



HAL
open science

Deciding Refinement Relation in Belief-Intention Databases

Zhanhao Xiao, Andreas Herzig, Laurent Perrussel, Dongmo Zhang

► **To cite this version:**

Zhanhao Xiao, Andreas Herzig, Laurent Perrussel, Dongmo Zhang. Deciding Refinement Relation in Belief-Intention Databases. International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Nov 2017, Bari, Italy. pp.186-199. <hal-02603831>

HAL Id: hal-02603831

<https://hal.science/hal-02603831v1>

Submitted on 16 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <https://oatao.univ-toulouse.fr/22221>

To cite this version:

Xiao, Zhanhao and Herzig, Andreas and Perrussel, Laurent and Zhang, Dongmo *Deciding Refinement Relation in Belief-Intention Databases*. (2017) In: International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), 14 November 2017 - 17 November 2017 (Bari, Italy).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Deciding Refinement Relation in Belief-Intention Databases

Zhanhao Xiao^{1,2(✉)}, Andreas Herzig^{1,3}, Laurent Perrussel¹,
and Dongmo Zhang²

¹ IRIT, University of Toulouse, Toulouse, France
`zhanhao.xiao@ut-capitole.fr`

² SCEM, Western Sydney University, Penrith, Australia

³ IRIT, CNRS, Toulouse, France

Abstract. Bratman’s Belief-Desire-Intention (BDI) theory is seminal in the literature on BDI agents. His BDI theory is taken into account to extend Shoham’s database perspective on beliefs and intentions. In the extended framework, an intentions is considered as a high-level action, which cannot be executed directly, with a duration. They have to be progressively refined until executable basic actions are obtained. Higher- and lower-level actions are linked by the means-end relation, alias instrumentality relation. In this paper, we investigate the complexity of the decision problems for satisfiability, consequence, refinement and instrumentality in the database. Moreover, we translate these problems into the satisfiability and validity problems in propositional linear temporal logic (PLTL). With such translations, we can utilize the efficient automated theorem provers for PLTL to solve the problem of deciding the refinement relation between an intention and an intention set, as well as the instrumentality relation.

1 Introduction

Bratman’s Belief-Desire-Intention (BDI) theory [4,5] is at the basis of the huge literature on BDI agents. According to his theory, intentions are high-level plans to which the agent is committed and they play a fundamental role in autonomous agents. Typically such high-level plans cannot be executed directly: they have to be *refined* as time goes by, resulting in more and more elaborate plans. A means-end relation, which is called *instrumentality*, should link the higher-level intention, which is refined, and the lower-level intentions which are inserted because of the refinement. At the end of the refinement process, there are only *basic actions*: actions the agent can perform intentionally. For example, my high-level plan to submit a paper to a conference is refined into writing a paper and uploading it to the paper submission management system; further down the line, the second intention is refined into logging into the system entering information about the paper and uploading the PDF file.

While operations of refinement is of fundamental importance and should be central in BDI agents, as more extensively discussed in [11], the literature on BDI

theories only contains very few such a concept [2, 14]. The operation of refinement is notably absent from Cohen&Levesque’s logic [8] which is one of the most influential BDI logics and Shoham’s belief-intention database framework [16]. Compared with Cohen and Levesque’s logic, Shoham’s database framework is a much simpler account that is based on a database of time-indexed basic actions and beliefs. Compared with the heavily implementation-driven BDI agents, the belief-intention database is more logical and more suitable for revising beliefs and intentions. In [13], Herzig *et al.* followed Shoham’s database perspective and extended his belief-intention database framework to formalize such a refinement relation between intentions. In the extended framework, beliefs and intentions are organized in a so-called belief-intention database: a belief is a propositional formula indexed by time points and an intention is considered as a high-level action, which cannot be executed directly, with a duration. To capture the change of the environments or actions of other agents, environment actions, alias events, are introduced in belief-intention databases.

As intention refinement plays an important role in BDI theories, the problem of deciding the refinement relation between an intention set and an intention is pivotal. The contributions of this paper are summarized as follows. First, we investigate the complexity of the decision problems for satisfiability, consequence, refinement and instrumentality in belief-intention databases. We show that the satisfiability and consequence problems in the belief-intention database are both PSPACE-complete and further show that the problems of deciding refinement and instrumentality are also PSPACE-complete by reducing them to the satisfiability and consequence problems.

Second, we translate the satisfiability and consequence problems in belief-intention databases into the satisfiability and validity problems in propositional linear temporal logic (PLTL). Then we can translate the problems of deciding refinement and instrumentality into PLTL. In the last decades, PLTL has obtained a lot of attention from the researchers, both theoretically and practically. Taking advantage of the automated tools of PLTL¹, we can solve the decision problems for satisfiability, consequence, refinement and instrumentality of belief-intention databases by translating into the satisfiability and validity problems in PLTL.

Our paper is organized as follows. Section 2 recalls notions of belief-intention databases. Section 3 shows the complexity results. Section 4 gives the translation to PLTL. Section 5 concludes the paper.

2 Belief-Intention Databases

In this section we recall the main definitions of belief-intention databases initially proposed in [13].

¹ Several theorem provers for PLTL can be found on <http://users.cecs.anu.edu.au/~rpg/PLTLProvers/> (accessed on 2 Sep. 2017).

2.1 Coherent Dynamic Theory

Let $\text{Evt}_0 = \{e, f, \dots\}$ be a set of basic events and $\text{Act}_0 = \{a, b, \dots\}$ a set of basic actions. Basic events and basic actions take one time unit. Basic actions can be directly executed by the planning agent. The set Act_0 is contained in the set of all actions $\text{Act} = \{\alpha, \beta, \dots\}$ which also contains non-basic, high-level actions. The set of propositional variables is $\mathbb{P} = \{p, q, \dots\}$. The language of boolean formulas built on \mathbb{P} is denoted by $\mathcal{L}_{\mathbb{P}}$.

We suppose that the sets \mathbb{P} , Evt_0 , and Act are all finite. The behavior of actions and events is described by dynamic theories.

