Consistency Conditions for Beliefs and Intentions

Marc van Zee, Mehdi Dastani, Dragan Doder, and Leendert van der Torre

Abstract

Icard et al. introduce a semantics for actions over time, provide an axiomatization for this logic, and use this logic to define coherence conditions for a belief-intention database. First, we show incompleteness of their axiomatization and we adapt their semantics and provide a complete axiomatization for it. Second, we show that Icard et al.'s definition of coherence is too weak, and we define a stronger notion of coherence using our new logic.

Introduction

Shoham (2009) identifies the psychological, social and artifactual perspectives from which theories of intention are considered. When considering how to formalize intention – or any other complex natural notion – these perspectives provide "the yardsticks by which one would evaluate the theory" (Shoham 2009, p.3). Each perspective "drives the formal theory of intention and its interaction with belief" (Shoham 2009, p.2), and leads to an approach to formalizing mental state. Shoham observes that the artifactual perspective can be instantiated in many ways. He explores a particular class of instantiations, which he deems useful in the context of intention, and he calls this the database perspective. Shoham then introduces the belief-intention database, capturing specific interactions between beliefs and intentions. The database is used by a planner that is engaged in some form of practical reasoning and stores its intentions and beliefs respectively in an intention database and a belief database. Shoham suggests that, besides the standard functionality of storage and retrieval, the belief and intention databases should satisfy the following three consistency conditions: First, the belief database is internally consistent. Secondly, the intention database is internally consistent. Thirdly, the belief database and the intention database are mutually consistent.

Icard *et al.* (2010) formalize the belief-intention database by introducing a semantics for actions over time and an axiomatization for this logic. They claim that this logic is sound and strongly complete but omit proofs. A belief database is a set of formulas from this logic and an intention database is a set of actions over time. Coherence between the belief and intention database is expressed syntactically by a formula of their logic, as well as semantically. They use these coherence conditions to define AGM-like revision postulates, for which they provide a representation theorem.

As we will show in the next section, the logic of Icard *et al.* is incomplete. This leads to the following three research questions:

1. Which axioms must be added to the logic of Icard *et al.* to obtain completeness?
2. How to (minimally) adapt the logic of action and time to deal with the incompleteness?
3. How to formalize the consistency conditions between beliefs and intentions?

We show that in order to obtain completeness, it is necessary to change the syntax of the logic of Icard *et al.* We correct their logic and obtain a branching time logic containing a modal operator $\Box_t$ such that $\Box_t\varphi$ means "the agent believes that it is necessary that φ at time moment t". We formalize the intention database using discrete atomic action intentions of the form (a,t), which stands for "the agent intends to do a at time t". We then show that the coherence condition proposed by Icard *et al.* is too weak and we propose a stronger version.

Our methodology is similar to Icard *et al.*'s, namely to develop a logic that is suitable to be used for AGM style revision of beliefs and intentions. Therefore, the logic should be as simple and as close to propositional logic as possible. The success criteria of this approach is future work, where we study the revision of both beliefs and intentions such that the consistency conditions remain satisfied.

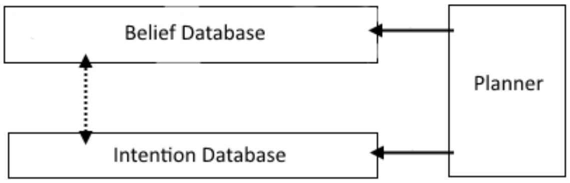

Figure 1: The Belief-Intention Database

Figure 1 gives a visual overview of the belief-intention database. The solid arrows depict the addition or removal of beliefs and intentions. The dashed arrows depict consistency conditions between elements. In this paper, we study how to formalize beliefs, the intentions, and the dashed arrows. We leave the solid arrows to future work.

The layout of this paper follows the research questions and is as follows: In the first section we introduce the belief-intention database and the axiomatization issues with Icard *et al.* In the second section we introduce our logic and we show that is sound and strongly complete. In the third section we show that Icard *et al.*'s definition of coherence is too weak and we propose a stronger version.

The Belief-Intention Database

Shoham proposes the Belief-Intention Database, which consists of a planner, in particular of the sort encountered in so-called "classical" AI planning (Weld 1999) that is engaged in some form of practical reasoning. The planner posits a set of actions to be taken at various times in the future in an intention database, and updates this database as it continues with its deliberations. Shoham considers so-called *discrete atomic action intentions* of the form "I intend to take action a at time t", where a belongs to a fixed set of atomic actions and t is an integer. Such atomic action intentions can be extended in various ways (see Shoham (2009) for a discussion), but Shoham focuses on atomic action intentions since he believes they are the basic building block for the more complex constructs and these basic actions already contain nontrivial complications.

