Mental State Abduction of BDI-Based Agents*

M.P. Sindlar, M.M. Dastani, F. Dignum, and J.-J.Ch. Meyer
{michal,mehdi,dignum,jj}@cs.uu.nl

University of Utrecht
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

Abstract. In this paper we present mental state abduction, a technique for inferring the mental states (beliefs, goals) of BDI-based agents from observations of their actions. Abduced mental states are considered to be explanations of observed behavior, which is assumed to be part of a (goal-directed) plan. The observer, who attempts to explain an agent's behavior, is assumed to know all of the agent's behavioral rules that can account for observed actions. Three explanatory strategies are introduced, based on different presumptions regarding perceptory conditions.

1 Introduction

Intelligent computational agents implemented in BDI-inspired programming languages are suited for developing virtual (non-player) characters for computer games and simulations [1]. In this work we tackle the problem of providing explanations for the observed behavior of such BDI-implemented virtual characters in terms of their mental states.

Agents in BDI-based programming languages such as 2/3APL, Jack, Jadex and Jason [2], are programmed in terms of mentalistic notions like beliefs, goals (desires), and plans (intentions). In order for agents to cooperate with or obstruct the plans of other agents, it is a prerequisite that they can draw (defeasible) conclusions about those other agents' mental states, a capacity we refer to as mental state inference. Because agent languages with declarative mental states are often logic-based, and because a logical relation $\mathsf{mentalstate} \Rightarrow \mathsf{behavior}$ can (approximately) be identified, logical abduction is a promising approach to providing explanations of agents' observed behavior in terms of mental state descriptions.

We envision the use of techniques for mental state inference as a means of designing characters that show a higher degree of believability [3]. Agents that incorporate beliefs about others into their plans can be expected to show more *socially aware* behavior, either correctly inferring the goals or beliefs of others and acting accordingly, or arriving at incorrect conclusions based on explanations that are nonetheless plausible, giving rise to a form of erroneous behavior which the user understands and tolerates. Based on their hypotheses, agents can also form expectations with respect to other agents' future actions.

* This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

The structure of this paper is as follows. In Section 2 we sketch the outline of our approach, and relate it to existing similar work. Basic notions are introduced in Section 3, and put to use in the next Section 4 to define the technique of mental state abduction. In Section 5 this technique is illustrated through a simple example, and we conclude by discussing our results and pointing out possible future work in the last Section 6.

2 Our approach and related work

In this paper, we investigate an approach to inferring the mental state of BDI agents based on their observed behavior, and take the agent programming language 2APL [4] as a departure point for our investigations. We have tried to make the framework as general as possible, though, in the hope that it will apply to any agent programming language in which declarative mental states drive the operation of the agent. Also, we explicitly mention that our intention is not to capture the full dynamics of current agent programming languages, but instead to present a simplified general language for agent specification.

One assumption we make is that the actions of agents are observable as atomic, unambiguous facts. Most of the accepted definitions of the concept 'agent' presume operation in some dynamic, uncertain environment. If this environment is the real world, then the assumption that actions can be unambiguously observed as discrete facts is disputable, to say the least. However, since we consider virtual characters that operate in a game or simulation setting, the environment as well as the virtual characters (agents) are under our direct control. Therefore we have the choice of how we represent actions, and also the perception of actions. In the present case, we think of agents as performing atomic external actions, the execution of which is presented to observers in the form of discrete perceptional events.

Not all actions are necessarily perceived. The environment might determine that the observer will not receive a perceptory event because the action occurs outside its perception radius, or perception might simply fail with some predetermined or context-dependent probability. For our case, we assume that if an action is perceived, this occurs as an event that is an unambiguous representation of this action.

BDI agents operate by means of goal-directed behavioral rules, which are selected and executed by way of some deliberation process. One strong assumption we make is that the observing entity, which attempts to infer an agent's mental state, has knowledge of the entire set of operational rules of this agent. At first inspection this assumption might seem unrealistic, but for the sake of clarity and simplicity we have adopted it. In future work, we hope to relax this requirement, which would mean that some observations might not be explainable at all because no known rule can account for them, or that erroneous conclusions are reached because some known rule was used to abduce an explanation in a case where an unknown rule actually produced the action.

The classical logical approach to explaining observations is abduction [5], which is perhaps best described as reasoning from observation to explanation. We have drawn inspiration from work on (classical) abduction in propositional logic in our approach to the problem of explaining the actions of agents in terms of their mental states. For this approach to work, the assumption has been made that it is possible to represent the procedural rules of agents as logical implications. It is a debatable assumption, but one we think we can successfully defend, as we will attempt to do in Section 3.2. Inferring the mental states of agents based on their observed behavior is what we call *mental state abduction*. Before going into the details of this technique, we will explain how the concept of abduction is generally understood.

2.1 Abduction

Logicians agree about the existence of several distinct modes of reasoning, of which *deduction*, *induction*, and *abduction* [6] are complementary. Deduction is the classical syllogism of modus ponens $\{\phi, \phi \rightarrow \psi\} \models \psi$, where ψ can be validly inferred if both ϕ and $\phi \rightarrow \psi$ are true. Induction can be generalized to the inference of $\phi \rightarrow \psi$ as a rule from the observation of ϕ followed by ψ, and abduction to the inference of ϕ from knowledge of the rule $\phi \rightarrow \psi$ and the observation ψ. Deduction is the only analytic form of inference, whereas induction and abduction both are defeasible, meaning that new observations can invalidate prior conclusions. With indefeasible inference we cannot use the classical entailment relation between premise and conclusion, instead we use some non-monotonic relation, for instance $\rightsquigarrow$. The inductive and abductive syllogisms are $\{\phi, \psi\} \rightsquigarrow (\phi \rightarrow \psi)$ and $\{\phi \rightarrow \psi, \psi\} \rightsquigarrow \phi$, respectively.

