

# An Epistemic Event Calculus for ASP-based Reasoning About Knowledge of the Past, Present and Future

Jiefei Ma<sup>1</sup>, Rob Miller<sup>2</sup>, Leora Morgenstern<sup>3</sup> and Theodore Patkos<sup>4</sup>

<sup>1</sup> Imperial College London, [j.ma@imperial.ac.uk](mailto:j.ma@imperial.ac.uk)

<sup>2</sup> University College London, [rsm@ucl.ac.uk](mailto:rsm@ucl.ac.uk)

<sup>3</sup> Leidos Corporation, [leora.morgenstern@leidos.com](mailto:leora.morgenstern@leidos.com)

<sup>4</sup> Foundation for Research and Technology, Hellas, [patkos@ics.forth.gr](mailto:patkos@ics.forth.gr)

## Abstract

We present a generalisation of the Event Calculus, specified in classical logic and implemented in ASP, that facilitates reasoning about non-binary-valued fluents in domains with non-deterministic, triggered, concurrent, and possibly conflicting actions. We show via a case study how this framework may be used as a basis for a “possible-worlds” style approach to epistemic and causal reasoning in a narrative setting. In this framework an agent may gain knowledge about both fluent values and action occurrences through sensing actions, lose knowledge via non-deterministic actions, and represent plans that include conditional actions whose conditions may be initially unknown.

## 1 Introduction

The Event Calculus (EC) [15, 23] is a well-established technique within AI for representing causal and narrative information about dynamic domains. However, compared to other action formalisms, little work has been done in developing epistemic extensions to the EC to facilitate reasoning about an agent’s changing state of knowledge and the state of its environment. An exception is [27], which develops an epistemic extension to the Discrete Event Calculus of [26], using a deduction-oriented rather than possible-worlds based model of knowledge. In this paper we propose an alternative epistemic EC variant, the *Epistemic Functional Event Calculus* (EFEC), that builds on a generalization of the EC of [23]. Differentiating characteristics of the EFEC are that (i) time can be either discrete or continuous (real-valued), (ii) it generalizes the EC to include non-binary (i.e. non-truth-valued) fluents, (iii) it uses a possible-worlds notion to model knowledge, following [29] and others, (iv) it facilitates reasoning about domains with concurrent, non-deterministic, and possibly conflicting actions, (v) it enables reasoning about domains with “triggered” or “natural” actions, and can model states of knowledge about action occurrences as well as fluent values, and (vi) it is able to represent knowledge states about past and future times relative to the agent’s “now” as well as the present. EFEC is implemented in Answer Set Programming (ASP), based on the translation [14] from circumscription into general stable model semantics. The ASP encoding is described in Section 3.7, and a full listing is given at <http://www.ucl.ac.uk/infostudies/efec>, along with other resources to aid the reader. Section 4 gives comparison to related work.

*Notation:* We use sorted classical predicate calculus with equality. All variables are universally quantified with maximum scope unless otherwise indicated.

## 2 Functional Event Calculus

As a foundation for our epistemic EC, we first give a generalization FEC of the non-deterministic EC of [23] to include non-binary-valued fluents. This sets the EC on a par with formalisms such as the Situation Calculus in this respect.

The FEC has a sort  $\mathcal{A}$  for *actions* (variables  $a, a', a_1, \dots$ ), a sort  $\mathcal{F}$  for *fluents* ( $f, f', f_1, \dots$ ), a sort  $\mathcal{V}$  for *values* ( $v, v', v_1, \dots$ ) and a sort  $\mathcal{T}$  for *timepoints* ( $t, t', t_1, \dots$ ). For this section only, the reader may assume that time is modeled as a total ordering (e.g.  $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{N}$  or  $\mathbb{Z}$ ). The key predicates and functions are  $Happens \subseteq \mathcal{A} \times \mathcal{T}$ ,  $ValueOf: \mathcal{F} \times \mathcal{T} \rightarrow \mathcal{V}$ ,  $CausesValue \subseteq \mathcal{A} \times \mathcal{F} \times \mathcal{V} \times \mathcal{T}$ ,  $PossVal \subseteq \mathcal{F} \times \mathcal{V}$ , and  $< \subseteq \mathcal{T} \times \mathcal{T}$ . To describe the general relationship between these predicates it is convenient to first define two auxiliary predicates,  $ValueCaused \subseteq \mathcal{F} \times \mathcal{V} \times \mathcal{T}$  and  $OtherValCausedBetween \subseteq \mathcal{F} \times \mathcal{V} \times \mathcal{T} \times \mathcal{T}$ .  $ValueCaused(F, V, T)$  means that some action happens at  $T$  that gives cause for  $F$  to take value  $V$ .  $OtherValCausedBetween(F, V, T_1, T_2)$  means that some action happens at some point in the half-open interval  $[T_1, T_2)$  that gives cause for  $F$  to take a value other than  $V$ . Note that *gives cause* is a weaker notion than the standard *causes*: non-deterministic actions do not cause specific predictable effects. E.g. (see the example at the end of this section), rolling a die *gives cause* for each number to show, but we cannot predict *which* number will show.

$$ValueCaused(f, v, t) \stackrel{\text{def}}{=} \exists a [Happens(a, t) \wedge CausesValue(a, f, v, t)]. \quad (\text{FEC1})$$

$$OtherValCausedBetween(f, v, t_1, t_2) \stackrel{\text{def}}{=} \exists t, v' [ValueCaused(f, v', t) \wedge t_1 \leq t < t_2 \wedge v \neq v']. \quad (\text{FEC2})$$