Example 1. *We now introduce the running example that we will use throughout the rest of the paper. Suppose that a household agent is planning tasks for the day. The first intention that it forms is to fetch the newspaper for its owner at time 0, after which it will bring the newspaper to the owner at time 1. Finally, the agent plans to make a breakfast containing ham and eggs for its owner at time 2, this breakfast is called "breakfast 1", or simply "bf1". Formally, the agent's intention base I contains* $\{(fetch,0),(bring,1),(bf1,2)\}$.

Planners usually associate pre-and postconditions with atomic actions, which respectively denote the execution conditions and the effects of actions. Therefore, Shoham argues that the database must represent both beliefs and intentions. Shoham proposes the following consistency conditions that must be satisfied by the database:

(C1) Beliefs must be internally consistent.
(C2) At most one action can be intended for any given time moment.
(C3) If you intend to take an action, you believe that its postconditions hold.
(C4) If you intend to take an action you cannot believe that its preconditions do not hold.

Note that (C3) and (C4) contain a certain asymmetry. Shoham motivates this by stating that "(...) only at the conclusion of planning– and sometimes not even then– are all these preconditions established. (...) it is a good fit with how planners operate. Adopting an optimistic stance, they feel free to add intended actions so long as those are consistent with current beliefs, but once they do they continue acting based on the assumption that these actions will be taken, with all that follows from it." (Shoham 2009, p.7-8).

Icard *et al.* (2010) formalize Shoham's belief-intention database using a paths semantics for actions over time to model the belief database, and they define a notion of "appropriate sets of paths". This intuitively means that if an agent believes the precondition of some action at time t, then the agent considers it possible that it carries out this action. They characterize the appropriateness condition with the following axiom:

$$pre(a)_t \rightarrow \Diamond do(a)_t \tag{1}$$

The following proposition shows that this axiom does not capture the semantics correctly.

Proposition 1. *Equation* (1) *is not sound for the semantics of Icard* et al. *(2010).*

This already means that the axiomatization of Icard *et al.* is not correct. We prove even more.

Proposition 2. *The logic of Icard* et al. *(2010) is not compact.*

The next corollary follows directly.

Corollary 1. *There is no finitary*[1] *sound and complete axiomatization for the syntax and semantics of Icard* et al. *(2010).*

This means that it is not possible to correct the axiomatization of Icard *et al.*, while maintaining the syntax. This is not surprising if consider that all propositions of their logics are parameterized by a time point, while the modality is not. This prevents one to express semantically the property that precondition for an action a in t should enable execution of the action a in t.

This motivates us to change the syntax of Icard *et al.* by replacing their modality $\Box$ with a parameterized version $\Box_t$. In the next section, we present the logic with this modified syntax and we show that our logic is sound and strongly complete with respect to a tree semantics.

The Logic

Definition 1 (Language)**.** *Let* $A = \{a,b,c,\ldots\}$ *be a finite set of deterministic primitive actions and* $P = \{p,q,r,\ldots\} \cup \bigcup_{a\in A}\{pre(a), post(a), do(a)\}$ *be a finite set of propositions, such that* P *and* A *are disjoint. The language* $\mathscr{L}$ *is inductively defined by the following BNF grammar*

$$\varphi ::= \chi_t \mid \Box_t\varphi \mid \varphi\wedge\varphi \mid \neg\varphi$$

with $\chi \in P$ *and* $t \in \mathbb{Z}$. *We abbreviate* $\neg\Box\neg$ *with* $\Diamond$, *and define* $\bot \equiv p \wedge \neg p$ *and* $\top \equiv \neg\bot$.

Example 2. *The following formulas are syntactically correct in our language:* $\neg sunday_0 \rightarrow do(fetch)_0$ *(if it is not sunday at time 0, then fetch the newspaper),* $do(bring)_1$ *(bring the newspaper at time 1),* $\Box_0(do(bf_1)_3 \vee do(bf_2)_3)$ *(it is necessary at time 0 that either breakfast 1 or breakfast 2 is prepared at time 3),* $\Diamond_0 pre(fetch)_0$ *(it is possible that the precondition to fetch the newspaper is true at time 0).*

[1] Without infinitely long formulas and infinitary rules of inference

The semantics are similar to CTL*, namely a branching time structure, where branches are sequences of states. This general structure can be constrained in various ways, leading to different kinds of semantics such as the trees, bundles, or complete bundles (see (Reynolds 2002) for an overview of different kinds of semantics). We choose the first semantics and use a tree frame, which is a single tree containing nodes and edges connecting the nodes.