The process of abduction is best described as *reasoning from observation to explanation*. The fact that an observed phenomenon ψ (called the *explanandum*) requires explanation, is taken to mean that this phenomenon does not follow directly from a body of knowledge, the background theory Θ. In logical terms, $\Theta \not\models \psi$. Moreover, for some fact ϕ to count as an explanation, a restriction is often enforced that the so-called *explanans* ϕ does not account for the explanandum on its own, so $\phi \not\models \psi$. There are also issues of consistency; the explanans ϕ should not be contradicted by the theory, so $\Theta \cup \phi \not\models \bot$ (or some other criterion for consistency, allowing for revision of the background theory). The schema for abductive reasoning, where ϕ explains ψ with respect to the background theory Θ, is then defined by the following conditions [7]:

Definition 1 (abductive explanation). *The conditions which define when a fact ϕ (the explanans) qualifies as a valid abductive explanation for an observed fact ψ (the explanandum).*

$$\Theta \cup \phi \models \psi$$
$$\Theta \cup \phi \not\models \bot$$
$$\Theta \not\models \psi \quad \textit{and} \quad \phi \not\models \psi$$

Abductive inference, where observing ψ leads to inference of ϕ as an explanation w.r.t. background theory Θ, is defined by these conditions.

$$\Theta, \psi \rightsquigarrow \phi \quad \text{iff the conditions in Definition 1 apply.}$$

In logic programming, the restriction that explanations should belong to a special set of so-called *abducibles* is often enforced, to prevent explaining observations in terms of other effects instead of causes, and for reasons of computational efficiency [8,9]. We will use this restriction in our approach as well, to limit the space of candidate explanations.

It should be noted that the notion of abduction is a murky one, and there is no general agreement on its definition [9,10,11]. Some logicians only discern between deduction and induction, and see what we have called abduction as belonging to the latter. What is important for this account, though, is not so much complete knowledge of these inferential concepts, but a precise definition of these concepts *as we understand and use them*. In Section 4 we define mental state abduction, and link it to classical abduction as presented here.

2.2 Related work

This work is largely related to work in the area of plan and intention recognition; see [12] for an overview of this field. In plan recognition, a common dichotomy is that of opposing *keyhole recognition* to *intended recognition*. The former deals with the recognition of an agent's plan through unobtrusive observation, implying that the agent is unaware of the fact that it's being observed. The latter allows for the agent to actively thwart the inference of its intentions by performing misleading actions. Keyhole recognition best describes our approach, as we don't deal with deception on part of the agent in the form of misleading actions. However, we are able to deal with an agent that tries to hide its actions from sight. The robustness of a plan recognition technique is described in [12] as its ability to deal with incomplete observations. Incomplete perception would occur if an agent were actively trying to prevent the perception of its actions, and this is something we deal with explicitly by proposing several explanatory strategies in Section 4.2.

Albrecht et al. [13] use a Bayesian approach to identify users' plans and goals in an adventure game setting. For this to work, extensive data of users' actions and target quests is processed, and probability distributions are extracted in the form of dynamic belief networks. Such an approach might work for applications where such data is available, but most often this is not the case, simply because logs of users' actions and motivations are not recorded. Also, for inter-agent plan recognition, it is easier to use the agent implementation as a starting point.

In [14], Appelt and Pollack present an approach to abductive reasoning called weighted abduction, which they use to ascribe mental states to an agent based on observed actions. Their approach makes use of inference weights to compare competing explanations. The set of assumptions that lead to the lowest cost proof is selected as explanation. Apart from the obvious similarity in the use

of abduction to infer mental states, their approach and ours differ significantly. Our starting point is the inference of mental states of characters in games or simulations, which are implemented in some variety of BDI agent programming language. Such assumptions are not present in Appelt and Pollack's approach. Moreover, the use of inference weights to compare explanations does not readily apply in our case, as such weights would depend on the procedural mechanism underpinning agents' reasoning, and the context in which observed actions took place. The idea of using (dynamic) weights to compare and select explanations is, however, something we consider for future work.

One other approach that is grounded in an agent programming methodology is that of Goultiaeva and Lespérance [15], which is based on the situation calculus, and ConGolog specifically. It details a technique for recognizing the plan an agent is performing, and corresponds to our approach insofar that inference is based on observed actions and a library of known plans. It also supports incremental recognition: each new perception narrows the space of possible explanations. However, it is limited in the respect that it requires all actions to be observed, and is not robust in the terms of [12] presented above. Moreover, it stops with *recognizing* plans, whereas we provide explanations (in terms of mental states) that represent the *reasons why* an agent performed its actions. This allows for the observing party to intervene, either cooperatively or obtrusively.

3 Basic notions and terminology

In this section, we present the syntax of a simplified logic-based agent programming language, with programming constructs for implementation of rational BDI-agents. Also, we present the syntax of a language for programming an abstract observer entity, which serves as a proof of concept for the inferential mechanism we introduce. This separation of observer and agent serves to clarify the interaction between the roles of agent and observer, for ourselves as well as the reader. Ultimately, though, we envision the possibility of integrating the abductive faculties of the observer with the general reasoning mechanism of agents.

3.1 Plans and behavioral rules

Let $\mathcal{L}$, with typical element ϕ, be a propositional language with negation $\neg$ and the standard conjunctional connective $\wedge$. Belief formulas β and goal formulas κ are elements from $\mathcal{L}$. Also, assume a set $\mathcal{A}$ of basic actions, with typical element α. Agents' plans are then represented as a sequence of basic actions α, tests on propositional formulas, and an operator for non-deterministic choice.

Definition 2 (plans). *Plan expressions $\mathcal{L}_\Pi$ with typical element π are made up of elementary actions $\alpha \in \mathcal{A}$, tests on propositional formulas ϕ, non-deterministic choice between plans, or a sequential composition of plan elements.*

$$\pi ::= \alpha \mid \phi? \mid \pi_1 + \pi_2 \mid \pi_1; \pi_2$$

In this work we are mainly concerned with the syntactical form of those plans, and do not define the semantics of the agent's operation. We do, however, make some assumptions about how an agent executes plans. Only external actions can be observed, and for the present discussion we have defined all actions $\alpha \in \mathcal{A}$ to be external, and therefore observable. Internal test actions $\phi?$ on propositional formulas of type $\phi \in \mathcal{L}$ are not observable. The choice operator $+$ is a non-deterministic choice operator, which has scope over two elements $\pi \in \mathcal{L}_\Pi$. It can also be used to represent deterministic instructions of the form `if` ϕ `then` π_1 `else` π_2 as $(\phi?;\pi_1)+(\neg\phi?;\pi_2)$. For completeness we introduce an empty or `null` action ϵ, such that $(\pi;\epsilon) \equiv (\epsilon;\pi) \equiv \pi$.