The notions of cause, effect, and inertia are captured in two FEC axioms. (FEC3) states that a fluent has a particular value at a particular time if either (i) it already had that value at an earlier time or (ii) was given cause to take that value from an earlier time, and in the meantime (including that earlier time) nothing has happened that might give cause for it to take an alternative value. Conversely, (FEC4) states that fluent  $f$  cannot have value  $v$  at time  $t_2$  if its most recent causal influences prior to  $t_2$  do not include a cause for  $v$ . Finally, (FEC5) additionally constrains each fluent's value to be at all times among the set of values defined by  $PossVal$ :

$$ValueOf(f, t_2) = v \leftarrow [(ValueOf(f, t_1) = v \vee ValueCaused(f, v, t_1)) \wedge t_1 < t_2 \wedge \neg OtherValCausedBetween(f, v, t_1, t_2)]. \quad (\text{FEC3})$$

$$ValueOf(f, t_2) \neq v \leftarrow [t_1 < t_2 \wedge OtherValCausedBetween(f, v, t_1, t_2) \wedge \neg \exists t (t_1 \leq t < t_2 \wedge ValueCaused(f, v, t))]. \quad (\text{FEC4})$$

$$ValueOf(f, t) = v \rightarrow PossVal(f, v). \quad (\text{FEC5})$$

Definitions of the predicates  $Happens$ ,  $CausesValue$  and  $PossVal$  are given in the domain-dependent part of the theory (or partially defined and then minimised to address the Frame Problem and related issues). The reader may note that if  $PossVal(f, v) \equiv [v = True \vee v = False]$  for all fluents, axioms (FEC1)-(FEC5) are effectively equivalent to axioms (EC1)-(EC6) from [23]. Translating (FEC1)-(FEC5) to ASP is straightforward following [14] (see Section 3.7).

The following domain example of rolling a die illustrates that FEC can capture non-determinism by expressing with  $CausesValue$  that an action can give cause for more than one value for a given fluent. The effect of rolling a 6-sided die twice (at times 10 and 20) after it initially shows "2" is described as follows:

$$f = DieFaceShowing \wedge a = RollDie. \quad (\text{D1})$$

$$v = 1 \vee v = 2 \vee v = 3 \vee v = 4 \vee v = 5 \vee v = 6. \quad (\text{D2})$$

$$PossVal(DieFaceShowing, v). \quad (\text{D3})$$

$$CausesValue(RollDie, DieFaceShowing, v, t). \quad (\text{D4})$$

$$ValueOf(DieFaceShowing, 0) = 2. \quad (\text{D5})$$

$$\text{Happens}(\text{RollDie}, t) \equiv [t=10 \vee t=20]. \quad (\text{D6})$$

(D1) and (D2) are domain closure axioms. (D4) expresses that rolling a die gives cause for each of its six faces to show uppermost. (In a more complex domain (D4) might be written as the circumscription of a set of *CausesValue* clauses.) (D5) and (D6) express *narrative information*. Again, (D6) could alternatively be expressed as the circumscription of the clauses  $\text{Happens}(\text{RollDie}, 10)$  and  $\text{Happens}(\text{RollDie}, 20)$ . If time is represented by  $\mathbb{R}_{\geq 0}$  or  $\mathbb{N}$ , this example gives rise to  $6^2$  models. In each model the die has face “2” showing throughout the interval  $[0, 10]$ , and an arbitrary one of its six faces showing throughout each of the intervals  $(10, 20]$  and  $(20, +\infty)$ .

### 3 Epistemic Reasoning

**A Motivating Example – the Shopping Outlet:** *To obtain an item from a catalogue shopping outlet, a customer first purchases the item using a terminal. This causes the customer to be (non-deterministically) assigned to one of three collection points color-coded “red”, “blue” and “green”. The customer can ascertain which collection point she has been assigned to from a display, and collect the item accordingly. To discourage unnecessary queuing, an attempt to collect the item from the wrong collection point cancels the purchase. Additionally, customers assigned to “red” receive a free gift on successful item collection.* This scenario can be regarded as having an action (purchasing the item) for which one effect is non-deterministic, a “sensing” or knowledge-producing action (ascertaining the collection point), conditional actions (collect from a particular collection point *if* assigned there), and a conditionally “triggered” action (giving a gift at the red collection point). The example is pertinent because it is impossible for the customer to obtain the item for certain without the sensing action (assuming she cannot collect from all collection points simultaneously).

Other than “epistemic fluents” (described later) our representation of this example uses the three fluents: *CollectionPoint* (possible values  $\{\text{Red}, \text{Blue}, \text{Green}\}$ ), *Bought* and *Collected* (truth-valued). Other than sensing actions (described later) we use actions *Purchase*, *GiveFreeGift*, *CollectFromRed*, *CollectFromBlue* and *CollectFromGreen*. *GiveFreeGift* is a “triggered” action performed by the environment automatically in the circumstances described. The UNA axioms for values and definition of *PossVal* for the non-epistemic fluents are:

$$\text{PossVal}(\text{Bought}, v) \equiv [v = \text{True} \vee v = \text{False}]. \quad (\text{S1})$$

$$\text{PossVal}(\text{CollectionPoint}, v) \equiv [v = \text{Red} \vee v = \text{Blue} \vee v = \text{Green}]. \quad (\text{S2})$$

$$\text{PossVal}(\text{Collected}, v) \equiv [v = \text{True} \vee v = \text{False}]. \quad (\text{S3})$$

$$\text{Red} \neq \text{Blue} \neq \text{Green} \neq \text{True} \neq \text{False}. \quad (\text{S4})$$

To fully represent this domain, we need a time structure, predicates and axioms to facilitate reasoning about knowledge.