Definition 2 (Tree frame)**.** *A tree frame is a pair* (S,R) *where* $S = \bigcup_{n\in\mathbb{Z}} S_n$ *is the union of disjoint sets of states (i.e.* $S_i \cap S_j = \emptyset$ *for any* $i, j \in \mathbb{Z}, i \neq j$*), each* S_i *containing a set of states at time i, and R is an accessibility relation with the following properties:*

1. $R \subseteq \bigcup_{n\in\mathbb{Z}} S_n \times S_{n+1}$
2. *R is serial:* $(\forall s \in S).((\exists s' \in S).(sRs'))$
3. *For each* $s \in S$*, the past is linearly ordered by R:* $(\forall s \in S).((\exists^1 s' \in S).(s'Rs))$
4. *R is connected:* $(\forall s, s' \in S).((\exists s'' \in S).(s''R^*s \wedge s''R^*s'))$*, where* R^* *is the transitive closure of R.*

The intuition of tree frames is as follows: With each integer $i \in \mathbb{Z}$ we associate a set of states S_i such that all these sets are disjoint. Property 1 of the accessibility relation R states that it only relates two states at subsequent time points. Property 2 and 3 together ensure that R is generating an infinite tree. Finally, property 4 ensures that this tree is unique.

Next, we add valuations to both the nodes and the edges of this tree. The valuation of the nodes represent the facts that hold in that state, and the edges contain actions representing transitions from one state to another.

Definition 3 (Tree structure)**.** *A tree structure is a tuple* $T = (S,R,v,a)$ *where* (S,R) *is a tree frame,* $v : S \rightarrow 2^{\mathcal{L}}$ *is a valuation function from states to sets of propositions, and* $a : R \rightarrow \mathsf{A}$ *is another valuation function from accessibility relations to actions.*

Example 3. *Consider the tree in Figure 2. The tree structure* $T = (S,R,v,a)$ *consists of states* $S = S_0 \cup \ldots \cup S_3$ *with* $S_0 = \{s_0\}, S_1 = \{s_1, s_2\}, \ldots, S_3 = \{s_5, s_6\}$*. The relations are defined as* $R = \{(s_0,s_1),(s_0,s_2),\ldots,(s_3,s_6)\}$ *and the valuation functions v and a are defined such that for instance* $v(s_0) = \{pre(fetch)\}, v(s_3) = \{egg, post(bring)\}$ *and* $a((s_0,s_1)) = fetch, a((s_1,s_3)) = bring$*, and* $a((s_1,s_4)) = wait$*.*

Next, we define a notion of a *path*, which is a trace through the tree on which formulas are evaluated.

Definition 4 (Path)**.** *Given a tree structure* $T = (S,R,v,a)$*, a* path $\pi = (s_0, s_1, \ldots)$ *in T is a sequence of states such that* $(s_t, s_{t+1}) \in R$*. We write* π_t *to refer to the t'th state of the path* π *and we thus write* $v(\pi_t)$ *and* $a(\pi_t)$ *to refer respectively to the propositions true and the next action on path* π *at time t.*

Definition 5 (Path equivalence)**.** *Two paths* π *and* π' *are equivalent up to time t, denoted* $\pi \sim_t \pi'$*, if and only if they contain the same states up to and including time t and the same actions up to time t, i.e.* $\pi \sim_t \pi'$ *iff.* $(\forall t' \leq t).(\pi_{t'} = \pi'_{t'}) \wedge (\forall t' < t).(a(\pi,t') = a(\pi',t'))$*.*

Note that path equivalence is defined on states, so two paths may describe exactly the same situation up to and including t (i.e. the same valuation), but they still may not be equivalent.