Definition 1 (Dynamic theory). A dynamic theory is a tuple $\mathcal{T} = \langle \text{pre}, \text{post} \rangle$ with $\text{pre}, \text{post} : \text{Act} \cup \text{Evt}_0 \longrightarrow \mathcal{L}_{\mathbb{P}}$. The effects of basic actions and events are conjunctions of literals, given by $\text{eff}^+, \text{eff}^- : \text{Act}_0 \cup \text{Evt}_0 \longrightarrow 2^{\mathbb{P}}$ where for every $x \in \text{Act}_0 \cup \text{Evt}_0$, $\models \text{post}(x) \leftrightarrow \left(\bigwedge_{p \in \text{eff}^+(x)} p \right) \wedge \left(\bigwedge_{p \in \text{eff}^-(x)} \neg p \right)$ and $\text{eff}^+(x) \cap \text{eff}^-(x) = \emptyset$.

So basic actions and events are STRIPS-like. The functions $\text{pre}, \text{post}, \text{eff}^+$ and eff^- are extended to sets, e.g. $\text{pre}(X) = \bigwedge_{x \in X} \text{pre}(x)$ for $X \subseteq \text{Act}_0 \cup \text{Evt}_0$.

We use $|S|$ to denote the cardinality of a set S . We use $\text{len}(\varphi)$ to denote the length of a formula φ which is the number of symbols used to write down φ except for parentheses. The length of a dynamic theory \mathcal{T} , denoted by $\text{len}(\mathcal{T})$, is the sum of the length of all pre- and postcondition formulas in \mathcal{T} .

Definition 2. A dynamic theory \mathcal{T} is coherent if and only if for every basic action $a \in \text{Act}_0$ and event set $E \subseteq \text{Evt}_0$, if $\text{pre}(\{a\} \cup E)$ is consistent then $\text{post}(\{a\} \cup E)$ is consistent.

Proposition 1. A dynamic theory \mathcal{T} is coherent iff the following formula, denoted by $\text{Coh}(\mathcal{T})$, is valid:

$$\bigwedge_{\substack{e \in \text{Evt}_0, x \in \text{Act}_0 \cup \text{Evt}_0, \\ (\text{eff}^+(e) \cap \text{eff}^-(x)) \cup (\text{eff}^-(e) \cap \text{eff}^+(x)) \neq \emptyset}} (\text{pre}(e) \wedge \text{pre}(x) \rightarrow \perp). \quad (1)$$

Proof. “ \Rightarrow ”: Suppose dynamic theory \mathcal{T} is coherent and $\text{post}(\{a\} \cup E)$ is inconsistent. Because all basic actions and events have a consistent postcondition in form of a conjunction of literals, only a pair of an action or event $x \in \{a\} \cup E$ and an event $e \in E$ such that one has a positive effect on propositional variable p and the other has a negative effect on p , would make $\text{post}(\{x, e\})$ inconsistent and further $\text{post}(\{a\} \cup E)$ inconsistent. definition of coherence their jointly precondition $\text{pre}(\{x, e\})$ is inconsistent. Thus $\text{Coh}(\mathcal{T})$ is valid.

“ \Leftarrow ”: Suppose $\text{Coh}(\mathcal{T})$ is valid and there exists some action a and event set E such that $\text{pre}(\{a\} \cup E)$ is consistent while $\text{post}(\{a\} \cup E)$ is inconsistent. As $\text{post}(a)$ and $\text{post}(E)$ can be rewritten into a conjunction of literals, there is a pair of p and $\neg p$ occurring in $\text{post}(\{a\} \cup E)$. Then there are $x, y \in \{a\} \cup E$ such that $x \neq y$ and $p \in \text{eff}^+(x) \cap \text{eff}^-(y)$. Being a conjunct of $\text{Coh}(\mathcal{T})$, $\text{pre}(x) \wedge \text{pre}(y) \rightarrow \perp$ is true, which entails $\text{pre}(a) \wedge \text{pre}(E) \rightarrow \perp$, contradicting that $\text{pre}(\{a\} \cup E)$ is consistent.

Theorem 1 (Complexity of Coherence). *Given any dynamic theory \mathcal{T} , to decide whether \mathcal{T} is coherent is co-NP-complete.*

Proof. As the length of the formula $Coh(\mathcal{T})$ is bounded by $O(|\text{Act}_0 \cup \text{Evt}_0|^2 \times \text{len}(\mathcal{T}))$, the problem of deciding coherence is in co-NP.

To establish hardness, let $\text{Act}_0 = \{a\}$, $\text{Evt}_0 = \{e\}$ with $\text{pre}(a) = \text{pre}(e) = \varphi$, $\text{post}(a) = p$ and $\text{post}(e) = \neg p$. As $\text{post}(a) \wedge \text{post}(e)$ is inconsistent, φ is inconsistent iff \mathcal{T} is coherent. It follows that deciding coherence is co-NP-hard.

Therefore, to decide whether \mathcal{T} is coherent is co-NP-complete.

2.2 Belief-Intention Databases

An agent's belief-intention database contains her intentions plus her beliefs about initial state and event occurrences which may be incomplete.

Occurrence of an event $e \in \text{Evt}_0$ at time point t is noted (t, e) . The non-occurrence of events is also considered: let $\overline{\text{Evt}}_0 = \{\bar{e} : e \in \text{Evt}_0\}$ be the set of event complements. Non-occurrence of e is noted (t, \bar{e}) . An intention is a triple $i = (t, \alpha, d) \in \mathbb{N}^0 \times \text{Act} \times \mathbb{N}$ with $t < d$. It means that the agent wants to perform α in the time interval $[t, d]$: action α should start after t and end before d . When $\alpha \in \text{Act}_0$ then i is a *basic intention*.

