

# A General Representation and Approximate Inference Algorithm for Sensing Actions

Hanne Vlaeminck<sup>1,\*</sup>, Joost Vennekens<sup>1,2</sup>, and Marc Denecker<sup>1</sup>

<sup>1</sup> Department of Computer Science, KU Leuven

<sup>2</sup> Campus De Nayer, Lessius Mechelen, Sint-Katelijne-Waver

**Abstract.** Sensing actions, which allow an agent to increase its knowledge about the environment, are problematic for traditional planning languages. In this paper we propose a very general framework for representing both changes to the real world and to the knowledge of an agent, based on a first order linear time calculus. Our framework is more general than most existing approaches, because our semantics explicitly represents, for each point in time, not only the agent's knowledge about that timepoint, but also about the past and the future. By applying a general approximation method for classical logic to this framework, we obtain an efficient and sound but incomplete reasoning method.

## 1 Introduction

Classical planning assumes that agents have complete information about the world. A more realistic assumption, however, is that they only partially know their environment, and must use *sensing actions* to gain additional knowledge. Several approaches exist that formalize such reasoning [9,12,14,7,13,4,5]. One thing that is lacking from these approaches, however, is that their semantics do not (explicitly) model that, at every point in time, an agent can also have knowledge about both the past and the future, and not just about the current time point itself. For example, an agent might initially already know that he will *never* be able to do a certain action and that he therefore will not be able to reach a certain state. Conversely, by observing the effects of one of its past actions, the agent might learn that some property held at the time of this action, even if it currently no longer does. In this paper, we propose a more general semantic framework, based on a linear time calculus, which supports this.

The aforementioned approaches, as well as ours, all use a Kripke-style representation of the agent's knowledge. While this is semantically the right way of doing it, it is also computationally quite challenging. Therefore, several other approaches use more limited representations of knowledge with better computational properties [1,8,10,15]. Ideally, it would be possible to view such an approach as an incomplete approximation, in some sense, of a semantically correct approach. However, proving such a connection is often non-trivial [11]. An interesting property of our framework is that it allows us to apply a general approximation method for first-order logic (FO) [17] to obtain an incomplete method

---

\* Hanne Vlaeminck is supported by IWT-Vlaanderen.

for solving the projection problem in polynomial time. The soundness of this method follows immediately from the fact that this approximation is sound for FO in general. Even though the method is incomplete, it is able to reach conclusions about properties that must have held at an earlier point in time, even if they no longer hold now. This is something that other incomplete reasoning methods either cannot do [15,8,1], or can only do in an ad-hoc way [10].

## 2 Preliminaries

We assume familiarity with first order logic (FO). For simplicity, we consider only relational vocabularies (no function symbols of arity  $> 0$ ). As usual, an *interpretation*  $S$  for a vocabulary  $\Sigma$  consists of a non-empty domain  $D$ , a mapping from each constant symbol  $c$  to a domain element  $d \in D$ , and a mapping from each predicate symbol  $P/n$  to a relation  $R \subseteq D^n$ .

**Definition 1.** A Linear Time Calculus vocabulary  $\Sigma_{ltc}$  consists of

- a set of types, including a type *Time*,
- for every type  $S$ , a set of constants  $\Sigma_S$  of type  $S$ ,
- a set  $\Sigma_{stat}$  of static predicate symbols without an argument of type *Time*,
- a set  $\Sigma_{dyn}$  of dynamic predicate symbols that have exactly one temporal argument;  $\Sigma_{dyn}$  is divided into a set  $\Sigma_{act}$  of actions and  $\Sigma_{flu}$  of fluents,
- a set  $\Sigma_{init}$  containing for every predicate  $P/n \in \Sigma_{flu}$  a predicate  $Init\_P/n-1$  having only the non-temporal arguments of  $P$ .

Throughout this paper, we assume that every type  $S$  apart from *Time* has only a finite number of constants and restrict attention to Herbrand interpretations of  $S$ , i.e., each constant is interpreted by itself. We will fix the interpretation of *Time* to the natural numbers  $\mathbb{N}$ , and use constants  $0, 1, \dots$ , the arithmetic functions  $+$ , and the comparison operator  $\leq$ , also all fixed to their usual interpretation in  $\mathbb{N}$ .

This paper uses the logic FO(ID) [3], an extension of FO with a construct to represent common forms of inductive definitions, such as monotone induction and induction over a well-founded order. Such a definition is represented by a set of logic programming-style rules of the form  $P(\bar{t}) \leftarrow \varphi$ , with  $\varphi$  an FO formula. The semantics of such a definition is given by a parametrized version of the well-founded semantics for logic programs, which construct an interpretation for the defined predicates (those that appear in the head of a rule) given an input interpretation for the open predicates (those that do not), by applying the rules in appropriate way. Due to space restrictions, we refer to [3] for details.

To represent a dynamic domain, one of the central problems is to define the values of the fluents at different points in time, in terms of their initial values and the actions that happen. The inductive definition construct of FO(ID) allows this to be done in a natural and elegant way [2].

*Example 1.* An agent has a glass, which can be clean or not. A glass can be used for drinking, which makes the glass dirty, and it can be cleaned by wiping it. The agent's room has a light, which can be switched on and off.