Agents' behavioral (planning) rules, of the form $\kappa \leftarrow \beta \mid \pi$, govern the selection and adoption of a plan π, which is appropriate for achieving a specific goal κ given that some condition β holds.

Definition 3 (rules). *Rules $\mathcal{L}_\mathrm{R}$ with typical element r are defined as follows, where $\beta, \kappa \in \mathcal{L}$, and $\pi \in \mathcal{L}_\Pi$.*

$$\mathsf{r} ::= \kappa \leftarrow \beta \mid \pi$$

The selection of these rules is done by the agent's deliberation process. Again, we do not define the semantics of this process, but instead make some explicit assumptions about how this process operates. We refer to [4] for a more precise treatment of those assumptions. We assume that an agent selects a plan only if it is not actively dealing with some other plan. This means that at no point an agent is executing more than one plan. Moreover, we assume that an agent only executes a plan to achieve some goal it explicitly has. Also, plans may be discarded, either because the goal for which they were selected has been achieved, or has been dropped.

The reasons for postulating these assumptions is to ensure some properties we require for the technique of mental state abduction to work. They make sure that an agent always performs its actions to achieve the single goal it is pursuing at that moment; in other words, an agent does not execute plans concurrently. The requirement of the agent only performing actions to achieve some goal — therefore not responding to external events — is to ensure that there is always *some* explanation for an action, since every action is produced by a rule with a mental state as precondition. We do allow for an agent discarding its plan for whatever reason, as we see the ability to deal with such a circumstance as a requirement for a plan recognition approach to be considered robust.

3.2 Observables, abducibles, and rule descriptions

The observer is defined as an abstract entity — insofar that it does not act upon the environment — which perceives the actions of the agent, and attempts to come up with a plausible explanation. The observer's perception is a sequence of actions as performed by the agent. Perception might be complete in the sense that all the actions which the agent has performed are perceived, or some actions might not be perceived and thus perception might be incomplete.

Definition 4 (percepts). *With α as the typical element of a set of basic actions $\mathcal{A}$, the perception language $\mathcal{L}_\Delta$ with typical element δ defines the perception of the observer.*

$$\delta ::= \alpha \mid \delta_1; \delta_2$$

As mentioned in Section 2.1, in computational abduction candidate explanations are constrained to a set of explicitly defined abducibles. In the present context, because we want to abduce the mental states that underlie an agent's actions, these abducibles are the agent's beliefs and goals. Abducible belief and goal formulas are predicated by belief and goal, respectively. Because conjunction between propositions is allowed in belief as well as goal formulas, this distinction is needed to discern between beliefs and goals in the process of abduction.

Definition 5 (abducibles). *If $\beta, \kappa \in \mathcal{L}$, let λ be the typical element of the formulae of abducibles $\mathcal{L}_\Lambda$, and be defined as follows.*

$$\lambda ::= \mathsf{belief}(\beta) \mid \mathsf{goal}(\kappa) \mid \mathsf{belief}(\beta) \wedge \mathsf{goal}(\kappa)$$

To reason about an agent's mental state based on its observed behavior, the observer needs a description of the agent's rules of operation. Since the rules as such have meaning only in their original agent programming context, and we wish to use them in logic, a translation is needed. We present the translated rules as implications which state that a plan is implied by its (mental) preconditions.

Definition 6 (rule descriptions). *If $\lambda \in \mathcal{L}_\Lambda$ and $\pi \in \mathcal{L}_\Pi$, then $\mathsf{rd} \in \mathcal{L}_{\mathrm{RD}}$ is a rule description, and is defined as follows.*

$$\mathsf{rd} ::= \lambda \rightarrow \mathsf{plan}(\pi)$$

One might argue that such a description is imperfect with respect to the actual execution of those rules by the agent, as it does not take the deliberation process into account. This is absolutely true, because if some precondition would be satisfied for multiple plans, then logically all these plans would have to be applied simultaneously. It goes without saying that in actual agent execution this would be quite disastrous! Moreover, a belief precondition β might cease to hold after the plan has been selected and execution started.

However, in the context of *explaining* observed behavior, such a description is sufficiently accurate. Better still, the generality which would be undesirable with respect to execution, actually serves to benefit in the case of explanation. This is because in the latter case the reasoning goes from observation to precondition, and if multiple explanations (mental preconditions) can account for some (partially) observed plan then this is all the more desirable, considering that these preconditions *could* indeed apply from the viewpoint of the observer.

This is not to say that taking into account procedural specifics would not aid mental state inference. If such specifics — which might for example entail that some rule is more likely to have been applied because of the way the agent's underlying reasoning mechanism works — are available, then they can, and probably should, be used. However, since our goal is to provide a general account, we omit such specifics here.

4 Mental state abduction

In this section, we provide the details of our framework for mental state abduction. Before moving on to more high-level concepts, we introduce some prerequisite functions and relations which will be used later on to define the primary notions of our approach.

4.1 Functions and relations

First, we introduce a trace generator function τ, which takes a single plan expression π, as defined in Definition 2, as its input, and maps to a set of perception expressions δ. These perception expressions can be thought of as the possible *observable traces* of the execution of the plan π. Internal test actions on propositional formulas are discarded (mapped to the empty action ϵ) and do not appear in the trace. Where choice between plans occurs, two traces are generated; one for each plan in the scope of the choice operator. The operator $\cup$ used here is set union, and $\circ : \wp(\mathcal{L}_\Delta) \times \wp(\mathcal{L}_\Delta) \to \wp(\mathcal{L}_\Delta)$ is a non-commutative composition operator, which is defined as $\Phi \circ \Psi = \{\phi; \psi \mid \phi \in \Phi \text{ and } \psi \in \Psi\}$, and $\Phi, \Psi \subseteq \mathcal{L}_\Delta$.