Definition 3. *A database is a finite set*

$$\Delta \subseteq (\mathbb{N}^0 \times \mathcal{L}_{\mathbb{P}}) \cup (\mathbb{N}^0 \times \text{Evt}_0) \cup (\mathbb{N}^0 \times \overline{\text{Evt}}_0) \cup (\mathbb{N}^0 \times \text{Act} \times \mathbb{N}).$$

Given an intention $i = (t, \alpha, d)$, we define $\text{end}(i) = d$. For a database Δ , we let $\text{end}(\Delta)$ be the greatest time point occurring in Δ . When $\Delta = \emptyset$, $\text{end}(\Delta) = 0$. This is well defined because databases are finite.

2.3 Semantics

The semantics of dynamic theories and databases is in terms of *paths*. A path defines for each time point which propositional variables are true, which basic actions the agent will perform, and which events will occur.

Definition 4. *A \mathcal{T} -path is a triple $\pi = \langle V, H, D \rangle$ with $V : \mathbb{N}^0 \rightarrow 2^{\mathbb{P}}$, $H : \mathbb{N}^0 \rightarrow 2^{\text{Evt}_0}$, and $D : \mathbb{N}^0 \rightarrow \text{Act}_0$.*

So a path π associates to every time point t a valuation $V(t)$ (alias a state), a set of events $H(t)$ happening at t , and a basic action $D(t)$ performed at t .

Definition 5. *A model of \mathcal{T} , or \mathcal{T} -model, is a path $\pi = \langle V, H, D \rangle$ such that for every time point $t \in \mathbb{N}^0$,*

$$\text{eff}^+(H(t) \cup \{D(t)\}) \cap \text{eff}^-(H(t) \cup \{D(t)\}) = \emptyset \quad (2)$$

$$\text{and} \quad V(t+1) = (V(t) \cup \text{eff}^+(H(t) \cup \{D(t)\})) \setminus \text{eff}^-(H(t) \cup \{D(t)\})$$

$$H(t) = \{e \in \text{Evt}_0 \mid V(t) \models \text{pre}(e)\}$$

$$D(t) \in \{a \in \text{Act}_0 \mid V(t) \models \text{pre}(a)\}$$

So in a \mathcal{T} -model: (1) the state at $t + 1$ is determined by the state at t and the basic action and events occurring at t ; (2) event e occurs *iff* $\mathbf{pre}(e)$ is true; (3) basic action a occurs *implies* that $\mathbf{pre}(a)$ is true. Next we show the satisfaction relation $\Vdash_{\mathcal{T}}$ between a path and an intention or a database.

Definition 6. *Intention $i = (t, \alpha, d)$ is satisfied at a path $\pi = \langle V, H, D \rangle$, noted $\pi \Vdash_{\mathcal{T}} i$, if there exist t', d' such that $t \leq t' < d' \leq d$, $V(t') \models \mathbf{pre}(\alpha)$, $V(d') \models \mathbf{post}(\alpha)$, and $\alpha \in \text{Act}_0$ implies $D(t') = \alpha$.*

So π satisfies (t, α, d) if α is executable at some point after t and can end before the deadline at a point where the postcondition of α is true. Moreover, when α is basic then it conforms to the ‘do’-function D of π .

Definition 7. *A \mathcal{T} -model $\pi = \langle V, H, D \rangle$ is a \mathcal{T} -model of Δ , noted $\pi \Vdash_{\mathcal{T}} \Delta$, if*

- for every $(t, \varphi) \in \Delta$: $V(t) \models \varphi$;
- for every $(t, e) \in \Delta$: $e \in H(t)$;
- for every $(t, \bar{e}) \in \Delta$: $e \notin H(t)$;
- for every $i \in \Delta$: $\pi \Vdash_{\mathcal{T}} i$.

When $\pi \Vdash_{\mathcal{T}} \Delta$, the agent’s beliefs about the state and the (non-)occurrence of events are correct w.r.t. π , and all intentions in Δ are satisfied on π . A database Δ is *\mathcal{T} -satisfiable* when Δ has a \mathcal{T} -model.

Δ is a *\mathcal{T} -consequence* of Δ' , noted $\Delta' \Vdash_{\mathcal{T}} \Delta$, if every \mathcal{T} -model of Δ' is also a \mathcal{T} -model of Δ . When Δ is a singleton $\{i\}$ we write $\Delta' \Vdash_{\mathcal{T}} i$ instead of $\Delta' \Vdash_{\mathcal{T}} \{i\}$.

2.4 Refinement and Instrumentality

Refinement consists in adding new intentions to the database while staying consistent. Intuitively, to refine an intention i means to add a minimal set of new intentions J to the database which, together with other intentions but i , suffice to entail i . Moreover, the deadlines of the means are before that of the end.

Definition 8. *Intention i is refined by intention set J in Δ , noted $\Delta \Vdash_{\mathcal{T}} i \triangleleft J$, iff*

1. there is no $j \in J$ such that $\Delta \Vdash_{\mathcal{T}} j$;
2. $\Delta \cup J$ has a \mathcal{T} -model;
3. $(\Delta \cup J) \setminus \{i\} \Vdash_{\mathcal{T}} i$;
4. $(\Delta \cup J') \setminus \{i\} \not\Vdash_{\mathcal{T}} i$ for every $J' \subset J$;
5. $\text{end}(j) \leq \text{end}(i)$ for every $j \in J$.

A higher-level intention and the lower-level intentions refining it should stand in an instrumentality relation: the lower-levels contribute to the higher-levels.

Definition 9. *For a \mathcal{T} -satisfiable database Δ , let intention $i \in \mathcal{I}(\Delta)$ and intention set $J \subseteq \mathcal{I}(\Delta)$. Then J is instrumental for i in Δ , noted $\Delta \Vdash_{\mathcal{T}} J \triangleright i$, iff*

1. $\Delta \setminus J \not\models_{\mathcal{T}} i$;
2. $(\Delta \setminus J) \cup \{j\} \models_{\mathcal{T}} i$ for every $j \in J$;
3. $\text{end}(j) \leq \text{end}(i)$ for every $j \in J$.