This domain can be represented in FO(ID) by the following inductive definition  $\Delta_{flu}$ , which defines fluents  $Clean(t)$  and  $Light(t)$  in terms of their initial values  $Init\_Clean$  and  $Init\_Light$ , and of the actions  $Wipe(t)$ ,  $Drink(t)$  and  $Switch(t)$ .

$$\Delta_{flu} = \left\{ \begin{array}{ll} Clean(t+1) \leftarrow (Clean(t) \wedge \neg Drink(t)) \vee Wipe(t). \\ Clean(0) \leftarrow Init\_Clean. \\ Light(t+1) \leftarrow (\neg Switch(t) \wedge Light(t)) \vee (Switch(t) \wedge \neg Light(t)). \\ Light(0) \leftarrow Init\_Light. \end{array} \right\}.$$

### 3 Representing Knowledge and Sensing Actions

When an agent operates in a partially known environment, its actions will be constrained by its knowledge of the world. At the same time, its knowledge may also evolve through sensing actions. Let us extend Example 1 as follows.

*Example 2.* The agent may only drink if it *knows* the glass to be clean. If the light is on, then the sensing action of *inspecting* the glass will reveal if it is clean.

To handle such examples, we introduce two new language constructs. The first is a dynamic modal operator  $K(\cdot, t)$  that refers to the agent's knowledge at time  $t$ .

**Definition 2.** For a LTC signature  $\Sigma_{ltc}$ , a modal atom is an expression of the form  $K(\varphi[\bar{x}], t)$ , where  $\varphi[\bar{x}]$  is a formula in  $\Sigma_{ltc}$  and  $t \in \mathbb{N}$ . We define  $\mathcal{L}_K$  as the language that extends FO by allowing both atoms and modal atoms as base cases and closing under the usual operators  $\wedge, \vee, \neg, \forall, \exists$ . An  $\mathcal{L}_K$ -formula is called objective if no modal operator  $K(\cdot, t)$  occurs in it and subjective if every atom occurs in the scope of such a modal operator.

We will use formulas of  $\mathcal{L}_K$  to express that an agent may perform certain actions only if it knows that its preconditions are satisfied.

**Definition 3.** For a LTC vocabulary  $\Sigma_{ltc}$ , a knowledge precondition is a formula  $\forall \bar{x}t (A(\bar{x}, t) \Rightarrow \varphi)$ , where  $A$  is an action and  $\varphi$  a subjective  $\mathcal{L}_K$ -formula.

The precondition that an agent may drink from a glass only if he knows that it is clean can now be expressed by the  $\mathcal{L}_K$ -formula  $\forall t Drink(t) \Rightarrow K(Clean(t), t)$ . Note that the formula  $\varphi$  in a modal atom  $K(\varphi, t)$  may contain its own temporal argument, not necessary equal to  $t$ . For example, if the agent only wants to drink from a glass that has never been used before, this can be expressed as:  $\forall t Drink(t) \Rightarrow K(\forall t' (t' < t \Rightarrow \neg Drink(t')), t)$ . This is more general than most existing approaches, in which it is typically not even syntactically possible to refer to the knowledge in one situation about another situation.

These knowledge preconditions will actually be the only place in which we allow the modal operators  $K(\cdot, t)$ . On the one hand, this is motivated by the fact that the agent has no direct access to the real world, i.e., it can use only its own knowledge to decide upon its actions. On the other hand, the agent also

only affects the state of the world through its actions: the real world does not respond to what the agent *knows* but only to what it *does*. For instance, if the agent enters the correct code into a safe, this will open regardless of whether the agent actually knew the code or was just guessing.

To represent the effects of sensing actions, we introduce the following construct.

**Definition 4.** For a LTC signature  $\Sigma_{ltc}$ , a sensing definition is of the form:

$$A(\bar{x}, t) \textbf{ senses } \xi[\bar{x}, \bar{y}, t + 1] \textbf{ if } \gamma[\bar{x}, \bar{y}, t + 1], \quad (1)$$

where  $A \in \Sigma_{act}$ , and both  $\xi[\bar{x}, \bar{y}, t + 1]$  and  $\gamma[\bar{x}, \bar{y}, t + 1]$  are FO formulas in which  $t + 1$  is the only term of type *Time*. The action  $A$  is called a sensing action. When  $\gamma = \mathbf{t}$ , the sensing action is called unconditional.

Note that a sensing action performed at time  $t$  yields information about the state of the world at the next timepoint  $t + 1$ . For our running example, the *Inspect* action is described by the following sensing definition.

$$Inspect(t) \textbf{ senses } Clean(t + 1) \textbf{ if } Light(t + 1). \quad (2)$$

Putting this all together, we now arrive at the following.

**Definition 5.** Let  $\Sigma_{ltc}$  be an LTC vocabulary. An  $LTC_{K,S}$  theory consist of

- a inductive definition  $\Delta_{flu}$  of the fluents,
- a constraint theory  $T_{constr}$  of FO formulas over  $\Sigma_{act} \cup \Sigma_{stat}$  in which the action predicates appear only negatively,
- an FO theory  $T_{init}$  over  $\Sigma_{init}$  about the initial state of the fluents,
- a precondition theory  $T_{prec}$ , consisting of knowledge preconditions
- a set of sensing definitions, denoted by  $T_{sense}$ .

