r.t. the agent's respective observations such that it considers all threads possible that both comply with its current observations and were considered possible at the previous time point:

$$\begin{aligned} \mathcal{K}_i(Th(t)) := \{Th'(t) : (Th'(t-1) \in \mathcal{K}_i(Th(t-1)) \wedge \\ \{Obs_{\mathcal{G}}(l) \in Th(t) : i \in \mathcal{G}\} = \{Obs_{\mathcal{G}}(l) \in Th'(t) : i \in \mathcal{G}\})\} \end{aligned} \tag{2}$$

Note that — depending on the actual observations — different Kripke structures $\mathcal{K}_i$ may occur at a specific time point t. $K_i(t)$ is used to denote the set of all possible Kripke structures for agent i at time t.

Example 3. Consider the set of worlds $\omega_1, ..., \omega_3$ from the previous example. In the absence of any other information, the resulting Kripke structures in this case would be

$$\begin{aligned} \mathcal{K}_A(\omega_1) = \mathcal{K}_A(\omega_2) = \{\omega_1, \omega_2\}, \mathcal{K}_A(\omega_3) = \{\omega_3\}, \\ \mathcal{K}_D(\omega_1) = \{\omega_1\}, \mathcal{K}_D(\omega_2) = \{\omega_2\}, \mathcal{K}_D(\omega_3) = \{\omega_3\}, \end{aligned}$$

i.e., the attacker cannot distinguish between the worlds where her attack went undetected and where the attack was detected without her knowing about this. The defender in turn is able to distinguish between all three worlds, as his respective observations are unique in each of these worlds.

2.2 Subjective Posterior Temporal Probabilistic Interpretations

Each agent has probabilistic beliefs about the expected evolution of the world over time. This is expressed through subjective temporal probabilistic interpretations:

Definition 5. *Given a set of possible threads $\mathcal{T}$, some thread $\mathring{Th} \in \mathcal{T}$, a time point t' and an agent i, $\mathcal{I}_{i,t'}^{\mathring{Th}} : \mathcal{T} \to [0,1]$ specifies the* subjective posterior probabilistic temporal interpretation *from agent i's point of view at time t' in thread $\mathring{Th}$, i.e., a probability distribution over all possible threads: $\sum_{Th \in \mathcal{T}} \mathcal{I}_{i,t'}^{\mathring{Th}}(Th) = 1$. $\mathring{Th}$ is called the* point of view (pov) thread *of interpretation $\mathcal{I}_{i,t'}^{\mathring{Th}}$.*

The prior probabilities of each agent for all threads are then given by $\mathcal{I}_{i,0}^{\mathring{Th}}(Th)$. Since all threads are indistinguishable a priori, there is only a *single* prior distribution for each agent. Furthermore, in order to be able to reason about nested beliefs, it is assumed that prior probability assessments of all agents are commonly known (i.e., all agents know how all other agents assess the prior probabilities of each thread). This in turn requires that all agents have exactly the same prior probability assessment over all possible threads: if two agents have different, but commonly known prior probability assessments, we essentially have an instance of Aumann's well-known problem of "agreeing to disagree" [1]. Intuitively, if differing priors are commonly known, it is common knowledge that (at least) one of the agents is at fault and should revise its probability assessments. As a result, there is only one prior probability distribution which is the same from all viewpoints, denoted by $\mathcal{I}$.

Even though there is only a single prior probability distribution over the set of possible threads, it is still necessary to distinguish the viewpoints of different agents in different threads, as the definition of interpretation updates shows:

Definition 6. *Let i be an agent, t' a time point, and $\mathring{Th}$ a pov thread. Then, if the system is actually in thread $\mathring{Th}$ at time t', agent i's probabilistic interpretation over the set of possible threads is given by the update rule:*

$$\mathcal{I}_{i,t'}^{\mathring{Th}} = \begin{cases} \frac{1}{\alpha_{i,t'}^{\mathring{Th}}} \cdot \mathcal{I}_{i,t'-1}^{\mathring{Th}}(Th) & \text{if } Th(t') \in \mathcal{K}_i(\mathring{Th}(t')) \\ 0 & \text{if } Th(t') \notin \mathcal{K}_i(\mathring{Th}(t')) \end{cases} \tag{3}$$

with $\frac{1}{\alpha_{i,t'}^{\mathring{Th}}}$ being a normalization factor to ensure that $\sum_{Th \in \mathcal{T}} \mathcal{I}_{i,t'}^{\mathring{Th}}(Th) = 1$:

$$\alpha_{i,t'}^{\mathring{Th}} = \sum_{\substack{Th \in \mathcal{T}, \\ Th(t') \in \mathcal{K}_i(\mathring{Th}(t'))}} \mathcal{I}_{i,t'-1}^{\mathring{Th}}(Th) \tag{4}$$

Essentially, the update rule assigns all impossible threads a probability of zero and scales the probabilities of the remaining threads such that they are proportional to the probabilities of the previous time point.