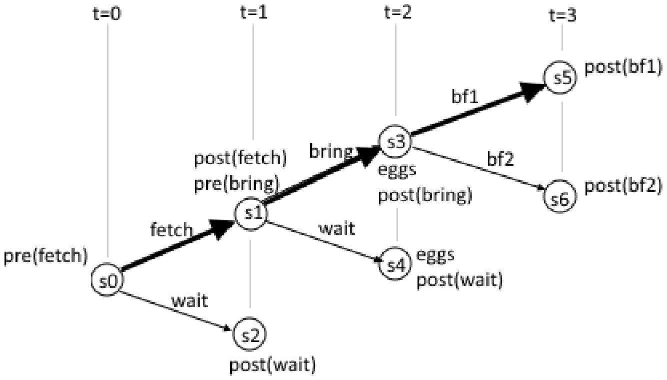

Figure 2: Example Model

Example 4. *The tree in Figure 2 contains four paths, namely* $\pi_1 = (s_0, s_1, \ldots, s_5), \pi_2 = (s_0, s_1, \ldots, s_6), \pi_3 = (s_0, s_1, s_4)$*, and* $\pi_4 = (s_0, s_2)$*. Paths are equivalent up to the moment where they describe the same situation, so we have for instance* $\pi_1 \sim_t \pi_2$ *for* $t \leq 2$*,* $\pi_2 \sim_t \pi_3$ *for* $t \leq 1$*, and* $\pi_3 \sim_t \pi_4$ *for* $t \leq 0$*.*

As the language in Definition 1 already made clear, the logic contains propositions that are explicitly indexed with the state at which they hold, where states are identified with integers. For instance, the formula p_t denotes that the proposition p is true at time t. Unlike most other branching time logics, formulas are not evaluated in a state on a path, but simply on a path as a whole. So, a model for a formula is pair of a tree and a path in this tree on which the formula is true. This, together with some additional constraints related to the pre-and postconditions of actions, is our definition of a *(pointed) model*.

Definition 6 (Model)**.** *A* pointed model*, or simply a* model *is a pair* (T,π) *where* $T = (S,R,v,a)$ *is a tree structure satisfying the following conditions:*

1. *If* $a(\pi_t) = a$*, then* $post(a) \in v(\pi_{t+1})$*.*
2. *If* $pre(a) \in v(\pi_t)$*, then there is some* π' *in T with* $\pi' \sim_t \pi$ *and* $a(\pi'_t) = a$*.*

and π *is a path in T. We denote models with* $m_1, m_2, \ldots$ *and sets of models with* $M_1, M_2, \ldots$ *We denote the set of all models with* $\mathcal{M}$*.*

The first condition denotes that if an action is selected as the next action on the path, then in the next state of this path the post-condition of this action should be true. The second action denotes that if the pre-condition of an action is true, then the agent should consider it possible that this action is in fact executed (i.e. there exist a path equivalent with the current path at which the action is carried out).

Example 5. *The tree structure of Figure 2 satisfies all of the conditions of Definition 6, so this is a model for each path in the tree. For instance* (M,π_1) *is a model (where* π_1 *as defined in the previous example), because we have (among others)* $pre(fetch) \in v(\pi_0)$*, so there should be some* $\pi' \sim_0 \pi_1$ *such that* $a(\pi')_0 = fetch$*. This is the case, since* $a(\pi_1) = fetch$ *and clearly* $\pi_1 \sim_t \pi_1$ *for any t.*

Definition 1 indicates that the language uses the time-indexed modality $\Box_t$. The reason that this modality is indexed with a time point is because formulas are not evaluated in a state, but on a branch. Therefore, one should explicitly specify up to what point branches are equivalent.

Definition 7 (Truth definitions)**.** *Let $m = (T,\pi)$ be a model with $T = (S,R,v,a)$:*
$T,\pi \models p_t$ *iff* $p \in v(\pi_t)$ *for any* $p \in \mathscr{L}$
$T,\pi \models do(a)_t$ *iff* $a(\pi_t) = a$
$T,\pi \models \neg\varphi$ *iff* $T,\pi \not\models \varphi$
$T,\pi \models \varphi \wedge \beta$ *iff* $T,\pi \models \varphi$ *and* $T,\pi \models \beta$
$T,\pi \models \Box_t\varphi$ *iff* $(\forall\pi').(\pi \sim_t \pi' \rightarrow T,\pi' \models \varphi)$

Note that throughout the paper we use $T,\pi \models \varphi$ and $m \models \varphi$ interchangeably. We have used the first notation in the above definition to emphasize the fact that formulas are evaluated in tree structures on paths, but we will mostly use the second notation in what follows since it is more concise. Validity, satisfiability and semantic consequence are defined as usual.