When $\Delta \models_{\mathcal{T}} J \triangleright i$ then J is a minimal set of intentions satisfying the counterfactual “if J was not in Δ then i would no longer be guaranteed by Δ ” and all intentions of J terminate before or together with i . Note that when $\Delta \models_{\mathcal{T}} J \triangleright i$ then J cannot be empty (because we require $i \in J$).

Instrumentality is connected with intention refinement: when $\Delta \models_{\mathcal{T}} i \triangleleft J$ then every element of J is instrumental for i in the refined database. Formally, $\Delta \cup J \models_{\mathcal{T}} \{i, j\} \triangleright i$ for every $j \in J$.

3 Complexity

In this section, we show the complexity results of the decision problems for satisfiability, consequence, refinement and instrumentality in belief-intention databases.

3.1 Complexity of Satisfiability

The coherence condition guarantees that there is no conflict on the effect of the action and the events occurring simultaneously at every time points, entailing the constraint formula (2) in the definition of \mathcal{T} -model (Definition 5). That is, if dynamic theory \mathcal{T} is coherent, then the empty database is \mathcal{T} -satisfiable. Given a coherent dynamic theory, it is not necessary to check whether the infinite path is a \mathcal{T} -model of a database and we only need to check the former part bounded by the greatest time point occurring in the database.

We define the restriction of natural number set \mathbb{N} by a natural number δ as a set of sequential natural numbers $[0, \dots, \delta]$, denoted by \mathbb{N}_{δ} . We define the restriction of a function $f : \mathbb{N} \rightarrow S$ such that the domain is natural number set to a natural number δ as $f|_{\delta} = \{(n, s) \mid n \in \mathbb{N}_{\delta}\}$.

Then we introduce the notion of bounded paths.

Definition 10. For a path $\pi = \langle V, H, D \rangle$ and a natural number δ we call the tuple $\bar{\pi} = \langle V|_{\delta+1}, H|_{\delta}, D|_{\delta}, \delta \rangle$ a bounded path of π and call δ the bound of $\bar{\pi}$.

Then we define bounded models by bounded paths.

Definition 11 (Bounded \mathcal{T} -model). Given a coherent dynamic theory \mathcal{T} , a \mathcal{T} -model π and a database Δ such that $\pi \models_{\mathcal{T}} \Delta$, we call the bounded path $\bar{\pi}$ of π a bounded model of Δ , noted $\bar{\pi} \models_{\mathcal{T}} \Delta$, if the bound of $\bar{\pi}$ is greater than $\text{end}(\Delta)$.

The coherence condition of the dynamic theory allows us to decide if a database has a model by checking a finite path, stated as follows.

Proposition 2. Given a coherent dynamic theory \mathcal{T} , a database Δ is \mathcal{T} -satisfiable iff Δ has a bounded model.

Proof. “ \Rightarrow ”: Straightforward.

“ \Leftarrow ”: When the dynamic theory \mathcal{T} is coherent, for every time point t , the sets $\mathbf{eff}^+(H(t) \cup \{D(t)\})$ and $\mathbf{eff}^-(H(t) \cup \{D(t)\})$ are totally disjoint. So, there is no state in which the performed action has an effect conflicting with the effect of the events which are happening. Thus, we can construct an infinite \mathcal{T} -model starting from the bounded model according to the definition of \mathcal{T} -models.

The lower bound of the complexity of the satisfiability problem comes from the reduction to a plan-existence problem. A plan-existence problem with bounded horizon is a tuple $\mathcal{P} = \langle \mathcal{I}, \mathcal{G}, \mathcal{T}_{\text{Act}_0}, \delta \rangle$ where δ is a natural number, \mathcal{I} is a subset of propositional variable set \mathbb{P} and \mathcal{G} is a conflict-free conjunction of literals and $\mathcal{T}_{\text{Act}_0}$ is a dynamic theory only for basic actions. The plan-existence problem is to decide whether there exists a sequence of basic actions, called plan, with a length less than δ from a initial state \mathcal{I} to a goal state satisfying \mathcal{G} . The plan-existence problem is PSPACE-complete [7].

Theorem 2. *Given a coherent dynamic theory \mathcal{T} , the \mathcal{T} -satisfiability problem of a belief-intention database is PSPACE-complete.*

Proof. First, we prove the problem is in PSPACE. Suppose the number of intentions in Δ is m . Consider a memory space with the size of $|\mathbb{P}| + 2m$. When the $|\mathbb{P}|$ cells can denote a state, the $2m$ cells can indicate the satisfaction of pre- and postcondition of the action in the corresponding intentions.

Guess a path $\bar{\pi} = \langle V, H, D, \text{end}(\Delta) \rangle$, we can change the $|\mathbb{P}|$ cells according to the valuation of each time point defined by $\bar{\pi}$. Because the basic actions and events are finite, it can be checked whether $\bar{\pi}$ is a \mathcal{T} -model in polynomial time. The $2m$ cells are initially set to 0 and with time point changing, we change the $2m$ cells according to the satisfaction of intentions. To be specific, consider an intention $i = (t, \alpha, d)$, from time point t if $\text{pre}(\alpha)$ is satisfied then the cell corresponding to the precondition of i is set to 1. Then in the following time points once $\text{post}(\alpha)$ is satisfied the cell corresponding to postcondition of i is set to 1. Unless this cell is 1 at time point d , we stop and conclude that the path guessed does not satisfy i and further does not satisfy Δ . Therefore, we can check whether the path is a \mathcal{T} -model of Δ in polynomial time, because every effect, pre- and postcondition is defined as a propositional formula which can be checked to be satisfied by the state in polynomial time. As only finite sets of basic actions, events, and propositional variables can be nondeterministically chosen, deciding whether the path guessed is a \mathcal{T} -model of Δ is in NPSPACE. Because $\text{NPSPACE} = \text{PSPACE}$, the \mathcal{T} -satisfiability problem is in PSPACE.