#### 3.1 EFEC Knowledge Axioms

Our approach to epistemic reasoning is inspired by the “possible-worlds” approaches of [29, 24, 10] and others, but has some ontological differences necessitated by the fact that the EC is a narrative framework, and typically uses linear (not branching) time. Intuitively, each model of an *epistemic* FEC (EFEC) theory contains a number of parallel “possible time lines” and a notion of accessibility (modeled via “epistemic fluents”) between these. The time lines accessible to an agent at any given moment are those that contain a narrative (fluent values and action occurrences) compatible with the agent’s current state of knowledge. In other

words, at any instant the agent “knows” whatever holds in every narrative accessible at that instant. Sensing actions terminate the accessibility of time lines whose narratives do not match the sensed information. Once terminated, there is no mechanism for re-initiating accessibility. Knowledge loss due to non-deterministic action occurrences is instead captured by ensuring (in each model) that, for every possible history of previous fluent values and action occurrences up to the given point of non-determinism, there is a sufficient number of identical, duplicate accessible narratives to ensure that at least one such narrative is available to be extended to reflect each possible non-deterministic outcome.

To represent time as a system of parallel time lines, we add two new sorts to the logic, a sort  $\mathcal{W}$  for *worlds*, to be understood as identifiers for possible time lines (variables  $w, w', w_1, \dots$ ), and a sort  $\mathcal{I}$  for *instants* (variables  $i, i', i_1, \dots$ ). The constant  $W_a$  of sort  $\mathcal{W}$  signifies the “actual world”, and  $\prec \subseteq \mathcal{I} \times \mathcal{I}$  is a partial (possibly total) ordering over  $\mathcal{I}$ . The function  $\langle \cdot \rangle: \mathcal{W} \times \mathcal{I} \rightarrow \mathcal{T}$  maps world/instant pairs to time points, so that time point  $\langle W, I \rangle$  represents “instant  $I$  in possible world  $W$ ”.  $\langle I \rangle$  is shorthand for  $\langle W_a, I \rangle$ . Axioms for the time structure are:

$$\forall t \exists w, i. (t = \langle w, i \rangle). \quad (\text{EFEC1})$$

$$\langle w, i \rangle = \langle w', i' \rangle \equiv (w = w' \wedge i = i'). \quad (\text{EFEC2})$$

$$\langle w, i \rangle < \langle w', i' \rangle \equiv (w = w' \wedge i < i'). \quad (\text{EFEC3})$$

$\leq, >$  and  $\geq$  have their usual meanings in terms of  $=$  and  $<$ . For ease of exposition, we assume here that  $\mathcal{I}$  is interpreted as  $\mathbb{R}_{\geq 0}$  or  $\mathbb{N}$  with usual ordering  $\prec$ .

To specify the dynamic accessibility relation between possible worlds, we adapt Scherl and Levesque’s notion of *epistemic fluents*, introducing a function  $K: \mathcal{W} \rightarrow \mathcal{F}$ . The epistemic fluent  $K(W)$  represents the “accessibility” property of  $W$ , so that  $\text{ValueOf}(K(W), \langle W', I \rangle) = \text{true}$  means that “ $W$  is accessible from  $W'$  at instant  $I$ ”.  $K$  is a truth-valued fluent, and, since we are modeling knowledge, the relationship it represents between worlds is reflexive, symmetric and transitive (i.e. an equivalence relation):

$$\text{PossVal}(K(w), v) \equiv [v = \text{True} \vee v = \text{False}]. \quad (\text{EFEC4})$$

$$\text{ValueOf}(K(w), \langle w, i \rangle) = \text{True}. \quad (\text{EFEC5})$$

$$\text{ValueOf}(K(w), \langle w', i \rangle) = \text{True} \equiv \text{ValueOf}(K(w'), \langle w, i \rangle) = \text{True}. \quad (\text{EFEC6})$$

$$\text{ValueOf}(K(w_3), \langle w_1, i \rangle) = \text{True} \leftarrow [\text{ValueOf}(K(w_2), \langle w_1, i \rangle) = \text{True} \wedge \text{ValueOf}(K(w_3), \langle w_2, i \rangle) = \text{True}]. \quad (\text{EFEC7})$$

For simplicity, we assume that each fluent  $F$  in the domain can be sensed via an action  $\text{Sense}(F)$  (although nothing in our approach dictates that this must be so<sup>1</sup>). Sensing  $F$  terminates the accessibility of possible worlds in which the value of  $F$  is different from that in the agent’s world:

$$\text{CausesValue}(\text{Sense}(f), K(w'), \text{False}, \langle w, i \rangle) \quad (\text{EFEC8})$$

$$\leftarrow \text{ValueOf}(f, \langle w, i \rangle) \neq \text{ValueOf}(f, \langle w', i \rangle).$$

EFEC includes six knowledge predicates regarding actions and non-epistemic fluents, whose arity and argument sorts can be seen from the following list of intended meanings:

- $\text{KnowsValueIsNot}(\langle W, I \rangle, F, I', V)$  - When in world  $W$  at instant  $I$ , the agent knows that the value of fluent  $F$  at instant  $I'$  was not / is not / will not be  $V$ .
- $\text{KnowsValueIs}(\langle W, I \rangle, F, I', V)$  - When in  $W$  at instant  $I$ , the agent knows that the value of  $F$  at  $I'$  was / is / will be  $V$ .