2.3 The Belief Operator

Now, with the definitions of subjective posterior probabilistic temporal interpretations, the belief operator $B_{i,t'}^{\ell,u}(\varphi)$ to express agents' beliefs can be defined. Intuitively, $B_{i,t'}^{\ell,u}(\varphi)$ means that at time t', agent i believes that some fact φ is true with a probability $p \in [\ell, u]$. The probability interval $[\ell, u]$ is called the *quantification* of agent i's belief. F_t is used to denote that formula F holds at time t and, accordingly, $Obs_{\mathcal{G}}(l)_t$ to denote that an observation $Obs_{\mathcal{G}}(l)$ occurs at time t. These expressions are called time–stamped formulae and time–stamped observation atoms, respectively.

Definition 7. *Let i be an agent, t' a time point, and $[\ell, u] \subseteq [0, 1]$. Then,* belief formulae *are inductively defined as follows:*

1. *If F is a formula and t is a time point, then $B_{i,t'}^{\ell,u}(F_t)$ is a belief formula.*
2. *If $r_{\Delta t}(F, G)$ is a temporal rule, then $B_{i,t'}^{\ell,u}(r_{\Delta t}(F, G))$ is a belief formula.*
3. *If F and G are belief formulae, then so are $B_{i,t'}^{\ell,u}(F)$, $F \wedge G$, $F \vee G$, and $\neg F$.*

For a belief $B_{i,t'}^{\ell,u}(\varphi)$ about something, φ is called the belief object.

The semantics of this operator is defined as follows:

Definition 8. *Let i be an agent and $\mathcal{I}_{i,t'}^{\mathring{Th}}(Th)$ be agent i's interpretation at time t' in pov thread $\mathring{Th}$. Then, it follows from this interpretation that agent i believes at time t' with a probability in the range $[\ell, u]$ that*

1. *(Belief in ground formulae)*
   a formula F holds at time t (denoted by $\mathcal{I}_{i,t'}^{\mathring{Th}} \models B_{i,t'}^{\ell,u}(F_t)$) iff:

$$\ell \leq \sum_{Th \in \mathcal{T}, Th(t) \models F} \mathcal{I}_{i,t'}^{\mathring{Th}}(Th) \leq u. \tag{5}$$

2. *(Belief in rules)*
   a temporal rule $r_{\Delta t}(F, G)$ holds (denoted by $\mathcal{I}_{i,t'}^{\mathring{Th}} \models B_{i,t'}^{\ell,u}(r_{\Delta t}(F, G))$) iff:

$$\ell \leq \sum_{Th \in \mathcal{T}} \mathcal{I}_{i,t'}^{\mathring{Th}}(Th) \cdot \mathsf{fr}(Th, F, G, \Delta t) \leq u. \tag{6}$$

   with the function fr *giving the frequency of rule $r_{\Delta t}(F, G)$, i.e.,* fr *divides the number of occurrences where F is followed by G in Δt time units by the total number of occurrences of F in thread Th.*
3. *(Nested beliefs)*
   a belief $B_{j,t}^{\ell_j,u_j}(\varphi)$ of some other agent j holds at time t' (denoted by $\mathcal{I}_{i,t'}^{\mathring{Th}} \models B_{i,t'}^{\ell,u}(B_{j,t}^{\ell_j,u_j}(\varphi))$) iff:

$$\ell \leq \sum_{\substack{Th \in \mathcal{T} \\ \mathcal{I}_{j,t}^{Th} \models B_{j,t}^{\ell_j,u_j}(\varphi)}} \mathcal{I}_{i,t'}^{\mathring{Th}}(Th) \leq u. \tag{7}$$

Note that with respect to this semantics, a belief $B_{i,t}^{1,1}(\varphi)$ with a quantification $\ell = u = 1$ represents certainty. Thus, $B_{i,t}^{1,1}(\varphi)$ represents *knowledge* regarding a fact φ and is therefore equivalent to the established knowledge operator $K_i(\varphi)$ (cf. e.g., [8]).