Next we prove the problem is PSPACE-hard by reducing the plan-existence problem with bounded horizon. For a plan-existence problem $\mathcal{P} = \langle \mathcal{I}, \mathcal{G}, \mathcal{T}_{\text{Act}_0}, \delta \rangle$, we construct a dynamic theory \mathcal{T} by extending $\mathcal{T}_{\text{Act}_0}$ with a high-level action $\text{Goal} \in \text{Act} \setminus \text{Act}_0$ such that $\text{pre}(\text{Goal}) = \top$ and $\text{post}(\text{Goal}) = \mathcal{G}$. We also suppose the event set Evt_0 is empty, entailing that \mathcal{T} is coherent. Suppose $\varphi_{\mathcal{I}} = \bigwedge_{p \in \mathcal{I}} p \wedge \bigwedge_{q \in \mathbb{P} \setminus \mathcal{I}} \neg q$. Then the database $\Delta = \{(0, \varphi_{\mathcal{I}}), (0, \text{Goal}, \delta)\}$ has a \mathcal{T} -model iff there exists a plan in \mathcal{P} . So the \mathcal{T} -satisfiability problem is PSPACE-hard.

Hence the \mathcal{T} -satisfiability problem is PSPACE-complete.

3.2 Complexity of Consequence

Next we will show the complexity of the consequence problem in belief-intention databases which is also PSPACE-complete.

Theorem 3. *Given a coherent dynamic theory \mathcal{T} , the \mathcal{T} -consequence problem deciding whether $\Delta' \models_{\mathcal{T}} \Delta$ is PSPACE-complete.*

Proof. For the lower bound, consider the special case of the consequence problem that $\Delta' \models_{\mathcal{T}} \perp$ which means Δ' is \mathcal{T} -unsatisfiable. Because the \mathcal{T} -satisfiable problem is PSPACE-complete by Theorem 2, the problem deciding whether Δ' is \mathcal{T} -unsatisfiable is co-PSPACE-complete. As $\text{co-PSPACE} = \text{PSPACE}$, the \mathcal{T} -consequence problem is PSPACE-hard.

For the upper bound, suppose the number of intentions in Δ and Δ' is m and m' respectively. As \mathcal{T} is coherent, we can consider the complementary problem deciding whether there exists a path which \mathcal{T} -satisfies Δ' but not Δ . Consider a memory space with a size of $|\mathbb{P}| + 2(m + m')$ where the $|\mathbb{P}|$ cells denote a state and the $2(m + m')$ cells indicate the satisfaction of pre- and postcondition of the actions w.r.t. intentions in Δ and Δ' .

Suppose $k = \max(\text{end}(\Delta), \text{end}(\Delta'))$. Guess a path $\bar{\pi} = \langle V, H, D, k \rangle$, we can change the $|\mathbb{P}|$ cells according to the valuation of each time point defined by $\bar{\pi}$. Because the basic actions and events are finite, it can be checked if $\bar{\pi}$ is a \mathcal{T} -model in polynomial time. The $2(m + m')$ cells are initially set to 0 and with time point changing, we change the $2(m + m')$ cells according to the satisfaction of intentions. Consider an intention $i = (t, \alpha, d)$, in the time points after t if $\text{pre}(\alpha)$ is satisfied then the cell corresponding to the precondition of i is set to 1. Then in the following time points once $\text{post}(\alpha)$ is satisfied the cell corresponding to the postcondition of i is set to 1. Unless this cell is 1 at time point d , we stop and conclude that the path guessed satisfies neither i nor the database containing i . Therefore, it can be checked whether the path is a \mathcal{T} -model of Δ or Δ' in polynomial time, because every effect, pre- and postcondition is defined as a propositional formula which can be decided to be satisfied by the state in polynomial time. As only finite sets of basic actions, events, and propositional variables can be nondeterministically chosen, deciding whether the path guessed is a \mathcal{T} -model of Δ' but not Δ is in NPSPACE. Because $\text{NPSPACE} = \text{PSPACE}$, the \mathcal{T} -consequence problem is in PSPACE.

Hence, the \mathcal{T} -consequence problem is PSPACE-complete.

3.3 Complexity of Refinement and Instrumentality

From the definition of refinement and instrumentality, we know that the satisfiability and consequence problems are subproblems of deciding refinement and instrumentality. So, the problems deciding refinement and instrumentality are both PSPACE-hard. Next we show that these two problems are PSPACE-complete by translating them into several satisfiability and consequence problems.

Theorem 4. *Given a coherent dynamic theory \mathcal{T} and a belief-intention database Δ , to check whether an intention $i \in \Delta$ is refinable to an intention set J in Δ is PSPACE-complete.*

Proof. Condition 1, 2 and 4 are \mathcal{T} -satisfiability problems and Condition 3 is a \mathcal{T} -consequence problem and it is easy to check Condition 5 in polynomial time. As the refinement checking problem can be reduced to several \mathcal{T} -satisfiability problems and a \mathcal{T} -consequence problem which are all in PSPACE, the decision problem of deciding refinement is also in PSPACE. As the \mathcal{T} -satisfiability problem is its subproblem, deciding refinement is PSPACE-complete.

Theorem 5. *Given a coherent dynamic theory \mathcal{T} and a belief-intention database Δ , to decide whether an intention set is instrumental for an intention in Δ is PSPACE-complete.*

Proof. (1) Condition 1 is a \mathcal{T} -satisfiability problem; (2) Condition 2 is a set of \mathcal{T} -consequence problems with a number of $|J|$; (3) it is easy to check condition 3 in polynomial time. As the instrumentality checking problem can be reduced polynomially to a \mathcal{T} -satisfiability problem and a \mathcal{T} -consequence problem which are both in PSPACE, the decision problem of deciding instrumentality is also in PSPACE. As the \mathcal{T} -satisfiability problem is its subproblem, deciding instrumentality is PSPACE-complete.