Example 6. *In Figure 2 (using paths $\pi_1,\dots,\pi_4$ from the previous example), the following are true: $T,\pi_1 \models pre(fetch)_0 \wedge do(fetch)_0$, i.e. at time 0 the precondition of $fetch$ is true and the agent also carries out this action. $T,\pi_2 \models \Box_1 egg_2$ holds as well, since at time 1 on path π_2, it holds that on all paths equivalent with π_2 the formula egg is true at time 2 (i.e. it is true in state s_3 and state s_4). Finally, $T,\pi_1 \models \Diamond_0 do(wait)_1 \wedge \Diamond_0 do(bf_1)_2$ holds as well, but $T,\pi_1 \models \Diamond_0(do(wait)_1 \wedge do(bf_1)_2)$ does not hold.*

Definition 8 (The logic PBTL)**.** *The logic PBTL consists of the following axiom schemas and rules:*

1. *Propositional tautologies*
2. *The K-axiom:* $\Box_t(\varphi \rightarrow \psi) \rightarrow (\Box_t\varphi \rightarrow \Box_t\psi)$
3. $\Box_t\varphi \rightarrow \Box_t\Box_t\varphi$
4. $\Box_t\varphi \rightarrow \varphi$
5. $\Diamond_t\varphi \rightarrow \Box_t\Diamond_t\varphi$
6. $\Box_t\varphi \rightarrow \Box_{t+1}\varphi$
7. $\chi \rightarrow \Box_t\chi$*, where* $\chi \in \{p_t, pre(a)_t, post(a)_t\}$
8. $\Diamond_t\chi \rightarrow \chi$*, where* $\chi \in \{p_t, pre(a)_t, post(a)_t\}$
9. $do(a)_t \rightarrow \Box_{t+1}do(a)_t$
10. $\Diamond_{t+1}do(a)_t \rightarrow do(a)_t$
11. $\bigvee_{a\in A} do(a)_t$
12. $do(a)_t \rightarrow \bigwedge_{b\neq a}\neg do(b)_t$
13. $do(a)_t \rightarrow post(a)_{t+1}$
14. $pre(a)_t \rightarrow \Diamond_t do(a)_t$
15. $\Diamond_t(do(a)_t \wedge \varphi) \rightarrow \Box_t(do(a)_t \rightarrow \varphi)$
16. *Necessitation: from* φ*, infer* $\Box_t\varphi$*, where* $t \in \mathbb{Z}$
17. *Modus Ponens*

Axiom 2-5 together define the $\Box$-operator as an S5 equivalence relation. Axiom 6-8 together state that everything that is true at a time will necessarily be true in all future time points. Note that these three axioms together imply $\chi \rightarrow \Box_{t'}\chi$ with $t' \geq t$. Axiom 9 and 10 state that an action that is (not) executed will necessarily be (not) executed on all equivalent branches after executing it. Axiom 11 and 12 together state that for every time point, the agent believes it carries out a single action. Axiom 13 states that the agent believes the postconditions of the actions that it believes it will carry out, and Axiom 14 state that if the preconditions of an action are true, the agent believes that it is possible that he will in fact carry out the action. Axiom 15 enforces deterministic actions: It is not possible to have two relation from the same state with the same action.

Theorem 1. *The logic PBTL is sound and strongly complete with respect to the class of all models.*

Definition 9 (Models of (sets of) formulas)**.** *Given a PBTL-formula φ and a model $m = (T,\pi)$, we say that m is a* model for φ *iff. $m \models \varphi$. We denote the set of all models for φ with $Mod(\varphi)$, i.e. $Mod(\varphi) = \{m \mid m \models \varphi\}$. By abuse of notation, we denote the set of all models for a set of formulas Σ with $Mod(\Sigma)$ as well, i.e. $Mod(\Sigma) = \bigcap_{\varphi\in\Sigma} Mod(\varphi)$.*

Now that we have defined all the elements of our framework, we gather them together in our definition of an *agent*. Note that the agent does not contain the planner, since this is not part of our representation.

Definition 10 (Agent)**.** *The consequence of a set of PBTL-formulas Σ is defined as $Cn(\Sigma) = \{\varphi \mid \Sigma \vdash \varphi\}$. An* agent *$A = (B,I)$ consists of:*

1. *A* belief database *B: A set of PBTL-formulas closed under consequence, i.e. $B = Cn(B)$;*
2. *An* intention database *I: A set of intentions of the form (a,t) with $a \in A$ and $t \in \mathbb{Z}$ such that no two intentions occur at the same time point, i.e. if $\{(a,t),(a',t')\} \subseteq I$, then $t \neq t'$.*

Note that our definition of a belief database is not in line with the belief revision literature. In their terminology, a belief base is a set of ground facts not necessarily closed under consequence, while a belief set is always closed under consequence. We have chosen to use belief databases because this is in line with the *database* perspective.