<sup>1</sup>We could for example specify when an action  $a$  senses a fluent  $f$  using a predicate  $\text{Senses}$  along with a general rule such as  $\text{CausesValue}(a, K(w'), \text{False}, \langle w, i \rangle) \leftarrow [\text{Senses}(a, f, \langle w, i \rangle) \wedge \text{ValueOf}(f, \langle w, i \rangle) \neq \text{ValueOf}(f, \langle w', i \rangle)]$ .

- $KnowsValue(\langle W, I \rangle, F, I')$  - When in  $W$  at instant  $I$ , the agent knows the value of  $F$  at instant  $I'$ .
- $KnowsHappens(\langle W, I \rangle, A, I')$  - When in  $W$  at instant  $I$ , the agent knows that action  $A$  has happened / happens / will happen at instant  $I'$ .
- $KnowsNotHappens(\langle W, I \rangle, A, I')$  - When in  $W$  at instant  $I$ , the agent knows that  $A$  has not / does not / will not happen at  $I'$ .
- $KnowsIfHappens(\langle W, I \rangle, A, I')$  - When in  $W$  at instant  $I$ , the agent knows whether or not  $A$  has happened / happens / will happen at  $I'$ .

The corresponding definitional axioms are:<sup>2</sup>

$$KnowsValueIsNot(\langle w, i \rangle, f, i', v) \equiv \forall w' (f \neq K(w') \wedge \quad \text{(EFEC9)}$$

$$[ValueOf(K(w'), \langle w, i \rangle) = True \rightarrow ValueOf(f, \langle w', i' \rangle) \neq v]).$$

$$KnowsValueIs(\langle w, i \rangle, f, i', v) \equiv \quad \text{(EFEC10)}$$

$$\forall v' [(PossVal(f, v') \wedge v' \neq v) \rightarrow KnowsValueIsNot(\langle w, i \rangle, f, i', v')].$$

$$KnowsValue(t, f, i') \equiv \exists v. KnowsValueIs(t, f, i', v). \quad \text{(EFEC11)}$$

$$KnowsHappens(\langle w, i \rangle, a, i') \equiv \quad \text{(EFEC12)}$$

$$\forall w' [ValueOf(K(w'), \langle w, i \rangle) = True \rightarrow Happens(a, \langle w', i' \rangle)].$$

$$KnowsNotHappens(\langle w, i \rangle, a, i') \equiv \quad \text{(EFEC13)}$$

$$\forall w' [ValueOf(K(w'), \langle w, i \rangle) = True \rightarrow \neg Happens(a, \langle w', i' \rangle)].$$

$$KnowsIfHappens(t, a, i') \equiv \quad \text{(EFEC14)}$$

$$[KnowsHappens(t, a, i') \vee KnowsNotHappens(t, a, i')].$$

### 3.2 Representing Action Occurrences

To represent the class of domains exemplified by the shopping outlet scenario, we need to be able to represent three kinds of action occurrence. First, the agent may perform an action unconditionally (e.g. purchase an item). Second, the agent may perform an action if (and only if) it knows a particular condition holds (e.g. collect from the red collection point if assigned there). Third, an action might be automatically triggered in the environment (e.g. the giving of a free gift on item collection). Accordingly, we introduce three occurrence predicates,  $Perform \subseteq \mathcal{A} \times \mathcal{I}$ ,  $PerformIfValueKnownIs \subseteq \mathcal{A} \times \mathcal{F} \times \mathcal{V} \times \mathcal{I}$  and  $Triggered \subseteq \mathcal{A} \times \mathcal{T}$  and relate them to  $Happens$  as follows.

$$Perform(a, i) \rightarrow Happens(a, \langle w, i \rangle). \quad \text{(EFEC15)}$$

$$PerformIfValueKnownIs(a, f, v, i) \rightarrow \quad \text{(EFEC16)}$$

$$[Happens(a, \langle w, i \rangle) \equiv KnowsValueIs(\langle w, i \rangle, f, i, v)].$$

$$Triggered(a, t) \rightarrow Happens(a, t). \quad \text{(EFEC17)}$$

Definitions of these predicates are given in the domain-dependent theory, or, in the case of the “ $Perform\dots$ ” predicates, generated via a planning process. The “ $Perform\dots$ ” predicates have last argument of sort  $\mathcal{I}$  (rather than  $\mathcal{T}$ ) because the occurrences they refer to are by assumption under the control of the agent. In contrast the conditions under which triggered actions occur may or may not be known at particular times.