As the semantics of the belief operator is defined with respect to the subjective posterior interpretations of the respective agent, it is clear that beliefs change according to the interpretation updates as given in Definition 6. As the interpretations are updated with the occurrence of observations, it is clear that the beliefs of an agent can be influenced by ensuring that the respective agent makes certain observations. We will use this below to identify possible actions to induce the abduction goal. A detailed analysis on the resulting belief evolutions over time can be found in [12]. Further detailed examples that illustrate how PDT Logic can be used as a modeling language to formally specify a problem domain are discussed in [13].

A set of belief formulae $\mathfrak{B}$ *entails* a belief formula G (denoted by $\mathfrak{B} \models G$), iff every thread Th in the set of threads $\mathcal{T}$ induced by $\mathfrak{B}$ satisfies G.

3 Abduction in PDT Logic

Given a set of PDT Logic formulae $\mathfrak{B}$ describing a specific scenario, it is often useful to know what actions one could take to induce a certain belief $B_{i,t'}^{\ell,u}(\varphi)$ of some agent at a specific time t'. As the beliefs in PDT Logic change due to observations, it is natural to define possible actions as a set of observations that can be induced.

Definition 9. *Let $\mathfrak{B}$ be a set of PDT Logic formulae, H be a set of PDT Logic formulae representing observations $Obs_{\mathcal{G}}(l)_t$ and let $G \equiv B_{i,t_g}^{\ell,u}(\varphi)$ be an atomic belief formula. Then, the triple $\langle \mathfrak{B}, H, G \rangle$ is an instance of the PDT Abduction Problem. $S \subseteq H$ is a solution to the abduction problem iff $\mathfrak{B} \cup S$ is satisfiable and $\mathfrak{B} \cup S \models G$. A solution S is a minimal solution to the abduction problem if there exists no solution S' with $|S'| < |S|$ so that $\mathfrak{B} \cup S' \models G$.*

Intuitively, $\mathfrak{B}$ constitutes the background knowledge that models a specific environment, G describes the goal we want to achieve, and the hypotheses space H represents information that we can share with the agents in order to induce the belief described by G.

3.1 The Hypotheses Space $\mathcal{H}$

As the background knowledge $\mathfrak{B}$ induces a set of possible threads $\mathcal{T}$ (cf. [13]), we do not need to specify the hypotheses space H explicitly, but instead we can determine a set of hypothesis candidates H' from $\mathcal{T}$ as the set of all observations that can possibly occur:

$$H' = \{Obs_{\mathcal{G}}(l)_t : (\exists Th \in \mathcal{T} : Obs_{\mathcal{G}}(l) \in Th(t))\} \tag{8}$$

Before actually trying to solve the abduction problem specified in Definition 9, we can identify necessary preconditions that an observation $Obs_{\mathcal{G}}(l)_t \in H'$ has to satisfy in order to be able to contribute to a solution of the abduction problem: The set H' collects all observations that can possibly occur in the situation described by $\mathfrak{B}$. However, not all of these observations have the means to alter the quantification of the goal belief G. With a slight abuse of notation, we use $i \in G$ to denote that agent i is involved in the goal belief G, i.e., G contains a belief operator $B_{i,t'}^{\ell,u}$ (possibly as part of a nested belief). Then, we can define a dependency property $dep(G, Obs_{\mathcal{G}}(l)_t)$ between the goal and an observation as follows:

Definition 10 (Goal dependency). *Let G be the abduction goal and let $Obs_{\mathcal{G}}(l)_t$ be an observation. G is dependent on $Obs_{\mathcal{G}}(l)_t$, denoted by $dep(G, Obs_{\mathcal{G}}(l)_t)$, iff*

$$i \in G \ \wedge \ i \in \mathcal{G} \tag{9}$$

Naturally, any observation $Obs_{\mathcal{G}}(l)_t \in H'$ that does not satisfy this dependency property is unable to contribute to achieving the goal and can therefore be neglected when searching for a solution to the abduction problem. Thus, we can define the set of relevant atomic hypotheses as

$$H = \{Obs_{\mathcal{G}}(l)_t \in H' : \ dep(G, Obs_{\mathcal{G}}(l)_t)\} \tag{10}$$