Definition 11 (Agent Model)**.** *Given an agent $A = (B,I)$, the* agent model *of A is $(Mod(B),I)$.*

The Consistency Conditions

In this section, we formalize consistency conditions (C1)-(C4) that were introduced in the second section. The first two conditions follow directly from our framework.

(C1) *Beliefs must be internally consistent. B* is consistent.

(C2) *At most one action can be intended for any given time moment.* This is ensured by the definition of an intention database in Definition 10. Note that in our framework the agent can also not believe to intend more than one action at any given time moment due to Axiom 12 of Definition 8, stating that at each time point, the agent only believes that it does a single action. In fact, together with Axiom 11 this is equivalent to the constraint that the agent believes that it carries out exactly one action at each time moment.

For the last two consistency conditions, namely the ones characterizing consistency between beliefs and intention, we require additional constraints on our framework.

(C3) *If you intend to take an action, you believe that its postconditions hold.* As a first try, we may translate this statement into our framework directly:

$$B \vdash \bigwedge_{(a,t)\in I} post(a)_t.$$

However, this solution suffers from the well-known "Little Nell" problem, identified by McDermott (1982) and discussed, amongst others, by Cohen and Levesque (1990): Once a robot forms the intention to save Nell from the tracks of a train, the deliberation mechanism will notice that "Nell is going to be mashed" is no longer true, so the intention is removed because there is no longer a justification for it.

To deal with these subtleties, we make a distinction between intention-contingent beliefs, or simply contingent beliefs, and concrete "psychical" beliefs, or non-contingent beliefs. The intention-contingent beliefs are similar to the contingent beliefs by Icard *et al.* (2010) and the notion of *weak beliefs* by van der Hoek and Wooldridge (2003). Non-contingent beliefs concern the world as it is and what the pre-and postconditions of actions are, independent of the agent's plans about the future. The contingent beliefs, on the other hand, are the beliefs that the agents has, dependent on the success of her actions and the action of the agents in shared collective intentions. Thus, the contingent beliefs are derived from the non-contingent beliefs, simply adding the post-conditions (and all consequences) of any intended actions and collectively intended actions. These kinds of beliefs might also be called "optimistic" beliefs, since the agent assumes the success of the action without ensuring the preconditions hold. We denote the contingent beliefs by B^I:

$$B^I = Cl(B \cup \{post(a)_t \mid (a,t) \in I\})$$

Given our discussion above, we interpret Shoham's consistency condition as: *If you intend to take an action, you believe that its postconditions hold contingently.* This is equivalent to requiring that the contingent beliefs of an agent are consistent, so our formalisation of this consistency condition is: B^I is consistent.

(C4) *If you intend to take an action you cannot believe that its preconditions do not hold.* This condition is formalized semantically by Icard *et al.* (2010) as follows:

$$\pi, 0 \models \Diamond \bigwedge_{(a,t)\in I} pre(a)_t$$

Although their semantics is slightly different from ours, the general idea of this formula is clear: There exists a path, equivalent with the current path up to time 0, in which all the preconditions of the intended actions hold. The following example shows that this definition is too weak.

Example 7 (Coherence)**.** *Assume the robot has the intention base $I = \{(fetch,0),(bring,1),(bf_1,2)\}$ and that it has the model that is depicted in Figure 3, in which the path with the bold arrows is the path π of the model of the agent. In this model m_1, on the other path, all preconditions of the intended action are true, even though none of the intended actions are carried out on this path.*

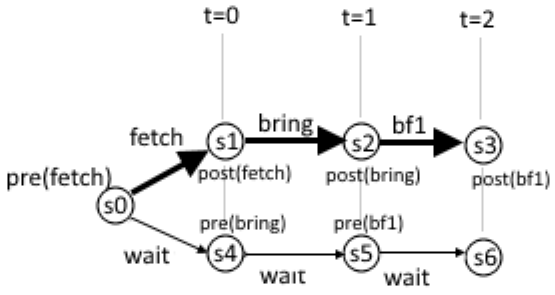

Figure 3: Example Model m_1

This is clearly too weak. An agent may believe that all the preconditions hold on a paths where none of its intended actions are carried out. In order to resolve this, we require that the beliefs of an agent should be consistent with the preconditions of it's intended action. So, the agent does not have to believe the preconditions of its intended actions, but he should not believe the negation of the precondition of an intended action. Therefore, we introduce precondition formulas that are derived from the contingent beliefs:

$$Pre(B^I) = Cl(B^I \cup \{\bigwedge_{(a,t)\in I} pre(a)_t\}\})$$

We can then express the condition as follows: $Pre(B^I)$ is consistent.