Rather than minimising  $Happens$  directly, we state that all occurrences of actions (at any instant in any possible world) are accounted for by “ $Perform\dots$ ” or  $Triggered$ :

$$Happens(a, \langle w, i \rangle) \rightarrow [\exists f, v. PerformIfValueKnownIs(a, f, v, i) \quad \text{(EFEC18)}$$

$$\vee Perform(a, i) \vee Triggered(a, \langle w, i \rangle)].$$

<sup>2</sup>KnowValues (KVI) is defined here in terms of the arguably more basic KnowValueIsNot (KVIN). KVIN is particularly useful for expressing complete ignorance (i.e., for all possible values  $v$  of a fluent, one does not know that the fluent does not have value  $v$ ). But note that KVI could instead be defined directly in terms of possible worlds; in that case, the current formulation of (EFEC10) could be derived as a corollary of (EFEC9) and the direct definition of KVI.

### 3.3 An Axiomatization of the Shopping Example

We next give the domain-dependent shopping axioms. Their ASP translations are at <http://www.ucl.ac.uk/infostudies/efec>. In addition to (S1)–(S4) we have domain closure and uniqueness-of-names axioms for actions and fluents:

$$f = \text{CollectionPoint} \vee f = \text{Bought} \vee f = \text{Collected} \vee \exists w. f = K(w). \quad (\text{S5})$$

$$\text{CollectionPoint} \neq \text{Bought} \neq \text{Collected} \neq K(w) \wedge [K(w) = K(w') \rightarrow w = w']. \quad (\text{S6})$$

$$a = \text{Purchase} \vee a = \text{CollectFromRed} \vee a = \text{CollectFromBlue} \quad (\text{S7})$$

$$\vee a = \text{CollectFromGreen} \vee a = \text{GiveFreeGift} \vee \exists f. a = \text{Sense}(f).$$

$$\text{Purchase} \neq \text{GiveFreeGift} \neq \text{CollectFromRed} \neq \text{CollectFromBlue} \neq \quad (\text{S8})$$

$$\text{CollectFromGreen} \neq \text{Sense}(f) \wedge [\text{Sense}(f) = \text{Sense}(f') \rightarrow f = f'].$$

We assume sort  $\mathcal{I}$  is interpreted as  $\mathbb{R}_{\geq 0}$ . We express knowledge about initial instant 0 in terms of *KnowsValueIsNot* (which gives more expressivity than *KnowsValueIs*), and completely specify *KnowsValueIsNot* at instant 0 (recall from Section 3.1 that  $\langle 0 \rangle$  is shorthand for  $\langle W_a, 0 \rangle$ ):

$$\text{KnowsValueIsNot}(\langle 0 \rangle, f, 0, v) \equiv [(f = \text{Bought} \vee f = \text{Collected}) \wedge v = \text{True}]. \quad (\text{S9})$$

Causal domain information is captured by a collection of clauses such as

$$\begin{aligned} \text{CausesValue}(\text{CollectFromRed}, \text{Collected}, \text{True}, t) \leftarrow \\ [\text{ValueOf}(\text{CollectionPoint}, t) = \text{Red} \wedge \text{ValueOf}(\text{Bought}, t) = \text{True}]. \end{aligned}$$

and the frame problem is addressed by circumscribing this collection together with (EFEC8) to give

$$\text{CausesValue}(a, f, v, t) \equiv \quad (\text{S10})$$

$$\begin{aligned} & [(a = \text{Purchase} \wedge f = \text{CollectionPoint} \wedge \text{PossVal}(\text{CollectionPoint}, v)) \\ & \vee (a = \text{Purchase} \wedge f = \text{Bought} \wedge v = \text{True}) \\ & \vee (a = \text{CollectFromRed} \wedge f = \text{Collected} \wedge v = \text{True} \\ & \quad \wedge \text{ValueOf}(\text{CollectionPoint}, t) = \text{Red} \wedge \text{ValueOf}(\text{Bought}, t) = \text{True}) \\ & \vee (a = \text{CollectFromRed} \wedge f = \text{Bought} \wedge v = \text{False} \wedge \text{ValueOf}(\text{CollectionPoint}, t) \neq \text{Red}) \\ & \vee (a = \text{CollectFromBlue} \wedge f = \text{Collected} \wedge v = \text{True} \\ & \quad \wedge \text{ValueOf}(\text{CollectionPoint}, t) = \text{Blue} \wedge \text{ValueOf}(\text{Bought}, t) = \text{True}) \\ & \vee (a = \text{CollectFromBlue} \wedge f = \text{Bought} \wedge v = \text{False} \wedge \text{ValueOf}(\text{CollectionPoint}, t) \neq \text{Blue}) \\ & \vee (a = \text{CollectFromGreen} \wedge f = \text{Collected} \wedge v = \text{True} \\ & \quad \wedge \text{ValueOf}(\text{CollectionPoint}, t) = \text{Green} \wedge \text{ValueOf}(\text{Bought}, t) = \text{True}) \\ & \vee (a = \text{CollectFromGreen} \wedge f = \text{Bought} \wedge v = \text{False} \wedge \text{ValueOf}(\text{CollectionPoint}, t) \neq \text{Green}) \\ & \vee \exists f', w, w', i. (a = \text{Sense}(f') \wedge f = K(w') \wedge v = \text{False} \wedge t = \langle w, i \rangle \wedge \\ & \quad \text{ValueOf}(f', \langle w, i \rangle) \neq \text{ValueOf}(f', \langle w', i \rangle))]. \end{aligned}$$

Various EC mechanisms compatible with the framework described here have been developed to represent triggered actions (see e.g. [23]). For triggering a free gift in the shopping example, the following is sufficient:

$$\begin{aligned} \text{Triggered}(a, t) \equiv [\text{ValueOf}(\text{Bought}, t) = \text{True} \wedge \text{Happens}(\text{CollectFromRed}, t) \\ \wedge a = \text{GiveFreeGift} \wedge \text{ValueOf}(\text{CollectionPoint}, t) = \text{Red}]. \end{aligned} \quad (\text{S11})$$