Definition 7 (trace generator function). *The function $\tau : \mathcal{L}_\Pi \to \wp(\mathcal{L}_\Delta)$ is a trace generator function that translates plan expressions $\pi \in \mathcal{L}_\Pi$ to sets of perception expressions $\delta \in \mathcal{L}_\Delta$, and is defined as follows.*

$$\begin{aligned} \tau(\alpha) &= \{\alpha\} & \tau(\pi_1 + \pi_2) &= \tau(\pi_1) \cup \tau(\pi_2) \\ \tau(\phi?) &= \{\epsilon\} & \tau(\pi_1; \pi_2) &= \tau(\pi_1) \circ \tau(\pi_2) \end{aligned}$$

Not only are we interested in the observable traces of a plan, we are also interested in partial traces that are structurally related to the original trace. For this we specify the prefix, suffix, subsequence and dilution relations $\preccurlyeq$, $\succcurlyeq$, $\sqsubseteq$, and $\lesssim$, respectively, which are all partial orders (reflexive, antisymmetric, transitive relations) on $\mathcal{L}_\Delta$. Note that $\sqsubseteq$ subsumes $\preccurlyeq$ and $\succcurlyeq$, and $\lesssim$ subsumes $\sqsubseteq$.

Definition 8 (prefix, suffix, subsequence relations). *The prefix relation $\preccurlyeq$, suffix relation $\succcurlyeq$, and subsequence relation $\sqsubseteq$ are partial orders on the domain $\mathcal{L}_\Delta$, and are defined as follows.*

$$\begin{aligned} \preccurlyeq &= \{\ (\delta_1\ ,\ \delta_1; \delta_2) \quad \mid \delta_1, \delta_2 \in \mathcal{L}_\Delta \quad \} \\ \succcurlyeq &= \{\ (\delta_1\ ,\ \delta_2; \delta_1) \quad \mid \delta_1, \delta_2 \in \mathcal{L}_\Delta \quad \} \\ \sqsubseteq &= \{\ (\delta_2\ ,\ \delta_1; \delta_2; \delta_3) \mid \delta_1, \delta_2, \delta_3 \in \mathcal{L}_\Delta \} \quad \cup \quad \preccurlyeq \quad \cup \quad \succcurlyeq \end{aligned}$$

Definition 9 (dilution relation). *The dilution relation $\lesssim$ is a partial order on the domain $\mathcal{L}_\Delta$, defined as the closure of $\sqsubseteq$ under the operator $\mathsf{dc} : (\mathcal{L}_\Delta \times \mathcal{L}_\Delta) \times (\mathcal{L}_\Delta \times \mathcal{L}_\Delta) \to \mathcal{L}_\Delta \times \mathcal{L}_\Delta$, such that $\lesssim = \bigcup_{n \in \mathbb{N}} \mathsf{cl}_n(\sqsubseteq)$.*

$$\begin{aligned} &\mathsf{dc}((\delta_1, \delta_2), (\delta_3, \delta_4)) = (\delta_1; \delta_3, \delta_2; \delta_4) \\ &\mathsf{cl}_0(\sqsubseteq) = \sqsubseteq \\ &\mathsf{cl}_n(\sqsubseteq) = \mathsf{cl}_{n-1}(\sqsubseteq) \cup \{\ \mathsf{dc}((\delta_1, \delta_2), (\delta_3, \delta_4)) \mid (\delta_1, \delta_2), (\delta_3, \delta_4) \in \mathsf{cl}_{n-1}(\sqsubseteq)\ \} \end{aligned}$$

The dilution relation might be somewhat difficult to understand. What it states is that a dilution of some formula can be obtained by randomly removing any number of elements from the original perception expression, whilst preserving the order. In Definition 10 we use the structural relations to define so-called relational functions, which map an expression that is given as input to the set of perception expressions related to it under one of the three structural relations. The output is the set of all prefixes / subsequences / dilutions of the expression that was given as input. Note that the suffix relation $\preccurlyeq$ is not used here explicitly, but only servers as part of the definition of the subsequence relation.

Definition 10 (relational functions). *The relational functions* $\mathsf{prefix} : \mathcal{L}_\Delta \to \wp(\mathcal{L}_\Delta)$, $\mathsf{subseq} : \mathcal{L}_\Delta \to \wp(\mathcal{L}_\Delta)$, *and* $\mathsf{dilute} : \mathcal{L}_\Delta \to \wp(\mathcal{L}_\Delta)$ *map perception expressions* $\delta \in \mathcal{L}_\Delta$ *to sets of perception expressions, and are defined as follows.*

$$\begin{aligned} \mathsf{prefix}(\delta) &= \{\delta' \mid \delta' \preccurlyeq \delta\} \\ \mathsf{subseq}(\delta) &= \{\delta' \mid \delta' \sqsubseteq \delta\} \\ \mathsf{dilute}(\delta) &= \{\delta' \mid \delta' \lesssim \delta\} \end{aligned}$$

4.2 Partial traces for explanatory strategies

Now that the basic notions have been defined, it is time to put them to use in our quest for explanations of observed behavior. First of all, though, three perceptory presumptions are considered, which reflect the observer's assumptions about either the environment or its own status with respect to perception of the agent's actions.

Definition 11 (perceptory presumptions). *The following presumptions with respect to perceptory conditions are distinguished.*

- ***Complete observation:*** *If complete observation is presumed, then the observer expects to see every action the agent performs.*
- ***Late observation:*** *Late observation reflects the observer's presumption that it has possibly failed in seeing one or more of the initial actions the agent performed. From the moment it starts observing the observer expects to see every future action of the agent.*
- ***Partial observation:*** *In the case of partial observation, the observer presumes that it might fail to see some of the actions the agent performs. Such failure might occur due to some environmental circumstance, or due to the agent deliberately obscuring its actions.*

The perceptory presumptions can be used to model the observer's expectations with respect to what it can and will perceive. Late observation, for instance, reflects that fact that the observer believes it has arrived 'late', and has missed some actions the agent has performed.[1] Partial observation can reflect the fact

[1] Note that 'late' refers to the observer possibly observing the plan after execution has started, and not to some temporal delay in perception.

that the observer has to divide its attention among several tasks, on which it bases its presumption that it will miss out on some perceptions. But perceptory presumptions can also reflect beliefs about properties outside the agent's control. Partial observation, again, might reflect the observer's belief that the agent is hiding its actions, or it might reflect incompleteness of perception due to the nature of the environment. Note that complete and late observation have in common that the observer expects to see every action of the agent.