If our formal condition of (C4) is satisfied, then the formal condition for (C3) is satisfied as well, so we can obtain the following definition of coherence:

Definition 12 (Coherence)**.** *An agent $A = (B,I)$ is* coherent *iff. $Pre(B^I)$ is consistent.*

An agent model $(Mod(B),I)$ is coherent *iff there exists some $m \in Mod(B) : m \models Pre(B^I)$.*

Example 8 (Coherence, ctd.)**.** *Given the new definition of coherence, it follows that the agent of our previous example is not longer coherent. For instance, it has the intention $(bring,1)$, but it also follows that $m_1 \not\models pre(bring)_1$. Since m_1 is the only model of the agent, the agent is not coherent.*

Discussion and Conclusion

We show that we cannot axiomatize the logic of Icard *et al.*, which motivates us to alter the syntax by replacing modalities with time-parameterized modalities. We prove that this logic is sound and strongly complete wrt. a tree semantics. We show that the coherence condition of Icard *et al.* is too weak and proposed a stronger version.

A large number of logical systems have been developed for reasoning about informational and motivational attitudes in dynamic environments, mostly following the paper of Cohen and Levesque (1990) (see Meyer and Veltman (2007) and van der Hoek and Woolridge (2003) for surveys). Most of these logics focus on the process of intention generation (e.g. practical reasoning) and the question of how to model the persistence of intentions over time (see Herzig and Lorini (2008) for a survey). The approach of Shoham (2009) and Icard *et al.* (2010), namely how an agent should revise its beliefs and intentions together given new information or a change of plans, has received relatively little attention (Van der Hoek and Wooldridge 2003; Lorini et al. 2009; Roy 2009). Broadly speaking, the logical framework we use in this paper falls into the category of the so-called "BDI-logics", in the sense that we model an agent using the mental states of belief and intention (we leave out desires).

Our semantics is close to CTL*, the branching-time models of Rao and Georgeff (1991). However, one important difference is that we focus on the intention to perform an action at a specific moment in time. The benefits of this are discussed by Shoham (2009). In this paper, plans are not explicitly part of the framework, but, as a feature of the database perspective, are conceived of in the background as a recipe describing precisely what actions the agent will perform at specific moments in time. In the framework for van der Hoek *et al.* (2003), a plan describes what needs to be true in order to fulfill some desire, and consequently they focus on the problem of revising intentions and beliefs in the presence of new information and less on the effect on beliefs of adopting new intentions.

We see three main directions for future work. Firstly, we would like to investigate the coherence condition in more detail, containing more involved examples. Although our current version is stronger than the one of Icard *et al.*, it does no longer make use of a modal operator. This seems quite strong, and we would like to investigate whether other version as possible. Secondly, now that we have provided a correct axiomatization for our logic, we would like to study the revision of beliefs and intentions. Icard *et al.* provide a representation theorem in their paper as well, but again omit proofs. We would like to see whether this theorem holds in our setting. Finally, we deem it interesting to add goals to our framework as well. BDI logics usually consider beliefs, desires or goals, and intentions. It thus seems a logical next step to introduce goals as well.

Acknowledgments

We thank Thomas Icard, Eric Pacuit, and Yoav Shoham for useful comments and insightful discussion. Furthermore, we thank the anonymous reviewers for their useful comments that helped to improve the paper significantly. This work was partially funded by the National Research Fund, Luxembourg.