4 Translating to PLTL

Linear temporal logics are widely used to describe infinite behaviors of discrete systems. In the last decades, different model checking techniques for PLTL have been developed, such as approaches based on binary decision diagrams (BDD-based) [6] and based on propositional satisfiability problems (SAT-based) [3]. Besides, by translating to Büchi automata, kinds of satisfiability and validity checkers for PLTL have been developed, such as LTL3BA [1] and SPOT [9]. Furthermore, the tableau-based decision procedure for the satisfiability problem of PLTL has been studied [15, 17].

In this section we translate the satisfiability and consequence problems in belief-intention databases into the satisfiability and validity problems of PLTL. As the problems of deciding refinement and instrumentality are based on the satisfiability and consequence problems, we can further translate them into PLTL. The translations are not in polynomial time,² but nevertheless we believe that the translations are of assistance to solve the decision problems of databases by taking advantage of theorem provers of PLTL.

Following the notations in [10], we define PLTL on a countably infinite set \mathbb{P}_L of propositional variables, classical propositional connectives and restrict it on the unique temporal operator \mathbf{X} (next). Propositional connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

² For the database, time points are encoded in a binary way while they are considered as decimal in the size of the resulted PLTL formula. Therefore, the size of the resulted formula is not polynomial with respect to the size of the database.

in PLTL formulas are defined in the standard way with abbreviating \mathbf{X}^n as n continuous operator \mathbf{X} where \mathbf{X}^0 is nothing.

Next, we introduce the semantics of PLTL which is based on a linear-time structure. A linear-time structure is a pair of $M = (S, \varepsilon)$ where S is a set of states and $\varepsilon : S \rightarrow 2^{\mathbb{P}_L}$ is a function mapping each state s_i to a set of propositional variables which hold in s_i . Let M be a linear-time structure, $i \in \mathbb{N}^0$ a position, and φ, ψ are PLTL formulas. We define the satisfiable relation \models as follows:

$$\begin{aligned} M, i &\models p \text{ iff } p \in \varepsilon(s_i), \text{ where } p \in \mathbb{P}_L \\ M, i &\models \neg\varphi \text{ iff } M, i \not\models \varphi \\ M, i &\models \varphi \wedge \psi \text{ iff } M, i \models \varphi \text{ and } M, i \models \psi \\ M, i &\models \mathbf{X}\varphi \text{ iff } M, i+1 \models \varphi \end{aligned}$$

If there exists a linear-time structure M such that $M, 0 \models \varphi$, we say φ is satisfiable. If for all linear-time structure M we have $M, 0 \models \varphi$, we say φ is valid.

We first start by defining some auxiliary propositional variables. For every event e we introduce an auxiliary propositional variable h_e , defining the set $\mathbb{P}_h = \{h_e | e \in \text{Evt}_0\}$ and for every basic action a we introduce an auxiliary propositional variable do_a , defining the set $\mathbb{P}_d = \{do_a | a \in \text{Act}_0\}$. Moreover, we introduce a set \mathbb{P}_c auxiliary propositional variables $pre_\alpha, post_\alpha, pre_e$ and $post_e$ for every action and event to denote their pre- and postcondition.

Definition 12. *Given a coherent dynamic theory \mathcal{T} , we define a conjunction of formulas $Tr(\mathcal{T})$ as:*

$$\bigwedge_{a \in \text{Act}_0} (do_a \rightarrow \mathbf{pre}_a) \wedge \bigwedge_{e \in \text{Evt}_0} (h_e \leftrightarrow \mathbf{pre}_e) \quad (3)$$

$$\bigwedge_{a \in \text{Act}_0} do_a \wedge \bigwedge_{a, b \in \text{Act}_0, a \neq b} \neg(do_a \wedge do_b) \quad (4)$$

$$\bigwedge_{p \in \mathbb{P}} (\mathbf{X}p \leftrightarrow \bigvee_{\substack{e \in \text{Evt}_0 \\ p \in \mathbf{eff}^+(e)}} h_e \vee \bigvee_{\substack{a \in \text{Act}_0 \\ p \in \mathbf{eff}^+(a)}} do_a \vee (p \wedge \bigwedge_{\substack{e \in \text{Evt}_0 \\ p \in \mathbf{eff}^-(e)}} \neg h_e \wedge \bigwedge_{\substack{a \in \text{Act}_0 \\ p \in \mathbf{eff}^-(a)}} \neg do_a)) \quad (5)$$

$$\bigwedge_{\alpha \in \text{Act}} ((pre_\alpha \leftrightarrow \mathbf{pre}(\alpha)) \wedge (post_\alpha \leftrightarrow \mathbf{post}(\alpha))) \quad (6)$$

$$\bigwedge_{e \in \text{Evt}_0} ((pre_e \leftrightarrow \mathbf{pre}(e)) \wedge (post_e \leftrightarrow \mathbf{post}(e))) \quad (7)$$

Intuitively, formula (3) means that basic action a is executable if its precondition is satisfied and that events are reactive: when their precondition is satisfied they will happen. Formula (4) says that exactly one basic action is allowed at one time point. Formula (5) means propositional variable p is true in the next state if and only if either it is “activated” or both it is already currently true and there is no action or event making it false. Formulas (6) and (7) link the formulas of pre- and postcondition of actions and events with propositional variables. Finally, $\text{Tr}(\mathcal{T})$ captures the definition and progression of valuations in one time point. Then we define $\text{Tr}(n, \mathcal{T})$ as $\text{Tr}(\mathcal{T}) \wedge \mathbf{X}\text{Tr}(\mathcal{T}) \wedge \dots \wedge \mathbf{X}^n \text{Tr}(\mathcal{T})$ to capture the first n time points of a \mathcal{T} -model.