The purpose of the theory  $T_{constr}$  is to express constraints on the occurrences of actions. However, we only allow to specify that actions cannot occur in certain circumstances, and not that they must sometimes occur. This will make it easier to define the semantics. The theory  $T_{init}$  specifies *all that is known* about the initial situation. In our running example, we take  $T_{init}$  to be empty, meaning that the agent knows *nothing* about the initial situation. Our  $\Delta_{flu}$  is as in Example 1,  $T_{constr}$  is also empty,  $T_{prec}$  consists of the single formula  $\forall t Drink(t) \Rightarrow K(Clean(t), t)$  and  $T_{sense}$  contains expression (2).

### 3.1 Semantics

Our semantics needs to consider both the real world  $I$  and the agent's knowledge about  $I$ . To represent this knowledge, we use a set of interpretations  $\mathcal{K}$ , which we refer to as *possible worlds*. Since each possible world  $K \in \mathcal{K}$  interprets the entire vocabulary, it specifies the value of all actions and fluents at *each* point in time.

**Definition 6.** A knowledge structure  $\mathcal{S}$  is a pair  $(I, \mathcal{K})$  of an interpretation  $I$  and a set of interpretations  $\mathcal{K}$ , such that all these interpretations have the same vocabulary (and therefore also the same domain).

Because we only care about knowledge, and not beliefs that might be false, we often require that knowledge structures be *consistent*, in the sense that  $I \in \mathcal{K}$ .

We first focus on the semantics of the sensing definitions. The purpose of these is to allow the agent to increase its knowledge. For a set  $\Psi$  of sentences, we denote by  $\mathcal{K} \circ \Psi$  the set of all  $J \in \mathcal{K}$  for which  $J \models \Psi$ . The smaller set of possible worlds  $\mathcal{K} \circ \Psi$  now describes the agent's knowledge after it has learned  $\Psi$  in a situation where it already knew  $\mathcal{K}$ . We will now define a set of formulas  $\Psi$  that captures precisely what the agent learns from a sensing action. Without loss of generality, we assume that the theory  $T$  contains at most one sensing definition for each action  $A \in \Sigma_{act}$ . If the unique sensing definition for  $A(\bar{x}, t)$  is of form (1), then we denote the formula  $\xi$  as  $\xi_A$  and the formula  $\gamma$  as  $\gamma_A$ .

**Definition 7.** *Given an interpretation  $I$ , a sensing action  $A(\bar{x}, t)$  and a time point  $i \in \mathbb{N}$ , the scope of  $A(\bar{x}, i)$  in  $I$  is defined as the set of all tuples  $(\bar{d}, \bar{e})$  of constants such that  $I \models \gamma_A[\bar{d}, \bar{e}, i + 1]$ .*

For the sensing action  $Inspect(t)$  in our running example, a timepoint  $i$  and an interpretation  $I$ , the scope is either the empty tuple  $\{()\}$  (i.e., “true”)  $I \models Light(i)$  or the empty set  $\emptyset$  (i.e., “false”) if  $I \not\models Light(i)$ .

**Definition 8.** *For a sensing action  $A(\bar{x}, t)$ , an interpretation  $I$ , a tuple  $\bar{d}$  of constants, and a time point  $i$ , the effect of  $A(\bar{d}, i)$  given  $I$  is denoted as  $Eff_I(A(\bar{d}, i))$  and defined as the set of all formulas  $\varphi[\bar{d}, \bar{e}, i + 1]$  such that  $\varphi$  is either  $\xi_A$  or  $\neg\xi_A$ , the tuple  $(\bar{d}, \bar{e})$  is in the scope of  $A(\bar{x}, i)$ , and  $I \models \varphi[\bar{d}, \bar{e}, i + 1]$ .*

Intuitively, the set  $Eff_I(A(\bar{d}, i))$  contains all ground formulas that the agent learns by performing the sensing action  $A(\bar{d}, i)$  in the real world  $I$ . To illustrate with our running example, consider an interpretation  $I$  such that  $Light^I = \mathbb{N}$ ,  $Clean^I = \{\}$  and  $Inspect^I = \{0\}$ . Then  $Eff_I(Inspect(0)) = \{\neg Clean(1)\}$ .

In addition to the results of its sensing actions, the agent also has a second source of information, namely, it will also know its own actions at each timepoint. Thus, in total, the agent gains the following information at timepoint  $i$ .

**Definition 9.** *For a time point  $i \geq 0$  and an interpretation  $I$ , the action information at  $i$ , denoted  $Act_I(i)$ , is defined as the set of all literals  $L(\bar{d}, i)$  such that  $I \models L(\bar{d}, i)$  and  $L(\bar{d}, i)$  is either  $A(\bar{d}, i)$  or  $\neg A(\bar{d}, i)$  with  $A \in \Sigma_{act}$ . The  $i^{th}$  update of  $I$ , denoted  $Upd_I(i)$ , is defined as the union of  $Act_I(i)$  and all sets  $Eff_I(A(\bar{d}, i))$  for which  $I \models A(\bar{d}, i)$  and  $A(\bar{x}, t)$  is a sensing action.*

Note that this definition assumes that an agent knows its actions and its effects the next timepoint after performing them. For our running example, we have that  $Upd_I(0) = \{\neg Wipe(0), \neg Switch(0), Inspect(0), \neg Clean(1)\}$ , and  $Upd_I(1) = \{\neg Wipe(1), \neg Switch(1), \neg Inspect(1)\}$ . We now arrive at the following update operation.