The descriptions of the agent's operational rules capture a logical connection between the agent's mental state and some plan. To abduce the mental state based on those descriptions, the observer somehow has to compare the sequence of actions it has perceived to some plan it believes the agent might be executing. This is where the trace generator function and the relational functions come together. Combined, they allow for generating partial traces which are structurally related, under one of the three relations $\preccurlyeq$, $\sqsubseteq$, and $\lesssim$, to the set of complete traces of a plan as produced by the plan translation function τ.

Those partial traces fit in nicely with the perceptory presumptions. In the case of complete observation, the observer expects to see every action the agent performs. This means the first action the observer has perceived should be the first action of one or more of the traces of the plan the agent is executing, and of which the observer should have information in its description of the agent's rules. Likewise, any sequence of actions the observer perceives should be the prefix of one of those traces. Late observation presumes the observer might have missed the initial actions of some plan, but has seen any actions performed since it started observing. Therefore, any observed sequence must be a subsequence of some plan trace. Partial observation presumes the agent has seen some actions of a plan — in the same order as they occurred in the plan — but might have missed others. It will be no surprise that in this case the observed sequence is a dilution of some plan trace.

Putting all this together, we define the functions in Definition 12, which take as input a plan expression, and generate the set of partial traces which capture the previously mentioned perceptory presumptions. For this reason, the functions are indexed with c in the case of complete observation, l for late observation, and p for partial observation.

Definition 12 (partial trace generator functions). *The partial trace generator functions* $\mathsf{traces_c} : \mathcal{L}_\Pi \to \wp(\mathcal{L}_\Delta)$*,* $\mathsf{traces_l} : \mathcal{L}_\Pi \to \wp(\mathcal{L}_\Delta)$*, and* $\mathsf{traces_p} : \mathcal{L}_\Pi \to \wp(\mathcal{L}_\Delta)$ *translate plan expressions* $\pi \in \mathcal{L}_\Pi$ *to sets of perception expressions, and are defined as follows.*

$$\mathsf{traces_c}(\pi) = \bigcup \{\ \mathsf{prefix}(\delta) \mid \delta \in \tau(\pi)\ \}$$

$$\mathsf{traces_l}(\pi) = \bigcup \{\ \mathsf{subseq}(\delta) \mid \delta \in \tau(\pi)\ \}$$

$$\mathsf{traces_p}(\pi) = \bigcup \{\ \mathsf{dilute}(\delta) \mid \delta \in \tau(\pi)\ \}$$

The output of $\mathsf{traces_c}(\pi)$ is the set containing every prefix of each trace of the plan π. The output of $\mathsf{traces_l}(\pi)$ is the set of subsequences of each trace of π, and $\mathsf{traces_p}(\pi)$ maps all traces of the plan expression π to the set of its dilutions.

4.3 Agent and observer

Here we define the configurations of the agent and the observer. We intentionally don't specify the semantics of the agent program formally, but present the agent configuration using similar terminology as [4], where possible semantics of agent operation are defined.

Definition 13 (agent configuration). *$\langle \sigma, \gamma, \Pi, \mathcal{R} \rangle$ is an agent configuration, defined as a tuple of a belief base $\sigma \subseteq \mathcal{L}$, a goal base $\gamma \subseteq \mathcal{L}$, the agent's plan base $\Pi \subseteq \mathcal{L} \times \mathcal{L}_\Pi$, and a set of operational rules $\mathcal{R} \subseteq \mathcal{L}_\mathrm{R}$.*

The configuration of an agent consists of three dynamic structures: the belief base σ, consisting of propositional facts expressing the agents beliefs, the goal base γ which consists of propositional facts that express the state of affairs the agent wishes to realise, and a plan base Π consisting of a tuple of goals and plans, which links a plan to the goal for which it has been generated. The only static structure (meaning it does not change during agent execution), and the one we are most interested in, is the agent's set of operational rules $\mathcal{R}$, as described in Section 3.1. From this set $\mathcal{R}$ of the agent's rules it is that the configuration of the observer is generated.

Definition 14 (observer configuration). *The configuration of the observer, is a tuple $\langle \Delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{c}, \Xi_\mathsf{I}, \Xi_\mathsf{p} \rangle$ of a perceptory base $\Delta \in \mathcal{L}_\Delta \times \mathcal{L}_\Delta$, rule descriptions $\mathcal{RD} \subseteq \mathcal{L}_\mathrm{RD}$, pre-specified abducibles $\Lambda \subseteq \mathcal{L}_\Lambda$, and sets of partial plan traces $\Xi_\mathsf{c}, \Xi_\mathsf{I}, \Xi_\mathsf{p} \subseteq \mathcal{L}_\Delta \times \mathcal{L}_\Pi$.*

$$
\begin{aligned}
\mathcal{RD} &= \{\ \mathsf{goal}(\kappa) \wedge \mathsf{belief}(\beta) \rightarrow \mathsf{plan}(\pi) \mid (\kappa \ \leftarrow \ \beta \mid \pi) \in \mathcal{R} \ \} \\
\Lambda &= \{ \quad \lambda \qquad \mid (\lambda \rightarrow \mathsf{plan}(\pi)) \in \mathcal{RD} \ \} \\
\Xi_\mathsf{c} &= \{\ (\delta, \pi) \quad \mid (\lambda \rightarrow \mathsf{plan}(\pi)) \in \mathcal{RD} \quad \& \quad \delta \in \mathsf{traces_c}(\pi) \ \}
\end{aligned}
$$

The sets Ξ_I and Ξ_p are defined similarly to Ξ_c, but are dependent on $\mathsf{traces_I}$ and $\mathsf{traces_p}$, respectively. We present them here as static precomputed entities for the sake of simplicity, although in practice an online algorithm would be preferred for the sake of resources. The only dynamic structure in the observer configuration is the base of incoming perceptory events Δ, of which we will speak more in the next section. The static structures can all be generated from the agent's set of rules $\mathcal{R}$, because these rules contain the necessary information for explaining the agent's behavior.