Whenever an observation occurs for some agent i, the set of threads it considers possible is reduced such that only those threads remain where the respective observation holds. We use $K_i^S(t_g)$ to denote the set of possibility relations for agent i at the time t_g of the goal belief[2] induced by a potential solution $S \subseteq H$. We can then leverage the semantics of the belief operator (cf. Definition 8) to obtain another necessary precondition: for $G \equiv B_{i,t_g}^{\ell,u}(\varphi)$ with $0 < \ell$ and $u < 1$, in every distinguishable situation $\mathcal{K}_i(t_g)$ that i considers possible at time t_g, there need to be two threads Th_1, Th_2 so that the respective belief object φ is satisfied in one thread and unsatisfied in another. If the belief is quantified with $\ell = u = 1$, all threads in all distinguishable situations $\mathcal{K}_i(t_g)$ have to satisfy the belief object φ. Otherwise, if these conditions are not met, it is clear that the goal belief is not valid, independently of any specific probability assignment. These conditions can be checked syntactically prior to evaluating the semantic entailment $\mathfrak{B} \cup S \models G$. Using $sp(S)$ to denote the syntactic possibility of a solution S, we can formally express these considerations as

$$sp(S) = \begin{cases} true & \text{if } 0 < \ell, u < 1 \text{ and } \forall \mathcal{K}_i(t_g) \in K_i^S(t_g) : \\ & \quad \exists Th_1, Th_2 \in \mathcal{K}_i(t_g) : Th_1(t_g) \models \varphi \wedge Th_2(t_g) \models \neg\varphi \\ true & \text{if } \ell = u = 1 \text{ and } \forall \mathcal{K}_i(t_g) \in K_i^S(t_g) : \\ & \quad (\forall Th \in \mathcal{K}_i(t_g) : Th(t_g) \models \varphi) \\ false & \text{otherwise} \end{cases} \tag{11}$$

[2] To simplify the presentation, we assume that (even for nested beliefs) the goal formula G involves only a single time point t_g. The proposed methods are also applicable to goal formulae involving multiple time points, but this will significantly increase the complexity of presentation.

With these considerations we can define the entire search space $\mathcal{H}$ for possible solutions to the abduction problem as

$$\mathcal{H} = \{S \in 2^H : \ sp(S)\} \tag{12}$$

3.2 The Abduction Process

To determine whether a candidate solution $S \subseteq \mathcal{H}$ is actually a solution to the abduction problem, i.e., S together with the background knowledge $\mathfrak{B}$ entails the goal G, we can reformulate the entailment problem as a satisfiability problem in the usual way [6], provided that $\mathfrak{B} \cup \{G\}$ is consistent:

$$\mathfrak{B} \cup S \models \{G\} \ \equiv \ \neg sat(\mathfrak{B} \cup S \cup \{\neg G\}) \tag{13}$$

Checking satisfiability of a set of PDT Logic can be performed as described in [13]. The complexity of satisfiability checking is as follows.

Theorem 1 (Complexity of PDT SAT). *Reference [13] Checking satisfiability of a set of PDT Logic belief formulae $\mathfrak{B}$ is NP–complete.*

Building on this result, we obtain the following complexity result for deciding whether a solution exists for an instance of the PDT Logic abduction problem:

Theorem 2 (Complexity of PDT Abduction). *Let $A = \langle \mathfrak{B}, H, G\rangle$ be an instance of the PDT Logic abduction problem. Deciding whether a solution exists is Σ_2^P–complete.*

Proof. Due to space constraints, we only give a proof sketch here. The complete proof works analogously to the proof of Theorem 4.2 in [15].

Showing membership is straightforward: We can guess a potential solution $S \subseteq H$. Using (13) and Theorem 1, it is easy to see that this solution can be verified in polynomial time by querying an NP oracle.