**Definition 10.** *For a knowledge structure  $\mathcal{S} = (I, \mathcal{K})$ , the  $i$ -update of  $\mathcal{S}$ , denoted by  $\mathcal{S} \circ i$ , is defined as  $(I, \mathcal{K} \circ Upd_I(i))$ .*

After performing such an update, the agent's new knowledge  $\mathcal{K} \circ \text{Upd}_I(i)$  will contain all the information that was obtained as a direct result of its sensing actions at time  $i - 1$ . However, by *reflecting* on the difference between  $\mathcal{K}$  and  $\mathcal{K} \circ \text{Upd}_I(i)$ , the agent may be able to deduce more information. For instance, if in our running example an *Inspect* action fails to produce knowledge about whether the glass is clean, the agent could deduce that the light must be switched off. So, in addition to ruling out all worlds  $J \not\models \text{Upd}_I(i)$ , the agent could also rule out all worlds  $J \in \mathcal{K}$  in which the knowledge contained in  $\mathcal{K} \circ \text{Upd}_J(i)$  would not have been the same as the knowledge  $\mathcal{K} \circ \text{Upd}_I(i)$  that the agent actually gained in the real world. Let us define that two knowledge structures  $\mathcal{S} = (I, \mathcal{K})$  and  $\mathcal{S}' = (I', \mathcal{K}')$  are *epistemically equal*, denoted  $\mathcal{S} =_k \mathcal{S}'$ , if  $\mathcal{K} = \mathcal{K}'$ . We now define a *second-order* update operator  $(I, \mathcal{K}) \circ^2 i$  as follows.

**Definition 11.** For a knowledge structure  $\mathcal{S} = (I, \mathcal{K})$ , if  $\mathcal{S} \circ i = (I, \mathcal{K}')$ , then  $(I, \mathcal{K}) \circ^2 i$  is defined as  $(I, \mathcal{K}'')$  with  $\mathcal{K}'' = \{J \in \mathcal{K}' \mid (I, \mathcal{K}) \circ i =_k (J, \mathcal{K}) \circ i\}$ .

Based on this principle, we can define an infinite sequence  $(\circ^n)_{n \in \mathbb{N}}$  of ever more introspective update operators.

**Definition 12.** For a knowledge structure  $\mathcal{S} = (I, \mathcal{K})$ , if  $\mathcal{S} \circ^n i = (I, \mathcal{K}_n)$ , then  $(I, \mathcal{K}) \circ^{n+1} i = (I, \mathcal{K}_{n+1})$  with  $\mathcal{K}_{n+1} = \{J \in \mathcal{K}_n \mid (I, \mathcal{K}) \circ^n i =_k (J, \mathcal{K}) \circ^n i\}$ . We define  $\bullet$  as the limit of this sequence, i.e.,  $\mathcal{S} \bullet i = \bigcap_{n \in \mathbb{N}} \mathcal{S} \circ^n i$ .

The limit  $\bullet$  is a strong operator, which allows the agent to reason about its knowledge of its knowledge of its knowledge of ... of its knowledge of the world. Luckily, however, it is never necessary to go more than two levels deep.

**Proposition 1.** For any set of sensing definitions,  $\bullet = \circ^2$ . Moreover, if all of the sensing actions are unconditional, then even  $\bullet = \circ$ .

This operator  $\bullet$  now allows us to define how the agent's knowledge will evolve over time, starting from an initial knowledge structure  $\mathcal{S}_0 = (I, \mathcal{K})$ .

**Definition 13.** A knowledge line  $\mathbb{K}$  is a function that maps each time point  $i \in \mathbb{N}$  to a set  $\mathbb{K}(i)$  of possible worlds. For an initial knowledge structure  $\mathcal{S}_0 = (I, \mathcal{K})$ , we construct the sequence  $(\mathcal{S}_i)_{i \geq 0}$  such that  $\mathcal{S}_{i+1} = \mathcal{S}_i \bullet i$ . By  $\overline{\mathcal{S}_0}$ , we denote the associated knowledge line  $\mathbb{K}$ , i.e., for each  $i$ ,  $\mathbb{K}(i)$  is such that  $\mathcal{S}_i = (I, \mathbb{K}(i))$ .

If  $\mathcal{K}$  represents the agent's initial knowledge (note again that  $\mathcal{K}$  may contain knowledge about *any* point in time), then a knowledge line  $\overline{(I, \mathcal{K})}$  represents how this knowledge evolves over time according to the real world  $I$ . A *knowledge line structure* is now a pair  $(I, \mathbb{K})$  of an interpretation  $I$  and a knowledge line  $\mathbb{K}$ . Since  $I \models \text{Upd}_I(i), \forall i$ , we have that for a consistent knowledge structure  $(I, \mathcal{K})$ , for each  $i$ ,  $I \in \mathbb{K}(i)$ , where  $\mathbb{K} = \overline{(I, \mathcal{K})}$ . It is now straightforward to evaluate formulas of  $\mathcal{L}_K$  in a knowledge line structure  $(I, \mathbb{K})$ .

**Definition 14.** For a formula  $\varphi$  of  $\mathcal{L}_K$ , a knowledge line structure  $(I, \mathbb{K})$  and a variable assignment  $\theta$ , we inductively define the relation  $(I, \mathbb{K}), \theta \models \varphi$  as follows:

- For an atom  $P(\bar{t})$ , we define  $(I, \mathbb{K}), \theta \models P(\bar{t})$  iff  $I, \theta \models P(\bar{t})$ ;

- For  $K(\varphi, t)$ , we define  $(I, \mathbb{K}), \theta \models K(\varphi, t)$  iff for each  $J \in \mathbb{K}(t)$ ,  $J, \theta \models \varphi$ ;
- The cases for  $\neg, \wedge, \vee, \forall, \exists$  are defined as usual.