Definition 13. *We translate a database Δ into a conjunction of formulas $\text{Tr}(\Delta)$ as:*

$$\bigwedge_{(t, \varphi) \in \Delta} \mathbf{X}^t \varphi \wedge \bigwedge_{(t, e) \in \Delta} \mathbf{X}^t h_e \wedge \bigwedge_{(t, \bar{e}) \in \Delta} \mathbf{X}^t \neg h_{\bar{e}} \quad (8)$$

$$\wedge \bigwedge_{\substack{\alpha \notin \text{Act}_0 \\ (t, \alpha, d) \in \Delta}} (\mathbf{X}^{t'} \text{pre}_\alpha \wedge \mathbf{X}^{d'} \text{post}_\alpha) \quad (9)$$

$$\wedge \bigwedge_{\substack{a \in \text{Act}_0 \\ (t, a, d) \in \Delta}} \mathbf{X}^{t'} do_a \quad (10)$$

The above definition actually formalizes the satisfaction of database in a path. For a coherent dynamic theory, we only consider the fragment of \mathcal{T} -model from time point 0 to $\text{end}(\Delta)$. The following proposition shows the satisfiability problem of database is connected to the satisfiability problem of PLTL.

Proposition 3. *Given a coherent dynamic theory \mathcal{T} , a database Δ is \mathcal{T} -satisfiable iff $\text{Tr}(\text{end}(\Delta), \mathcal{T}) \wedge \text{Tr}(\Delta)$ is satisfiable.*

Proof. Let $M = (S, \varepsilon)$ be a linear-time structure where $S = \{s_0, s_1, \dots\}$ and $\varepsilon : S \rightarrow 2^{\mathbb{P}_L}$ such that $\mathbb{P}_L = \mathbb{P} \cup \mathbb{P}_d \cup \mathbb{P}_h \cup \mathbb{P}_c$.

“ \Rightarrow ” : Suppose there exists a \mathcal{T} -model $\pi = \langle V, H, D \rangle$ of Δ . Let us build a linear-time structure M as follows: for every time point t , (i) $\varepsilon(s_t) \cap \mathbb{P} = V(t)$; (ii) $\varepsilon(s_t) \cap \mathbb{P}_h = \{h_e | e \in H(t)\}$; (iii) if $D(t) = a$ then $\varepsilon(s_t) \cap \mathbb{P}_d = do_a$; (iv) $\varepsilon(s_t) \cap \mathbb{P}_c = \{\text{pre}_x, \text{post}_y | V(t) \models \text{pre}(x), V(t) \models \text{post}(y), x, y \in \text{Act} \cup \text{Evt}_0\}$.

Now we first prove $M, 0 \models \text{Tr}(\text{end}(\Delta), \mathcal{T})$. According to the definition of \mathcal{T} -model, we immediately have $M, t \models (3) \wedge (4) \wedge (6) \wedge (7)$ for each time point t . For every propositional variable p , $M, t+1 \models p$ iff either there is a basic action or event to make it true or $M, t \models p$ and there is no action or event to make it false. Thus, for every time point t , we obtain $M, t \models \text{Tr}(\mathcal{T})$ and then $M, 0 \models \text{Tr}(\text{end}(\Delta), \mathcal{T})$. Next we prove $M, 0 \models \text{Tr}(\Delta)$. For every $(t, \varphi) \in \Delta$, we obtain $M, t \models \varphi$. For every $(t, e) \in \mathcal{E}(\Delta)$, $M, 0 \models \mathbf{X}^t h_e$ because $e \in H(t)$ and $h_e \in \varepsilon(s_t)$. The case of (t, \bar{e}) is similar. By the definition of satisfying an intention (Definition 6), if $\pi \Vdash_{\mathcal{T}} i$ we have $M, 0 \models (9) \wedge (10)$. So we have $M, 0 \models \text{Tr}(\Delta)$.

Thus, we conclude that $M, 0 \models \text{Tr}(\text{end}(\Delta), \mathcal{T}) \wedge \text{Tr}(\Delta)$.

“ \Leftarrow ” : Suppose there exists a linear-time structure M such that $M, 0 \models \text{Tr}(\text{end}(\Delta), \mathcal{T}) \wedge \text{Tr}(\Delta)$. Now we build a path $\pi = \langle V, H, D \rangle$ as follows: for every time points $t \leq \text{end}(\Delta)$, (i) $V(t) = \varepsilon(s_t) \cap \mathbb{P}$; (ii) $H(t) = \{e \mid h_e \in \varepsilon(s_t) \cap \mathbb{P}_h\}$; (iii) $D(t) = a$ if $do_a \in \varepsilon(s_t)$. For those time points $t > \text{end}(\Delta)$, we can construct π according to Definition 5, because \mathcal{T} is coherent.

Next we show π is a \mathcal{T} -model. For every time point $t \leq \text{end}(\Delta)$, due to (4), there must be an action a such that $M, t \models do_a$. So $M, t \models \text{pre}(a)$ then we have $V(t) \models \text{pre}(a)$ and $D(t) = a$. Because $M, t \models h_e$ iff $M, t \models \text{pre}(e)$, we have $H(t) = \{e \mid V(t) \models \text{pre}(e)\}$. For propositional variable p , it is in $V(t+1)$ iff either at time point t , there exists an action or event making it true or both $p \in V(t)$ and there is no action or event making it false. The constraint formula (2) of \mathcal{T} -models is satisfied because \mathcal{T} is coherent. So, we have π is a \mathcal{T} -model.

As Definition 7, it is easy to prove $\pi \Vdash_{\mathcal{T}} \Delta$ for $M, 0 \models \text{Tr}(\Delta)$.

The next proposition states the equivalence between the consequence problem in belief-intention databases and the validity problem in PLTL.

Proposition 4. *Given a coherent dynamic theory \mathcal{T} , Δ' is a \mathcal{T} -consequence of Δ iff $\text{Tr}(k, \mathcal{T}) \rightarrow (\text{Tr}(\Delta) \rightarrow \text{Tr}(\Delta'))$ is valid where k is the greater number between $\text{end}(\Delta)$ and $\text{end}(\Delta')$.*

Proof. “ \Rightarrow ”: for all \mathcal{T} -models of Δ , we have $\text{Tr}(\text{end}(\Delta), \mathcal{T}) \wedge \text{Tr}(\Delta)$. Then if these models are also models of Δ' , then $(\text{Tr}(\text{end}(\Delta), \mathcal{T}) \wedge \text{Tr}(\Delta)) \rightarrow (\text{Tr}(\text{end}(\Delta'), \mathcal{T}) \wedge \text{Tr}(\Delta'))$. Because \mathcal{T} -model is infinite on time, either $\text{end}(\Delta) \geq \text{end}(\Delta')$ or not, $\text{Tr}(k, \mathcal{T})$ must be satisfied. Thus, we have $\text{Tr}((k, \mathcal{T}) \rightarrow (\text{Tr}(\Delta) \rightarrow \text{Tr}(\Delta'))$.