Finally, actions the agent has done or (conditionally) intends to do can be represented by definitions of *Perform* and *PerformIfValueKnownIs*. For example, a plan for obtaining the item by instant 4 might be described by:

$$\text{Perform}(a, i) \equiv [(a = \text{Purchase} \wedge i = 1) \vee (a = \text{Sense}(\text{CollectionPoint}) \wedge i = 2)] \quad (\text{S12})$$

$$\text{PerformIfValueKnownIs}(a, f, v, i) \equiv \quad (\text{S13})$$

$$[(a = \text{CollectFromRed} \wedge f = \text{CollectionPoint} \wedge v = \text{Red} \wedge i = 3)$$

$$\begin{aligned} & \vee(a = \text{CollectFromBlue} \wedge f = \text{CollectionPoint} \wedge v = \text{Blue} \wedge i = 3) \\ & \vee(a = \text{CollectFromGreen} \wedge f = \text{CollectionPoint} \wedge v = \text{Green} \wedge i = 3)]. \end{aligned}$$

### 3.4 Existence of Possible Worlds

The domain-independent EFEC axiomatization is not yet complete. We need to ensure there is a sufficiently large collection of accessible possible worlds in each model to adequately represent lack of knowledge, both about what holds at the initial instant and about what holds after the occurrence of a set of simultaneous non-deterministic actions.

For example, if we have no knowledge about the initial values of  $N$  truth-valued fluents, our axiomatization should ensure there are at least  $2^N$  initially accessible worlds, one for each possible  $N$ -combination of truth values. If we were to allow models with less than  $2^N$  such worlds, then in these models some sensing actions would (by terminating accessibility – see (EFEC8)) give unwarranted knowledge about the values of fluents other than that being sensed.

To eliminate such models in the general case we first axiomatize the notion that two worlds differ at instant 0 in respect of non-epistemic fluents by at most one such fluent, by defining a predicate *InitiallyDifferAtMostBy*  $\subseteq \mathcal{W} \times \mathcal{W} \times \mathcal{F}$ :

$$\begin{aligned} \text{InitiallyDifferAtMostBy}(w_1, w_2, f) & \equiv \forall f' [(f' \neq f \wedge \neg \exists w'. f' = K(w')) \\ & \rightarrow \text{ValueOf}(f', \langle w_1, 0 \rangle) = \text{ValueOf}(f', \langle w_2, 0 \rangle)]. \end{aligned} \quad (\text{EFEC19})$$

Domain descriptions will typically include at least a partial specification for *KnowsValueIsNot* at instant 0 (see e.g. (S9)). Axiom (EFEC20) states that for every value  $v$  not known not to be the initial value of some non-epistemic fluent  $f$ , and for every initially accessible world  $w$ , we can find an initially accessible world  $w'$  in which  $f$  has initial value  $v$  and which is initially identical to  $w$  in all other respects:

$$\begin{aligned} [\neg \exists w'. f = K(w') \wedge \text{PossVal}(f, v) \wedge \neg \text{KnowsValueIsNot}(\langle 0 \rangle, f, 0, v) \wedge \\ \text{ValueOf}(K(w), \langle 0 \rangle) = \text{True}] \rightarrow \exists w' [\text{ValueOf}(K(w'), \langle 0 \rangle) = \text{True} \wedge \\ \text{ValueOf}(f, \langle w', 0 \rangle) = v \wedge \text{InitiallyDifferAtMostBy}(w, w', f)]. \end{aligned} \quad (\text{EFEC20})$$

For domains with a finite number of fluents, it is possible to show that any combination of fluent values, each of which are not known not to hold, corresponds to an initially accessible world, by repeated application of (EFEC20).

To guarantee the existence of a sufficient number of worlds accessible after the occurrence of a set of simultaneous non-deterministic actions, we first need to be able to identify time periods “immediately after” such occurrences. To do this we include the function *Next* :  $\mathcal{T} \rightarrow \mathcal{T}$  from [22]. For time point  $T$  axioms (EFEC21)–(EFEC23) constrain *Next*( $T$ ) as follows. If  $T$  is before the last action occurrence in  $T$ ’s timeline, then *Next*( $T$ ) is the point of the next action occurrence (or simultaneous occurrences) after  $T$ . Otherwise, *Next*( $T$ ) is any arbitrary time point after  $T$ .

$$t < \text{Next}(t). \quad (\text{EFEC21})$$

$$[t < t_1 \wedge t_1 < \text{Next}(t)] \rightarrow \neg \text{Happens}(a, t_1). \quad (\text{EFEC22})$$

$$[\text{Happens}(a_1, t_1) \wedge t < t_1] \rightarrow \exists a. \text{Happens}(a, \text{Next}(t)) \quad (\text{EFEC23})$$

(FEC3) therefore guarantees that for any  $T$ , values of fluents remain unchanged in the half-open interval  $(T, \text{Next}(T)]$ . In particular, if actions occur at  $T$  then the immediate effects of those actions remain apparent throughout  $(T, \text{Next}(T)]$ .