For sentences, the variable assignment  $\theta$  is irrelevant and we omit it from notation.

This evaluation is a multi-modal extension of the standard Kripke evaluation, since each  $K(\cdot, t)$  is evaluated w.r.t. the set  $\mathbb{K}(t)$  of possible worlds. We now have the following obvious way of defining the semantics of a  $LTC_{K,S}$  theory  $T$ .

**Definition 15.** A knowledge structure  $\mathcal{S} = (I, \mathcal{K})$  is a weak model of  $T$  if  $(I, \mathcal{K})$  is consistent, and for each  $J \in \mathcal{K}$ , it holds that  $(J, \mathcal{S}) \models T$ .

Every possible world in  $\mathcal{K}$  represents a world such that with  $\mathcal{K}$  as initial knowledge, the theory (that is, both non-epistemic and epistemic constraints) are satisfied. Note that this definition of weak model is completely symmetric in the sense that each  $J \in \mathcal{K}$  is just as plausible for the agent, i.e., if  $(I, \mathcal{K})$  is a weak model of  $T$ , then  $(J, \mathcal{K})$  is a *weak model* for each  $J \in \mathcal{K}$ .

A limitation of weak models is that they do not restrict the agent's knowledge about the initial situation: the agent must know *at least* the theory  $T_{init}$ , but is not prevented from arbitrarily knowing more. In our running example, the theory  $T_{init}$  is empty, so the agent cannot know whether the light is initially on or off (or, to be more precise, the agent *initially* does not know this, even though it is possible that later sensing actions will reveal that the light was actually on or off all along). Our theory now indeed correctly has a weak model  $(I, \mathcal{K})$  in which  $\mathcal{K}$  is simply the set of all possible interpretations in which  $Inspect(0)$  holds. However, there is also a weak model  $(I, \mathcal{K}')$  where  $\mathcal{K}'$  contains only interpretations  $J$  in which the light is always on, even though the agent should not know this.

As a final step, our semantics will therefore select precisely those weak models in which the agent knows *only* what it should, and nothing more. The agent's initial knowledge about the initial situation should consist of all models of  $T_{init}$ . Of course, the agent initially also has knowledge about the future timepoint 1. This should consist of all that is possible given the agent's initial knowledge about timepoint 0 and the theories  $\Delta_{flu}$ ,  $T_{constr}$  and  $T_{prec}$ . And so on for the initial knowledge about the next timepoints  $i \geq 2$ . To define our semantics properly, we first define when two sets of possible worlds contain the same knowledge about the time points up to some  $i \in \mathbb{N}$ .

**Definition 16.** We write  $I \stackrel{f}{\sim}_i J$  if  $I$  and  $J$  coincide on the static predicates (i.e., for each static predicate  $P$ ,  $P^I = P^J$ ) and on the fluents for all  $t \leq i$  (i.e., for each fluent predicate  $P$  and each correctly typed tuple  $\bar{d}$  whose temporal argument  $t$  is such that  $t \leq i$ , it holds that  $\bar{d} \in P^I$  iff  $\bar{d} \in P^J$ ). Similarly, we write  $I \stackrel{a}{\sim}_i J$  if  $I$  and  $J$  coincide on the static predicates and on the actions for all  $t \leq i$ . Given two possible world sets  $\mathcal{K}_1$  and  $\mathcal{K}_2$ , we write  $\mathcal{K}_1 \sim_i \mathcal{K}_2$  iff for each  $I \in \mathcal{K}_1$  there is a  $J \in \mathcal{K}_2$  such that  $I \stackrel{f}{\sim}_i J$  and  $I \stackrel{a}{\sim}_{i-1} J$ , and vice versa.

We now define a sequence of ever smaller possible world sets, in which we gradually eliminate the weak models in which the agent has unwarranted knowledge.

**Definition 17.** We define a sequence  $(\mathcal{K}_i)_{i \geq 0}$  as follows:

- $\mathcal{K}_0 = \{I \mid \exists \mathcal{K} \text{ such that } (I, \mathcal{K}) \text{ is a weak model.}\}$
- $\mathcal{K}_{i+1} = \{I \mid \exists \mathcal{K} \sim_i \mathcal{K}_i \text{ such that } (I, \mathcal{K}) \text{ is a weak model.}\}$

The limit of this sequence can now be used for the agent's initial knowledge.

**Proposition 2.** For a theory  $T$ , the sequence  $(\mathcal{K}_i)_{i \geq 0}$  defined above is descending, and thus has a limit, which we will denote by  $\mathcal{K}_\infty$ . Given an interpretation  $I$ , if the knowledge structure  $(I, \mathcal{K}_\infty)$  is consistent, then it is a weak model.

This now leads to the following definition of our semantics.

**Definition 18.** A weak model  $(I, \mathcal{K})$  is a model of  $T$  if  $\mathcal{K}$  is the limit of the sequence  $(\mathcal{K}_i)_{i \geq 0}$  (as defined in Definition 17).