A known Σ_2^P–complete problem [17] is validity checking of a quantified Boolean formula Φ of the form $\exists X \forall Y \psi(X,Y)$ with mutually distinct Boolean variables $X = \langle x_1, ..., x_n\rangle$ and $Y = \langle y_1, ..., y_m\rangle$, respectively and $\psi(X,Y)$ a Boolean formula over the variables x_i and y_j. Intuitively, this problem has a close connection to the PDT Logic abduction problem, as we need to find some assignment to X (i.e., an abductive solution) such that the goal Y is always satisfied. Thus, we use the respective x_i as potential observation objects of the abduction problem, and set $\psi(X,Y)$ as the abduction goal; i.e., we do not restrict the set of possible threads by leaving the background knowledge $\mathfrak{B}$ empty, pick an arbitrary agent a and define hypotheses and goal belief for this agent as follows:

$$\mathfrak{B} = \emptyset, \quad H = \bigcup_{i=1}^{n} \{Obs_a(x_i)_1, Obs_a(\neg x_i)_1\}, \quad S = B_{a,t}^{1,1}(\psi(X,Y))$$

Using this formulation, we can transform validity checks of any Boolean formula Φ of the above form to an instance of the PDT Logic abduction problem and thus show that the problem is Σ_2^P–hard. □

Adapting the approach from [15] by substituting geometric polytope operations with according satisfiability checks, we can identify several distinct cases that will guide the abduction procedure:

Proposition 1. *Let $\langle \mathfrak{B}, H, G \rangle$ be an instance of the PDT Logic abduction problem and let $S \subseteq H$ be a potential solution to this problem. Then, the following observations hold for the abduction problem:*

1. $\neg sat(\mathfrak{B} \cup \{G\})$*, background knowledge and goal are inconsistent. Then, there is no solution to the abduction problem and no hypothesis S has to be tested. Otherwise, if background knowledge and goal are consistent, we can identify the following scenarios:*
2. $\neg sat(\mathfrak{B} \cup \{\neg G\})$*, the background knowledge always entails the goal. Then, $\emptyset$ is already a solution to the abduction problem and no hypothesis S has to be tested.*
3. $\neg sat(\mathfrak{B} \cup S)$*, the potential solution is inconsistent w.r.t. the background knowledge. Then, every potential solution S' with $S \subseteq S' \subseteq H$ is also inconsistent, and therefore cannot be a solution to the abduction problem. Then, we can remove S' from $\mathcal{H}$ to prune the hypotheses space when searching for solutions to the abduction problem.*

The first two checks determine whether it is at all required to search for a solution to the abduction problem. The third case provides a pruning condition for the hypotheses search space $\mathcal{H}$: if a solution candidate is not satisfiable together with the background knowledge, it is futile to test any superset of this solution. Using these properties, we obtain the abduction procedure depicted in Algorithm 1: after checking whether it is required to search for a solution at all (lines 2–5), the procedure iterates through all potential solutions from $\mathcal{H}$, ordered by their respective size (lines 4–15), and prunes the search space whenever some potential solution S is inconsistent w.r.t. the background knowledge (line 14). The procedure terminates if a solution is found or the search space is empty.

Remark 1. Reference [15] provides another pruning condition for abductive reasoning in APT Logic by arguing that for $\mathfrak{B} \cup S \not\models G$ (with $sat(\mathfrak{B} \cup S)$), any subset $S' \subseteq S$ cannot solve the abduction problem, either. This is not applicable in PDT Logic, because beliefs change with additional observations, and thus it is possible that S' is indeed a solution to the abduction problem, while S with additional observations is not.

Iterating through the search space in increasing order with respect to the solution size has to important ramifications: First, it is ensured that any pruning operations due to inconsistent combinations of background knowledge and solution candidates are carried out as early as possible. The smaller the respective solution, the larger is the respective pruned superset and thus, pruning operations are applied most effectively. Second, any solution S returned by Algorithm 1 is a minimal solution to the abduction problem.

Algorithm 1. Abduction Algorithm for PDT Logic

```
 1: procedure ABDUCE($\mathfrak{B}$,$H$,$G$)
 2:     if $\neg sat(\mathfrak{B} \cup G)$ then                    ▷ case 1: $\mathfrak{B} \cup G$ is inconsistent
 3:         return $false$
 4:     if $\neg sat(\mathfrak{B} \cup \neg G)$ then                        ▷ case 2: $\mathfrak{B} \models G$
 5:         return $\emptyset$
 6:     $\mathcal{H} \leftarrow \{S \in 2^H : sp(S)\}$   ▷ init search space as set of syntactically possible solutions
 7:     $i \leftarrow 1$
 8:     while $(\mathcal{H} \neq \emptyset$ and $i \leq |H|)$ do               ▷ test solutions in order of simplicity
 9:         for $S \in \mathcal{H}$ with $|S| = i$ do
10:             if $\neg sat(\mathfrak{B} \cup S \cup \neg G)$ then                     ▷ $S$ is a solution
11:                 return $S$
12:             else
13:                 if $\neg sat(\mathfrak{B} \cup S)$ then   ▷ case 3: $\mathfrak{B} \cup S$ is inconsistent, prune supersets
14:                     $\mathcal{H} \leftarrow \mathcal{H} \setminus \{S' : S' \in \mathcal{H} \wedge S' \supseteq S\}$
15:         $i \leftarrow i + 1$
16:     return $false$
```