Two other predicates are needed, *DifferAfterAtMostBy*  $\subseteq \mathcal{W} \times \mathcal{W} \times \mathcal{I} \times \mathcal{F}$  and *EqualUpTo*  $\subseteq \mathcal{W} \times \mathcal{W} \times \mathcal{I}$ . *DifferAfterAtMostBy*( $W_1, W_2, I, F$ ) means that in the periods immediately following instant  $I$  on each of the timelines  $W_1$  and  $W_2$  – i.e. in the half open intervals

$(\langle W_1, I \rangle, \text{Next}(\langle W_1, I \rangle))$  and  $(\langle W_2, I \rangle, \text{Next}(\langle W_2, I \rangle))$  – all non-epistemic fluents except possibly  $F$  take the same value. The meaning of  $\text{EqualUpTo}(W_1, W_2, I)$  is that in the period from 0 up to and including  $I$  on each of the timelines  $W_1$  and  $W_2$  – i.e. in the intervals  $[\langle W_1, 0 \rangle, \langle W_1, I \rangle]$  and  $[\langle W_2, 0 \rangle, \langle W_2, I \rangle]$  – all non-epistemic fluents take the same value:

$$\begin{aligned} \text{DifferAfterAtMostBy}(w_1, w_2, i, f) &\equiv \forall f' [(f' \neq f \wedge \neg \exists w'. f' = K(w')) \\ &\rightarrow \text{ValueOf}(f', \text{Next}(\langle w_1, i \rangle)) = \text{ValueOf}(f', \text{Next}(\langle w_2, i \rangle))]. \end{aligned} \quad (\text{EFEC24})$$

$$\begin{aligned} \text{EqualUpTo}(w_1, w_2, i) &\equiv \\ \forall f, i' [(i' \preceq i \wedge \neg \exists w'. f = K(w')) &\rightarrow \text{ValueOf}(f, \langle w_1, i' \rangle) = \text{ValueOf}(f, \langle w_2, i' \rangle)]. \end{aligned} \quad (\text{EFEC25})$$

Axiom (EFEC26) is the counterpart of (EFEC20) for periods immediately following (possibly non-deterministic) action occurrences. It states that if a non-epistemic fluent  $f$  is given cause to have value  $v$  at instant  $i$  in the accessible world  $w$ , then there exists another accessible world  $w'$  identical to  $w$  up to  $\text{Next}(\langle w, i \rangle)$  except that at  $\text{Next}(\langle w', i \rangle)$   $f$  has value  $v$  (whereas  $f$  may or may not have value  $v$  at  $\text{Next}(\langle w, i \rangle)$ ):

$$\begin{aligned} [\text{ValueCaused}(f, v, \langle w, i \rangle) \wedge \neg \exists w_1. f = K(w_1) \\ \wedge \text{ValueOf}(K(w), \text{Next}(\langle i \rangle)) = \text{True}] \\ \rightarrow \exists w' [\text{ValueOf}(K(w'), \text{Next}(\langle i \rangle)) = \text{True} \wedge \text{ValueOf}(f, \text{Next}(\langle w', i \rangle)) = v \\ \wedge \text{EqualUpTo}(w, w', i) \wedge \text{DifferAfterAtMostBy}(w, w', i, f)]. \end{aligned} \quad (\text{EFEC26})$$

Note that these axioms rest on the assumption that all pairs of fluents are orthogonal (i.e., there are no state constraints).

### 3.5 Example Inferences

EFEC allows us to describe *epistemically feasible* plans. For example, the goal of obtaining an item at instant 4 might be expressed as  $\text{Goal}_S \equiv \text{KnowsValueIs}(\langle 0 \rangle, \text{Collected}, 4, \text{True})$  and a plan to achieve this as  $\text{Plan}_S \equiv [(\text{S12}) \wedge (\text{S13})]$ . Taking an abductive view of planning, Proposition 1 below shows that:  $(\text{FEC1}) \wedge \dots \wedge (\text{FEC5}) \wedge (\text{EFEC1}) \wedge \dots \wedge (\text{EFEC26}) \wedge (\text{S1}) \wedge \dots \wedge (\text{S11}) \wedge \text{Plan}_S \models \text{Goal}_S$ , or equivalently (under a deductive view of planning)  $(\text{FEC1}) \wedge \dots \wedge (\text{FEC5}) \wedge (\text{EFEC1}) \wedge \dots \wedge (\text{EFEC26}) \wedge (\text{S1}) \wedge \dots \wedge (\text{S11}) \models \text{Plan}_S \rightarrow \text{Goal}_S$ . Proposition 1 also shows that the agent is able to infer and preserve knowledge about instances that have passed, both regarding action occurrences (in particular triggered occurrences, e.g.  $\text{KnowsIfHappens}(\langle 5 \rangle, \text{GiveFreeGift}, 3)$ ) and fluent values (e.g.  $\text{KnowsValue}(\langle 3 \rangle, \text{CollectionPoint}, 2)$ ). Consequently, the agent can plan to discover currently unknown facts about past times.