4.4 Explanation of observed actions

In Definition 14, we mentioned the observer's perceptory base Δ, without defining its actual content. Assume $\Delta = (\delta, \alpha)$, by which we mean that Δ is tuple of some sequence of actions δ the observer has remembered[2], and the last perceived action α, which the observer receives as an event. The occurrence of a

[2] We assume that memory can hold a sequence of arbitrary length.

perception event, signifying that action α has been observed, is what triggers explanation. We now compare mental state abduction to logical abduction as presented in Section 2.1, and consider the following schema in analogy to the one in Definition 1.

$$\begin{array}{ll} \Theta \cup \phi \models \psi & \mathcal{RD} \cup \lambda \models \delta \\ \Theta \cup \phi \not\models \bot & \mathcal{RD} \cup \lambda \not\models \bot \\ \Theta \not\models \psi \text{ and } \phi \not\models \psi & \mathcal{RD} \not\models \delta \text{ and } \lambda \not\models \delta \end{array}$$

Most of the conditions that apply to classical abduction do not apply to the specific case of mental state abduction. There's no danger of having (logically) contradictory explanations, nor can the perceived action sequence δ directly follow from the rule descriptions or a lone abducible. But — and this is more worrying — the perceived action sequence can't even be explained from the set of rule descriptions $\mathcal{RD}$ and some abducible λ put together! Fortunately, it is possible to explain something of the form $\mathsf{plan}(\pi)$.

$$\Theta \cup \phi \models \psi \qquad\qquad \mathcal{RD} \cup \lambda \models \mathsf{plan}(\pi)$$

To explain an observation, the only thing left to do now is to relate observed actions of the form δ to a plan descriptor of the form $\mathsf{plan}(\pi)$. The mechanism for this is already in place; we can use the tuples (δ, π) of partial plan traces and plans in Ξ, as defined in Definition 14. The careful reader will have noticed that the sets of partial plan traces are indexed with c, l, and p, and indeed these correspond to the partial traces suited for relating observed sequences to plans, under the presumptions of complete, late, and partial observation, respectively.

To actually generate explanations based on some observation, we define three explanatory functions expl_c, expl_l, and expl_p, which generate the set of explanations that account for the observed sequence of actions, under the perceptory presumptions of complete, late, and partial observation. Thus, if some observed sequence of actions δ is a subsequence but *not* a prefix of one of the traces of plan π, then the explanation $\lambda \in \Lambda$ for which $\mathcal{RD} \cup \lambda \models \mathsf{plan}(\pi)$ will be in the set of explanations given by expl_l and expl_p, but not in the set given by expl_c (or only as explanation based on another plan π' of which δ *is* a prefix).

Definition 15 (explanatory functions). *Explanatory functions* $\mathsf{expl}_\mathsf{c} : \mathcal{L}_\Delta \times \wp(\mathcal{L}_{\mathrm{RD}}) \times \wp(\mathcal{L}_\Lambda) \times \wp(\mathcal{L}_\Delta \times \mathcal{L}_\Pi) \to \wp(\mathcal{L}_\Lambda)$, $\mathsf{expl}_\mathsf{l} : \mathcal{L}_\Delta \times \wp(\mathcal{L}_{\mathrm{RD}}) \times \wp(\mathcal{L}_\Lambda) \times \wp(\mathcal{L}_\Delta \times \mathcal{L}_\Pi) \to \wp(\mathcal{L}_\Lambda)$, *and* $\mathsf{expl}_\mathsf{p} : \mathcal{L}_\Delta \times \wp(\mathcal{L}_{\mathrm{RD}}) \times \wp(\mathcal{L}_\Lambda) \times \wp(\mathcal{L}_\Delta \times \mathcal{L}_\Pi) \to \wp(\mathcal{L}_\Lambda)$, *map to a finite set of abducibles* $\lambda \in \Lambda$, *given a finite perception* $\delta \in \mathcal{L}_\Delta$, *a finite set of rule descriptions* $\mathcal{RD} \subseteq \mathcal{L}_{\mathrm{RD}}$, *and a finite set of abducibles* $\Lambda \subseteq \mathcal{L}_\Lambda$.

$$\begin{aligned} \mathsf{expl}_\mathsf{c}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{c}) &= \{\ \lambda \in \Lambda \mid (\mathcal{RD} \cup \lambda) \models \mathsf{plan}(\pi)\ \&\ (\delta, \pi) \in \Xi_\mathsf{c}\ \} \\ \mathsf{expl}_\mathsf{l}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{l}) &= \{\ \lambda \in \Lambda \mid (\mathcal{RD} \cup \lambda) \models \mathsf{plan}(\pi)\ \&\ (\delta, \pi) \in \Xi_\mathsf{l}\ \} \\ \mathsf{expl}_\mathsf{p}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{p}) &= \{\ \lambda \in \Lambda \mid (\mathcal{RD} \cup \lambda) \models \mathsf{plan}(\pi)\ \&\ (\delta, \pi) \in \Xi_\mathsf{p}\ \} \end{aligned}$$

These explanatory functions can be used as part of different explanatory strategies, which reflect the presumptions defined in Def. 11. For example, if an observer believes it will perceive all actions of an agent which is known to have started execution of some plan, it might consider only those explanations provided by expl_l, and not those provided by expl_c.

An observer might also try different strategies, to see what works best. For example, it might be the case that the presumption of complete observation yields no observation. The observer then might try explanation using some other function, the success or failure of which can shape its ideas about the behavior of the agent and/or the nature of the environment.

4.5 Propositions and proofs

In this section we state some important properties of our technique, and provide formal proof of their veracity. In Proposition 1 we claim that some explanatory functions are guaranteed to yield more explanations than others, and in Proposition 2 we claim that the number of explanations decreases monotonically, as more actions are perceived.