Theorem 3. *Let $A = \langle \mathfrak{B}, H, G \rangle$ be an instance of the PDT Logic abduction problem. If A has a solution, then Algorithm 1 returns a minimal solution S so that $\mathfrak{B} \cup S \models G$. Otherwise, the algorithm returns $false$.*

Proof. We start with showing that any set discarded in the pruning step (line 14) cannot be a solution to the abduction problem. If $\mathfrak{B} \cup S$ is unsatisfiable, this set is already overly constrained so that no thread remains that could possibly satisfy all formulae in this set. Then, as observed in Proposition 1, adding further constraints will clearly still result in an empty set of possible threads. Thus, it is unnecessary to test any set $S' \supseteq S$ for possible solutions to the abduction problem.

If the abduction problem has a solution, it is clear that the loop in lines 4–15 will eventually find and return a solution, as all solution candidates are tested iteratively unless they are discarded as above. Since the algorithm iterates over the set of possible solutions by increasing size of the solution, any returned solution S will necessarily be minimal. If there had been a smaller solution S' with $|S'| < |S$, the algorithm would have terminated earlier by returning this solution S'. □

4 Conclusion

In this paper, we have presented how abduction can be formalized in the context of Probabilistic Doxastic Temporal (PDT) Logic. We have discussed how relevant hypotheses space can be determined automatically from a set of threads and have developed a sound and complete algorithm to give a minimal solution to the abduction problem. We have shown that the problem of searching for a solution to the abduction problem is Σ_2^P–complete and we have derived several criteria for effectively pruning the solution search space.

To the best of our knowledge, this is the first work that studies abduction in the context of dynamically evolving beliefs for multi–agent systems, and thus, the methods introduced in this work provide means for novel reasoning capabilities.

References

1. Aumann, R.J.: Agreeing to disagree. Ann. Stat. **4**(6), 1236–1239 (1976)
2. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese **139**(2), 165–224 (2004)
3. Baral, C.: Abductive reasoning through filtering. Artif. Intell. **120**(1), 1–28 (2000)
4. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, 1st edn. Springer, New York (2007)
5. Eiter, T., Gottlob, G.: The complexity of logic-based abduction. J. ACM **42**(1), 3–42 (1995)
6. Etchemendy, J.: Logical Consequence: The Cambridge Dictionary of Philosophy. Cambridge University Press, Cambridge (1999)
7. Fagin, R., Halpern, J.Y.: Reasoning about knowledge and probability. J. ACM **41**, 340–367 (1994)
8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
9. Josephson, J.R., Josephson, S.G. (eds.): Abductive Inference: Computation, Philosophy, Technology. Cambridge University Press, Cambridge (1996)
10. Kripke, S.A.: Semantical considerations on modal logic. Acta Philosophica Fennica **16**(1963), 83–94 (1963)
11. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, New York (1987)
12. Martiny, K., Möller, R.: A probabilistic doxastic temporal logic for reasoning about beliefs in multi-agent systems. In: 2015 Proceedings of the 7th International Conference on Agents and Artificial Intelligence, ICAART 2015. SciTePress (2015)
13. Martiny, K., Möller, R.: PDT logic - a probabilistic doxastic temporal logic for reasoning about beliefs in multi-agent systems. Technical report (2015). http://www.ifis.uni-luebeck.de/index.php?id=publikationen
14. Martiny, K., Motzek, A., Möller, R.: Formalizing agents beliefs for cyber-security defense strategy planning. In: Proceedings of the 8th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2015, 15–17 June 2015, Burgos, Spain (2015)
15. Molinaro, C., Sliva, A., Subrahmanian, V.S.: Super-solutions: succinctly representing solutions in abductive annotated probabilistic temporal logic. ACM Trans. Comput. Logic **15**(3), 18:1–18:35 (2014)
16. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artif. Intell. **94**(12), 7–56 (1997). Economic Principles of Multi-Agent Systems
17. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time (preliminary report). In: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC 1973, pp. 1–9. ACM, New York (1973)