“ \Leftarrow ”: from the proof of Proposition 3, if $\text{Tr}(k, \mathcal{T})$ is satisfied we can construct a \mathcal{T} -model π . Further if $\text{Tr}(\Delta)$ is satisfied then $\pi \Vdash_{\mathcal{T}} \Delta$. Thus, if $\text{Tr}(\Delta) \rightarrow \text{Tr}(\Delta')$, then we have $\Delta \models_{\mathcal{T}} \Delta'$.

As shown in the proof of Theorems 4 and 5, the problems of deciding refinement and instrumentality are based on the satisfiability and consequence problems. By Propositions 3 and 4, we can further translate these two decision problems into the satisfiability and validity problems of PLTL.

5 Conclusion

In this paper, we investigate the complexity of the decision problems for satisfiability, consequence, refinement and instrumentality in belief-intention databases and prove these problems are all PSPACE-complete. Moreover, we translate the satisfiability and consequence problems, and further, the problems of deciding refinement and instrumentality, in belief-intention databases into the satisfiability and validity problems of PLTL. With such reductions, the state of the art in the automated tools of PLTL contributes to develop an implementation for refining high-level intentions in the belief-intention databases.

Intention refinement is closed to Hierarchical Task Network (HTN) planning where higher-level actions are refined step-by-step into lower-level actions. In

HTN planning, refinement is defined in an explicit way while in the database the refinement is given in a derived way. The former requires the user to think throughout all possible refinement ways for all actions, which is a big challenge. Considering refinement in a derived way helps us to complete the refinement ways by discovering the implicit refinement relation between actions. Postulates of the soundness and completeness for refining actions were proposed in [12] and we believe, they provide rational postulates for improving the HTN domains.

Acknowledgments.. The work was supported by Chinese Scholarship Council and the project ANR-11-LABX-0040-CIMI within ANR-11-IDEX-0002-02.

References

1. Babiak, T., Křetínský, M., Řehák, V., Strejček, J.: LTL to Büchi automata translation: fast and more deterministic. In: Flanagan, C., König, B. (eds.) TACAS 2012. LNCS, vol. 7214, pp. 95–109. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-28756-5_8](https://doi.org/10.1007/978-3-642-28756-5_8)
2. Baral, C., Gelfond, M.: Reasoning about intended actions. In: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI), Menlo Park, CA, vol. 20, pp. 689–694. AAAI Press, MIT Press, Cambridge 1999 (2005)
3. Biere, A., Cimatti, A., Clarke, E.M., Fujita, M., Zhu, Y.: Symbolic model checking using SAT procedures instead of BDDs. In: Proceedings of the 36th annual ACM/IEEE Design Automation Conference, pp. 317–320. ACM (1999)
4. Bratman, M.E.: Intention, Plans, and Practical Reason. Harvard University Press, Cambridge (1987). (reedited 1999 with CSLI Publications)
5. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. *J. Comput. Intell.* **4**, 349–355 (1988)
6. Burch, J.R., Clarke, E.M., McMillan, K.L., Dill, D.L., Hwang, L.J.: Symbolic model checking: 10^{20} states and beyond. *Inf. Comput.* **98**(2), 142–170 (1992)
7. Bylander, T.: The computational complexity of propositional STRIPS planning. *Artif. Intell.* **69**(1), 165–204 (1994)
8. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *J. Artif. Intell.* **42**(2), 213–261 (1990)
9. Duret-Lutz, A., Poitrenaud, D.: Spot: an extensible model checking library using transition-based generalized büchi automata. In: Proceedings of the IEEE Computer Society’s 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS 2004), pp. 76–83. IEEE (2004)
10. Emerson, E.A.: Temporal and modal logic. *Handb. Theor. Comput. Sci. Volume B: Formal Models Semantics (B)* **995**(1072), 5 (1990)
11. Herzig, A., Lorini, E., Perrussel, L., Xiao, Z.: BDI logics for BDI architectures: old problems, new perspectives. *KI - Künstliche Intelligenz* **31**(1), 73–83 (2017)
12. Herzig, A., Perrussel, L., Xiao, Z.: On hierarchical task networks. In: Michael, L., Kakas, A. (eds.) JELIA 2016. LNCS (LNAI), vol. 10021, pp. 551–557. Springer, Cham (2016). doi:[10.1007/978-3-319-48758-8_38](https://doi.org/10.1007/978-3-319-48758-8_38)
13. Herzig, A., Perrussel, L., Xiao, Z., Zhang, D.: Refinement of intentions. In: Michael, L., Kakas, A. (eds.) JELIA 2016. LNCS (LNAI), vol. 10021, pp. 558–563. Springer, Cham (2016). doi:[10.1007/978-3-319-48758-8_39](https://doi.org/10.1007/978-3-319-48758-8_39)

14. Sardina, S., de Silva, L., Padgham, L.: Hierarchical planning in BDI agent programming languages: a formal approach. In: Proceedings of the 5th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1001–1008 (2006)
15. Shilov, N.V.: Designing tableau-like axiomatization for propositional linear temporal logic at home of arthur prior. Bull. Novosibirsk Comput. Cent. Ser.: Comput. Sci. **23**, 113–136 (2005)
16. Shoham, Y.: Logical theories of intention and the database perspective. J. Philos. Logic **38**(6), 633–647 (2009)
17. Wolper, P.: The tableau method for temporal logic: an overview. Logique et Analyse **28**(110–111), 119–136 (1985)