Proposition 1. *The number of explanations generated by the function* expl_c *is less or equal to that of the function* expl_l*, which is in turn less or equal to that of the function* expl_p*.*

Proof. The explanatory functions expl_c*,* expl_l*, and* expl_p*, as defined in Def. 15, yield a finite set of elements* $\lambda \in \mathcal{L}_\Lambda$*, given some finite input. They differ only in the fact that they are based on three different sets of tuples of perception expressions and plan expressions;* Ξ_c *for* expl_c*,* Ξ_l *for* expl_l*, and* Ξ_p *for* expl_p*. As follows from Def. 10, 12, and 14, the defining characteristic of these tuples are the relations presented in Def. 8 & 9. From these definitions it follows that* $(\preccurlyeq) \subseteq (\sqsubseteq) \subseteq (\lesssim)$*, and because these relations define the sets of partial plan traces* Ξ_c*,* Ξ_l*, and* Ξ_p*, respectively, it follows that* $\Xi_\mathsf{c} \subseteq \Xi_\mathsf{l} \subseteq \Xi_\mathsf{p}$ *and* $|\Xi_\mathsf{c}| \leq |\Xi_\mathsf{l}| \leq |\Xi_\mathsf{p}|$*. Given any finite perception expression* δ*, finite set of rule descriptions* $\mathcal{RD}$*, and finite set of abducibles* Λ*, let* $\mathsf{expl}_\mathsf{c}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{c}) = C$*,* $\mathsf{expl}_\mathsf{l}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{l}) = L$*, and* $\mathsf{expl}_\mathsf{p}(\delta, \mathcal{RD}, \Lambda, \Xi_\mathsf{p}) = P$*. It then follows that* $|C| \leq |L| \leq |P|$*.* □

Lemma 1. *For any relation* $R \in \{\preccurlyeq, \sqsubseteq, \lesssim\}$*, if* $(\delta'; \delta'', \delta) \in R$*, then* $(\delta', \delta) \in R$*.*

Proof. It follows from the definition of the relations that $(\delta', \delta'; \delta'') \in R$*, because every relation subsumes* $\preccurlyeq$*. If this is the case, then it follows by transitivity that if* $(\delta', \delta'; \delta'') \in R$ *and* $(\delta'; \delta'', \delta) \in R$*, then* $(\delta', \delta) \in R$*.* □

Proposition 2. *The number of explanations under any single explanatory relation either decreases or stays the same (decreases monotonically) when* α *is explained as the next action of an already explained sequence of actions* δ*.*

Proof. Take some explanatory function expl*. Assume some perception* δ *has been observed and explained, yielding a finite set of explanations* X*. The number of explanations in* X *is* $|X|$*, the cardinality of* X*. An incoming observation* α *is*

added to δ, yielding the new perception $\delta;\alpha$. The only way the cardinality of the explanation set could increase with an incoming perception α, would be if $(\delta,\pi) \notin \Xi$ and $(\delta;\alpha,\pi) \in \Xi$, for some plan π and some set of partial traces and plan tuples Ξ. We now consider Ξ_{c}, Ξ_{l}, and Ξ_{p}, specifically. By Lemma 1, for any defining relation R, if $(\delta;\alpha,\delta') \in R$, then $(\delta,\delta') \in R$. If $(\delta;\alpha,\pi) \in \Xi_{\mathsf{c}}$, then $\delta;\alpha \preccurlyeq \delta'$ for some $\delta' \in \mathsf{traces_c}(\pi)$. If $(\delta;\alpha,\pi) \in \Xi_{\mathsf{l}}$, then $\delta;\alpha \sqsubseteq \delta'$, for some $\delta' \in \mathsf{traces_l}(\pi)$. If $(\delta;\alpha,\pi) \in \Xi_{\mathsf{p}}$, then $\delta;\alpha \lesssim \delta'$ for some $\delta' \in \mathsf{traces_p}(\pi)$. In all cases, if $(\delta;\alpha,\pi) \in \Xi$ then $(\delta,\pi) \in \Xi$. Therefore $|X|$ cannot increase with incoming perceptions, and $|X|$ must decrease monotonically. □

One desirable property which we state but do not prove formally is the fact that a new perception can always be explained. This follows from the assumptions in Section 3.1; any action an agent executes is part of some plan belonging to a known rule. If $\Delta = (\delta,\alpha)$ and $\delta;\alpha$ cannot be explained (the explanatory function gives an empty set), perhaps because the agent started working on another plan, then it is always possible to explain $\Delta = (\epsilon,\alpha)$ using $\mathsf{expl_p}$, since α *must* be part of a known rule, and is in any case the dilution of some plan trace.

5 An example

In this section, we present a brief example of the technique of mental state abduction, situated in a role-playing game (RPG) setting. In RPGs, a player is often accompanied by one or more non-player characters (NPCs), which act autonomously and (most of the time) are not under the player's direct control. In a utopic future where all characters are implemented as BDI agents, the player and his or her NPC companions, on a quest to deliver an important letter, are under attack by a band of brigands.[3] One of the brigands somehow knows the whereabouts of this letter, and has a plan for having it in his possession.

$$\overbrace{\mathsf{have(letter)}}^{\kappa} \leftarrow \overbrace{\mathsf{in(chest,\,letter)} \wedge \neg\mathsf{guarded(chest)}}^{\beta} \mid \left.\begin{array}{l} {}^{\alpha_1}\mathsf{goto(chest)};\ {}^{\alpha_2}\mathsf{inspect(chest)}; \\ ({}^{\phi_1}\mathsf{locked(chest)?}; \\ \quad ({}^{\phi_2}\mathsf{sturdy(lock)?};\ {}^{\alpha_3}\mathsf{sheath(sword)};\ {}^{\alpha_4}\mathsf{pick(lock)})+ \\ \quad ({}^{\neg\phi_2}\neg\mathsf{sturdy(lock)?};\ {}^{\alpha_5}\mathsf{smash(lock)}))+ \\ ({}^{\neg\phi_1}\neg\mathsf{locked(chest)?})); \\ {}^{\alpha_6}\mathsf{open(chest)};\ {}^{\alpha_7}\mathsf{take(letter)} \end{array}\right\}\pi$$

This specific plan for having the letter has as a condition that the letter is believed to be in the chest, and the chest to be unguarded. The plan consists of inspecting the chest, picking or smashing the lock (depending on its quality) in case the chest is locked, then opening it and taking the letter. Notice that actions and tests have been annotated with α and ϕ, respectively.

[3] The setting of a fight might seem rather violent, but it serves nicely to illustrate various points which deserve attention.