The question that remains now is whether models of a theory  $T$  indeed know nothing more about time 0 then specified by  $T_{init}$ . First let us look at the simple case where  $T_{prec}$  is empty. A weak model  $(I, \mathcal{K})$  is a model iff  $\mathcal{K} = \{J \mid J \models \Delta_{fu} \wedge T_{constr} \wedge T_{init}\}$ . It is clear that this  $\mathcal{K}$  indeed has minimal knowledge about  $t = 0$ . In its full generality,  $LTC_{K,S}$  allows some unintuitive constraints to be expressed, such as stating that an agent's current action depends on its future knowledge. It turns out that such theories can lead to models that not always have minimal knowledge. We therefore restrict attention to a more sensible fragment, in which the agent's current actions depend only on its current knowledge about current or past events.

**Definition 19.** An  $LTC_{K,S}$  theory  $T$  is called a basic action theory if all of its knowledge preconditions are of the form  $\forall \bar{x} t A(\bar{x}, t) \Rightarrow K(\varphi[\bar{x}, t'], t)$ , where  $t' \leq t$ .

It is possible to define a concept of 'minimal initial knowledge', that formalizes our informal description on the previous page. Because of space constraints we refrain from giving it here, but it turns out that with this formal definition, one can prove that basic action theories indeed have minimal initial knowledge.

## 4 An Approximative Algorithm for the Projection Problem

A basic reasoning problem in temporal systems is the *projection problem*: given an action theory that formalizes the temporal domain, determine whether a formula holds after a sequence of actions is performed. In the context of incomplete information, we typically want to know whether the agent *knows* that a certain formula holds, since this will allow us to determine whether the preconditions to its actions are satisfied as well as whether it knows that it has reached its goal. Up to timepoint  $i$ , the information that the agent has so far learned about the real world  $I$  is precisely  $Upd_I^<(i) = \bigcup_{j < i} Upd_I(j)$ . We write  $I \sim_i J$  to say that  $J$  coincides with  $I$  on these formulas  $Upd_I^<(i)$ , i.e., for each  $\varphi \in Upd_I^<(i)$ ,  $I \models \varphi$  iff  $J \models \varphi$ . The projection problem can now be defined as follows.

**Definition 20.** Given a  $LTC_{K,S}$  theory  $T$ , an interpretation  $I$ , a timepoint  $i$  and a formula  $\varphi$ , the projection problem is the problem of deciding whether for all models  $(J, \mathcal{K})$  of  $T$  such that  $J \sim_i I$ , it holds that  $(J, \mathcal{K}) \models K(\varphi, i)$ .

Typically, this problem is only of interest for interpretations  $I$  in which the agent does not perform impossible actions (i.e., there exists some  $\mathcal{K}$  such that  $(I, \mathcal{K}) \models T$ ), so we will assume this is the case. We can therefore forget about the knowledge preconditions. Once we know the information that the agent has sensed, as it is recorded in  $Upd_I^<(i)$ , also the sensing definitions become irrelevant. Our method will therefore only look at the following theory.

**Definition 21.** *For an  $\mathcal{L}_{K,S}$  theory  $T$ , an interpretation  $I$  such that  $(I, \mathcal{K}) \models T$  for some  $\mathcal{K}$ , and a timepoint  $i$ , let  $T_{proj}^{I,i}$  be the theory  $\{\Delta_{flu}, T_{init}, T_{constr}, Upd_I^<(i)\}$ .*

As the following proposition shows, we can use the set of all models of  $T_{proj}^{I,i}$  as a sound approximation of the agent's real knowledge at timepoint  $i$ .

**Proposition 3.** *Given a model  $(I, \mathcal{K})$  of a basic action theory  $T$ , let  $\mathcal{M}$  be the set of all models (in the classical FO sense) of  $T_{proj}^{I,i}$ . Then  $\mathcal{M}$  is a superset of the agent's knowledge at time  $i$  according to  $\mathbb{K} = (I, \mathcal{K})$ , i.e.,  $\mathcal{M} \supseteq \mathbb{K}(i)$ .*

The theory  $T_{proj}^{I,i}$  incorporates the knowledge that the agent has gained through his sensing actions by simply including  $Upd_I^<(i)$ . However, knowledge that the agent might gain through introspection is therefore not taken into account. In other words, this theory represents the  $\circ$  operator instead of  $\bullet$ , and this is what accounts for its possible incompleteness. As shown earlier, however, these two operators are identical for theories containing only unconditional sensing actions.

**Proposition 4.** *Given a basic action theory  $T$  that contains only unconditional sensing actions, let  $(I, \mathcal{K})$  be a model of  $T$  and  $\mathcal{M}$  the set of all models of  $T_{proj}^{I,i}$ . Then  $\mathcal{M}$  and  $\mathbb{K}(i)$  have the same knowledge about all timepoints  $j \leq i$ , i.e., for each formula  $\varphi[j]$  with  $j \leq i$ , we have:  $\forall J \in \mathcal{M}, J \models \varphi[j]$  iff  $\forall J \in \mathbb{K}(i), J \models \varphi[j]$ .*