Appendix: Proofs

Proposition 1. *Equation* (1) *is not sound for semantics of Icard* et al. *(2010).*

Proof. Due to space constraints we refer to Icard *et al.* (Icard, Pacuit, and Shoham 2010) for the notation we use in these proofs. Let $A = \{a, b\}$, and let π and π' we two paths satisfying the following conditions:

- $pre(a) \in \pi(1)_1$ and $pre(a) \notin \pi(t)_1$ for $t \neq 1$
- $pre(b) \notin \pi(t)_1$ for every t
- $\pi(1)_2 = b$
- $\pi'(1)_2 = a$
- $\pi \sim_t \pi'$

It is easy to check that $\Pi = \{\pi, \pi'\}$ is an appropriate set of paths. By (Icard, Pacuit, and Shoham 2010, Def.2), $\pi, 2 \models_\Pi pre(a)_1$. If Eq. (1) is sound, then $\pi, 2 \models \Diamond do(a)_1$. So there is some $\pi'' \in \Pi$ such that $\pi'' \sim_2 \pi$ and $\pi'', 2 \models_\Pi do(a)_1$. However, this is not possible, since $\Pi = \{\pi, \pi'\}$. If $\pi'' = \pi$, then $\pi'', 2 \not\models_\Pi do(a)_1$, and if $\pi'' = \pi'$, then $\pi'' \not\sim_2 \pi$. Thus, we conclude that Eq. (1) is not sound. □

Proposition 2. *The logic of Icard* et al. *(2010) is not compact.*

Proof. Let p be a propositional letter and let $T = \{p_t \mid t \in \mathbb{Z}\} \cup \{\Diamond \neg p_t \mid t \in \mathbb{Z}\}$. Every finite subset T' of T is satisfiable. Indeed, suppose that t' is the smallest time index that appears in any formula of T', and let t_0 be any integer such that $t_0 < t'$. Let $\Pi = \{\pi_1, \pi_2\}$, where π_1 and π_2 are arbitrary paths such that $p \in \pi_1(t)_1$ for every $t \in \mathbb{Z}$, and $p \in \pi_2(t)_1$ iff $t \leq t_0$. Then $\Pi, \pi_1, t_0 \models T'$.

On the other hand, T is not satisfiable. If we suppose that there are Π, π, t' such that $\Pi, \pi, t' \models T$, then obviously $p \in \pi(t)_1$ for every $t \in \mathbb{Z}$. Let t_0 be any integer such that $t_0 < t'$. Since $\Pi, \pi, t' \models \Diamond \neg p_{t'}$, there is a path $\pi' \in \Pi$ such that $\pi' \sim_{t_0} \pi$ and $\Pi, \pi', t' \models \neg p_{t'}$. This is impossible, since $\pi' \sim_{t_0} \pi$ implies that $p \in \pi'(t')_1$, while $\Pi, \pi', t' \models \neg p_{t'}$ implies that $p \notin \pi'(t')_1$. Consequently, the Compactness theorem doesn't hold for the logic. Hence, it follows that the logic is non-compact. □

Theorem 1. *The logic PBTL is sound and strongly complete with respect to the class of all pointed models.*

Proof. The proof is given in a separate technical report (Doder and van Zee 2014). □

References

Cohen, P., and Levesque, H. 1990. Intention is choice with commitment. *Artificial intelligence*.

Doder, D., and van Zee, M. 2014. Parameterized branching time logic (technical report). `http://www.academia.edu/8397706/Parameterized_Branching_Time_Logic_Technical_Report_`. [Online; accessed 18-August-2014].

Icard, T.; Pacuit, E.; and Shoham, Y. 2010. Joint revision of belief and intention. *Proc. of the 12th International Conference on Knowledge Representation* 572–574.

Lorini, E., and Herzig, A. 2008. A logic of intention and attempt. *Synthese* 163(1):45–77.

Lorini, E.; Dastani, M.; van Ditmarsch, H. P.; Herzig, A.; and Meyer, J.-J. C. 2009. Intentions and assignments. In He, X.; Horty, J. F.; and Pacuit, E., eds., *LORI*, volume 5834 of *Lecture Notes in Computer Science*, 198–211. Springer.

McDermott, D. 1982. A temporal logic for reasoning about processes and plans*. *Cognitive science* 6(2):101–155.

Meyer, J., and Veltman, F. 2007. Intelligent agents and common sense reasoning. *Studies in Logic and Practical Reasoning*.

Rao, A., and Georgeff, M. 1991. Modeling rational agents within a BDI-architecture. *KR*.

Reynolds, M. 2002. An axiomatization of full computation tree logic. *Journal of Symbolic Logic* 66(3):1011–1057.

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

Shoham, Y. 2009. Logical theories of intention and the database perspective. *Journal of Philosophical Logic*.

Van der Hoek, W., and Wooldridge, M. 2003. Towards a logic of rational agency. *Logic Journal of IGPL* 11(2):135–159.

Weld, D. S. 1999. Recent advances in ai planning. *AI Magazine* 20(2):93–123.