The rule can also be represented as follows, using the annotations instead of the actual formulas.

$$\kappa \;\leftarrow\; \beta \mid \alpha_1;\alpha_2;(\phi_1?;(\phi_2?;\alpha_3;\alpha_4)+(\neg\phi_2?;\alpha_5))+(\neg\phi_1?);\alpha_6;\alpha_7$$

In the chaos of the fight, nobody is guarding the chest in which the letter has been stored. The brigand with the plan for stealing the letter takes notice of this. Fortunately, one of the NPCs sees the brigand inspecting the chest, and also has knowledge of the brigand's plan.

Assume the NPC is abducing the brigand's mental state using our technique of mental state abduction. The set of possible plan traces $\tau(\pi)$ then looks as follows, with overbraced conditions indicating a trace.

$$\tau(\pi) = \{\overbrace{\alpha_1;\alpha_2;\alpha_3;\alpha_4;\alpha_6;\alpha_7}^{\phi_1\wedge\phi_2},\ \overbrace{\alpha_1;\alpha_2;\alpha_5;\alpha_6;\alpha_7}^{\phi_1\wedge\neg\phi_2},\ \overbrace{\alpha_1;\alpha_2;\alpha_6;\alpha_7}^{\neg\phi_1}\}$$

The NPC observing the brigand perceived that the brigand was inspecting the chest. Assuming it didn't perceive anything prior to that, $\Delta = (\epsilon, \alpha_2)$. Because of the tumultuous nature of the fight, the NPC assumes it might miss perceptions and therefore uses $\mathsf{expl_p}$. In this case, if $\mathsf{expl_p}(\alpha_2, \mathcal{RD}, \Lambda, \Xi_\mathsf{p}) = X$, then $\{\mathsf{goal}(\mathsf{have}(\mathsf{letter})) \wedge \mathsf{belief}(\beta)\} \subseteq X$. Note that if $\mathsf{expl_c}$ had been used, this mental state could not have been abduced, as α_2 is not a prefix of a trace of π. Of course, other plans might involve opening the chest, and their mental state preconditions will also be part of the set of explanations.

The NPC then diverts its attention because of a brigand charge, and the next time it looks it sees the agent picking the lock, and since $\mathsf{pick}(\mathsf{lock}) = \alpha_4$, the new perception is $\Delta = (\alpha_2, \alpha_4)$. Since $(\alpha_2;\alpha_4,\pi) \notin \Xi_\mathsf{c}$ and $(\alpha_2;\alpha_4,\pi) \notin \Xi_\mathsf{l}$, but $(\alpha_2;\alpha_4,\pi) \in \Xi_\mathsf{p}$, only $\mathsf{expl_p}$ can explain $\alpha_2;\alpha_4$ based on π.

Depending on its rules, the NPC can respond to the brigand trying to take the letter. It might inform others, or actively intervene itself. Note that the plan as represented here is only for illustrative purposes, an actual plan would be somewhat less specific. Also, in the observation there is no indicator of which agent performed an action. This is due to our presentation of an observer as separated from the agent; in an actual implementation this indicator would be required.

6 Conclusion and future work

In this paper we presented a technique called mental state abduction, which allows for inferring the mental states of BDI-agents based on their observed behavior, using an approach inspired by logical abduction. Three explanatory strategies were presented, based on perceptory conditions reflecting either properties under the observer's influence, or properties of the environment beyond the observer's control. Our technique was demonstrated to work for agents programmed in a simplified BDI-based programming language.

In future research we hope to extend mental state abduction to deal with the dynamics of a current agent programming language such as 2APL. Specifically, we intend to make use of the internal actions inside plans, such as conditional (belief) tests, as a guideline for better abduction. Furthermore, making use of specifics of the underlying deliberation process of agents can provide information that can be useful for narrowing the space of plausible explanations. Another possibility we see is to incorporate institutional notions such as roles to provide restrictions on specific clusters of rules an agent might be believed to have. In line with this, we intend to relax the requirement of the observing party having to know all of the agent's rules. Last but not least, we will explore the possibilities of inter-agent cognitive modelling using our technique.

References

1. Norling, E., Sonenberg, L.: Creating interactive characters with BDI agents. Proc. of the Australian Workshop on Interactive Entertainment (2004) pp. 69–76
2. Bordini, R., Dastani, M., Dix, J., El Fallah Seghrouchni, A., eds.: Multi-Agent Programming: Languages, Platforms and Applications. Springer (2005)
3. Doyle, P.: Believability through context. In: Proceedings of AAMAS '02, ACM (2002) pp. 342–349
4. Dastani, M., Meyer, J.-J.Ch.: A practical agent programming language. Proceedings of ProMAS 2007 (2008)
5. Flach, P.: Conjectures: An Inquiry Concerning the Logic of Induction. PhD thesis, University of Tilburg (1995)
6. Hartshorne, C., Weiss, P., eds.: Collected Papers of C.S. Peirce. Harvard University Press (1931–58)
7. Aliseda, A.: A unified framework for abductive and inductive reasoning in philosophy and AI. In: Proceedings of the ECAI'96 Workshop on Abductive and Inductive Reasoning. (1996)
8. Kakas, A., Kowalski, R., Toni, F.: The Role of Abduction in Logic Programming. In: Handbook of Logic in Artificial Intelligence and Logic Programming 5. Oxford University Press (1998) pp. 235–324
9. Aliseda-Llera, A.: Seeking Explanations: Abduction in Logic, Philosophy of Science, and Artificial Intelligence. PhD thesis, University of Amsterdam (1997)
10. Harman, G.: Inference to the best explanation. The Philosophical Review **74** (1965) pp. 88–95
11. Lipton, P.: Inference to the Best Explanation. 2nd edn. International Library of Philosophy. Routledge (2004)
12. Carberry, S.: Techniques for plan recognition. User Modeling and User-Adapted Interaction **11**(1-2) (2001) pp. 31–48
13. Albrecht, D.W., Zukerman, I., Nicholson, A.E.: Bayesian models for keyhole plan recognition in an adventure game. User Modeling and User-Adapted Interaction **8**(1-2) (1998) pp. 5–47
14. Appelt, D.E., Pollack, M.E.: Weighted abduction for plan ascription. User Modeling and User-Adapted Interaction **2**(1-2) (1991) pp. 1–25
15. Goultiaeva, A., Lespérance, Y.: Incremental plan recognition in an agent programming framework. Working Notes of the AAAI Workshop on Plan, Activity, and Intention Recognition (PAIR) (2007)