Finding out whether a formula holds in all models of a first order theory is co-NP complete. Since applications such as planning need to solve the projection problem as an atomic step, a direct application of the above proposition would be infeasible in practice. However, it is possible to efficiently approximate the set of all models of a theory. We will use here a general approximation method for FO(ID), that efficiently allows us to detect that a formula holds in all models of a theory [17,16]. While we lack space to recall this method in detail, the essence is as follows. An FO(ID) theory  $T$  over a vocabulary  $\Sigma$  is syntactically transformed into an inductive definition  $Approx(T)$  over a new vocabulary that contains, for every predicate  $P/n \in \Sigma$ , two predicates  $P^{ct}/n$  and  $P^{cf}/n$  that represent whether  $P$  is certainly true (i.e., true in all models of  $T$ ) or certainly false (i.e., false in all models of  $T$ ). The definition  $Approx(T)$  has the property that if  $Approx(T) \models P^{ct}(\bar{d})$ , then  $T \models P(\bar{d})$  (see [16], proposition 5.4) and similar for  $P^{cf}$ . In addition,  $Approx(T)$  also defines a number of predicates  $A_\varphi^{ct}$  and  $A_\varphi^{cf}$  that tell us whether certain non-atomic formulas  $\varphi$  (or their negations, respectively) are entailed by  $T$ . In a finite domain, the model of an inductive definition can

be computed in *polynomial time*. Since we only need natural numbers up to  $i$  to solve the projection problem,  $Approx(T_{proj}^{I,i})$  therefore offers a polynomial time approximate solution to the projection problem.

*Example 3.* We have a gun that can be loaded or not, and a turkey, which can be alive or dead. Shooting kills the turkey if the gun was loaded, and also unloads the gun. Finally, the agent can inspect the turkey to see whether it is dead.

$$\Delta_{flu} = \left\{ \begin{array}{ll} Alive(t+1) & \leftarrow Alive(t) \wedge \neg(Shoot(t) \wedge Loaded(t)). \\ Alive(0) & \leftarrow Init\_Alive. \\ Loaded(t+1) & \leftarrow Loaded(t) \wedge Shoot(t). \\ Loaded(0) & \leftarrow Init\_Loaded. \end{array} \right\}$$

$$T_{sense} = \{CheckAlive(t) \text{ senses } Alive(t+1) \text{ if } t.\}$$

$$T_{init} = \{Init\_Alive.\}$$

If, after shooting a turkey that was known to be alive, the agent discovers that the turkey is dead, it should be able to conclude that the gun was previously loaded, even though it also knows that it is currently no longer loaded. We consider the real world  $I$  in which  $Shoot^I = \{0\}$ ,  $CheckAlive^I = \{1\}$ ,  $Alive^I = \{0\}$ ,  $Loaded^I = \{0\}$ ,  $Init\_Alive^I = \mathbf{t}$  and  $Init\_Loaded^I = \mathbf{t}$ . Let us define  $\psi_1 = Alive(t)$ ,  $\psi_2 = \neg Shoot(t) \vee \neg Loaded(t)$ ,  $\varphi = \psi_1 \wedge \psi_2$ . The definition  $Approx(T_{proj}^{I,3})$  now contains, among others, the following rules.

$$\left\{ \begin{array}{ll} Alive^{ct}(t+1) & \leftarrow A_{\varphi}^{ct}(t). \\ A_{\varphi}^{cf}(t) & \leftarrow Alive^{cf}(t+1). \\ A_{\psi_2}^{ct}(t) & \leftarrow Shoot^{cf}(t) \vee Loaded^{cf}(t). \\ A_{\psi_2}^{cf}(t) & \leftarrow A_{\varphi}^{cf}(t) \wedge Alive^{ct}(t). \\ Alive^{cf}(t) & \leftarrow A_{\varphi}^{cf}(t) \wedge A_{\psi_2}^{ct}(t). \\ Loaded^{ct}(t) & \leftarrow A_{\psi_2}^{cf}(t). \\ & \dots \\ Init\_Alive^{ct} & \leftarrow \\ Shoot^{ct}(t) & \leftarrow t = 0. \\ Shoot^{cf}(t) & \leftarrow t = 1 \vee t = 2. \\ Alive^{cf}(t) & \leftarrow t = 2. \end{array} \right\}$$

This definition is able to derive that, because the turkey is certainly not alive at 2, the conjunction  $\varphi(1)$  cannot hold. Since there is no  $Shoot$  at timepoint 1, the second disjunct  $\psi_2(1)$  is certainly true and so the first disjunct  $Alive(1)$  must be false. Again, this means that the disjunction  $\varphi(0)$  must be false. Now, it is the first disjunct  $Alive(0)$  that is certainly true, since  $Init\_Alive$  is true. Hence, the second disjunct  $\psi_2(1)$  cannot hold. Since  $\psi_2(1)$  is itself again a disjunction, neither of its disjuncts hold, and one of these is  $\neg Loaded(0)$ . Therefore, this approximation is indeed able to reason backwards in time to conclude that initially the gun must have been loaded. Moreover, it will also be able to conclude that, because of the  $Shoot(0)$  action,  $Loaded(1)$  and  $Loaded(2)$  no longer hold.

## 5 Discussion and Related Work

Knowledge and sensing actions were first investigated in [9], and combined with a solution for the frame problem in [12]. Since then many other approaches have been developed. A large number of these cannot handle the *postdiction* reasoning of Example 3, i.e., they cannot draw the conclusion that at time point 2 the agent knows that the gun was initially loaded, but not anymore. To handle such examples, it must be possible to trace back the current objective state of the world to the state from which it originated. This is not possible in the approaches to sensing that were developed in the context of the action language  $\mathcal{A}$  [13,15], since they only allow forward reasoning. An interesting feature of this approach, however, is that they also develop an approximation method, which transforms an  $\mathcal{A}$  specification into an answer set program. This method has a very similar flavour to ours, and also to that of [8], which defines a three-valued progression operator. Again, since only progression is considered, postdiction is not possible here. The approach of [5] uses epistemic logic and has both progression and regression operations. However, the regression operator cannot take into account current knowledge about previous points in time, and therefore this semantics is also unable to handle our Example 3.

Some approaches do have the necessary semantic machinery in place, but simply lack the syntax to refer to knowledge about past or future events. This is the case for several approaches based on situation calculus or similar formalisms [12,14,7], where the structure of the situation terms tracks the history of each state. While it would therefore be less problematic to extend these formalisms to handle postdiction, this has, as far as we know, not yet been done. The same can be said about the dynamic epistemic logic of [4].

Rather than extend an existing approach, however, we have chosen to develop a new framework. The main motivation for this was to be able to apply the approximation method of Section 3.1 to obtain an efficient but incomplete solution to the projection problem. The approaches of [1,10] try to achieve the same goal by abandoning a Kripke-style semantics in favour of a more limited concept of knowledge. This is done by introducing, for every original fluent  $F$ , a new “knowledge fluent”  $K_F$ . The way in which such a  $K_F$  changes over time is then explicitly modeled by separate update axioms. The approach of [10] is also able to handle knowledge about past or future events, and do postdiction reasoning. However, this is done in an algorithmic way, through a syntactical analysis of the specification. A disadvantage of these approaches in general is that it takes a substantial amount of work to prove a semantic connection between the knowledge fluents and the actual knowledge of that fluent [11]. The semantics of our approach allowed us to apply a general approximation method for FO, which means that soundness follows almost immediately.

Finally, [6] focuses mainly on belief change due to actions and observations where the agents may have erroneous beliefs. This is more general than our setting, since we assume that the agent’s beliefs are always correct. As we do, they also allow beliefs about previous timepoints to change as a result of later (sensing)

actions. However, their work does not focus on the knowledge representation side, assuming a given transition system instead of a logical representation.

In summary, this paper introduces a general framework for representing sensing actions, knowledge preconditions, conditional action effects, and knowledge about the past or the future. We also present a polynomial, sound but incomplete inference method for solving the projection problem, that we obtain directly by making use of a general approximation method for FO(ID). Even though this is an incomplete method, it still is capable to reach conclusions about properties that must have held at an earlier point in time, which is something that typical progression based reasoning methods cannot do.

## References

1. Demolombe, R., del Pilar Pozos Parra, M.: A Simple and Tractable Extension of Situation Calculus to Epistemic Logic. In: Ohsuga, S., Raś, Z.W. (eds.) ISMIS 2000. LNCS (LNAI), vol. 1932, pp. 515–524. Springer, Heidelberg (2000)
2. Denecker, M., Ternovska, E.: Inductive situation calculus. *Artificial Intelligence* 171(5-6), 332–360 (2007)
3. Denecker, M., Ternovska, E.: A logic of nonmonotone inductive definitions. *ACM Transactions on Computational Logic (TOCL)* 9(2), Article 14 (2008)
4. van Ditmarsch, H., van der Hoek, W., Kooi, B.: *Dynamic Epistemic Logic*, 1st edn. Springer (2007) (incorporated)
5. Herzig, A., Lang, J., Marquis, P.: Action representation and partially observable planning using epistemic logic. In: *IJCAI*, pp. 1067–1072 (2003)
6. Hunter, A., Delgrande, J.P.: Iterated belief change due to actions and observations. *J. Artif. Intell. Res (JAIR)* 40, 269–304 (2011)
7. Lakemeyer, G., Levesque, H.J.: Aol: A logic of acting, sensing, knowing, and only knowing. In: *KR*, pp. 316–329 (1998)
8. Liu, Y., Levesque, H.: Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In: *IJCAI*, pp. 522–527 (2005)
9. Moore, R.C.: A Formal Theory of Knowledge and Action. In: *Formal Theories of the Commonsense World*, pp. 319–358 (1985)
10. Petrick, R.P.A., Bacchus, F.: Extending the knowledge-based approach to planning with incomplete information and sensing. In: *ICAPS*, pp. 2–11 (2004)
11. Petrick, R.P.A., Levesque, H.J.: Knowledge equivalence in combined action theories. In: *KR*, pp. 303–314 (2002)
12. Scherl, R.B., Levesque, H.J.: Knowledge, action, and the frame problem. *Artificial Intelligence* 144, 1–39 (2003)
13. Son, T.C., Baral, C.: Formalizing sensing actions a transition function based approach. *Artif. Intell.* 125(1-2), 19–91 (2001)
14. Thielscher, M.: Representing the knowledge of a robot. In: *KR*, pp. 109–120 (2000)
15. Tu, P.H., Son, T.C., Baral, C.: Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming. *TPLP* 7(4), 377–450 (2007)
16. Vlaeminck, H., Vennekens, J., Denecker, M., Bruynooghe, M.: An approximate inference method for solving  $\exists\forall$ SO satisfiability problems. Technical Report (2012)
17. Wittocx, J., Denecker, M., Bruynooghe, M.: Constraint propagation for extended first-order logic. *CoRR*, abs/1008.2121 (2010)