**Proposition 1.** *The theory  $(\text{FEC1}) \wedge \dots \wedge (\text{FEC5}) \wedge (\text{EFEC1}) \wedge \dots \wedge (\text{EFEC26}) \wedge (\text{S1}) \wedge \dots \wedge (\text{S13})$  entails the sentences:*

$$\begin{aligned} &\text{KnowsValueIs}(\langle 0 \rangle, \text{Collected}, 4, \text{True}). \\ &\neg \text{KnowsValue}(\langle 2 \rangle, \text{CollectionPoint}, 2). \\ &\text{KnowsValue}(\langle 3 \rangle, \text{CollectionPoint}, 2). \\ &\text{KnowsValue}(\langle 3 \rangle, \text{CollectionPoint}, 3). \\ &\text{KnowsHappens}(\langle 5 \rangle, \text{Purchase}, 1). \\ &\text{KnowsIfHappens}(\langle 5 \rangle, \text{GiveFreeGift}, 3). \end{aligned}$$

### 3.6 Another Brief Example Showing the Sensing of Evidence About The Past

*A disease triggers the production of antibodies that cure the disease and remain in the bloodstream afterwards. A person knows she did not have the antibodies at time 0 but wonders if*



she had just contracted the disease at that time. At time 1 she can ascertain this by testing (sensing) for the antibodies. Assuming  $\mathcal{I} = \mathbb{N}$ , fluents are truth-valued, and domain closure and uniqueness-of-names axioms  $DC_A$  and  $UNA_A$  (analogous to (S1)-(S8)) our representation is:

$$\text{KnowsValueIsNot}(\langle 0 \rangle, f, 0, v) \equiv [f = \text{Antibodies} \wedge v = \text{True}]. \quad (\text{A1})$$

$$\text{CausesValue}(a, f, v, t) \equiv \quad (\text{A2})$$

$$\begin{aligned} & [(a = \text{MakeAntibodies} \wedge f = \text{Antibodies} \wedge v = \text{True}) \\ & \vee (a = \text{MakeAntibodies} \wedge f = \text{Disease} \wedge v = \text{False}) \\ & \vee \exists f', w, w', i (a = \text{Sense}(f') \wedge f = K(w') \wedge v = \text{False} \wedge t = \langle w, i \rangle \wedge \\ & \quad \text{ValueOf}(f', \langle w, i \rangle) \neq \text{ValueOf}(f', \langle w', i \rangle))]. \end{aligned}$$

$$\text{Triggered}(a, t) \equiv [a = \text{MakeAntibodies} \wedge \text{ValueOf}(\text{Disease}, t) = \text{True}]. \quad (\text{A3})$$

$$\text{Perform}(a, i) \equiv [a = \text{Sense}(\text{Antibodies}) \wedge i = 1]. \quad (\text{A4})$$

$$\neg \text{PerformIfValueKnownIs}(a, f, v, i). \quad (\text{A5})$$

**Proposition 2.** *The theory  $(\text{FEC1}) \wedge \dots \wedge (\text{FEC5}) \wedge (\text{EFEC1}) \wedge \dots \wedge (\text{EFEC26}) \wedge DC_A \wedge UNA_A \wedge (\text{A1}) \wedge \dots \wedge (\text{A5})$  entails the following sentences:*

$$\begin{aligned} & \neg \text{KnowsValue}(\langle 0 \rangle, \text{Disease}, 0). \\ & \text{KnowsValue}(\langle 2 \rangle, \text{Disease}, 0). \\ & \neg \text{KnowsIfHappens}(\langle 0 \rangle, \text{MakeAntibodies}, 0). \\ & \text{KnowsIfHappens}(\langle 2 \rangle, \text{MakeAntibodies}, 0). \end{aligned}$$

### 3.7 ASP Implementation

For  $\mathcal{I} \subset \mathbb{N}$  we have encoded the FEC/EFEC domain independent axioms in ASP, along with the shopping and antibodies examples. Coding (see <http://www.ucl.ac.uk/infostudies/efec>) is based on the reformulation of circumscription into stable model semantics given by Kim et al. [14]. The ASP solver `Clingo` [12] is used for computation.

For example, (FEC1) can be first rewritten into

$$\text{valueCaused}(F, V, T) \text{ :- happens}(A, T), \text{causesValue}(A, F, V, T).$$

and then a *choice formula* [14] (e.g.,  $\forall X (p(X) \vee \neg p(X))$ ) is added (as a rule, e.g.,  $\forall X (\neg p(X) \leftarrow \text{not } p(X))$ ):

$$\neg \text{valueCaused}(F, V, T) \text{ :- possVal}(F, V), \text{time}(T), \text{not causedValue}(F, V, T).$$

where “ $\neg$ ” and “**not**” are the classical negation and negation as failure in ASP, respectively.  $\text{possVal}(F, V)$  are given as facts (e.g., `possVal(bought, true)`, `possVal(bought, false)`) based on (S1)–(S3), and  $\text{time}(T)$  represents the time sort (described shortly). These two predicates are added to the body of the rule to ensure the rule is *safe* – all variables appearing in a rule must appear in a domain predicate – a condition for grounding.

In the translation of EFEC into ASP, we need to guarantee that the sort  $\mathcal{T}$  of timepoints is finite. This is achieved by using positive integers for the sort  $\mathcal{W}$  of worlds and the sort  $\mathcal{I}$  of instants, with two constants `maxworld` and `maxinstant` whose meanings are indicated by their names<sup>3</sup>. The ASP sort definitions are:

$$\begin{aligned} \text{time}((W, I)) & \text{ :- world}(W), \text{instant}(I). \\ \text{world}(1.. \text{maxworld}). \\ \text{instant}(0.. \text{maxinstant}). \end{aligned}$$

<sup>3</sup>Although classical EFEC allows infinite worlds, if the action, fluent, value and instant sorts are all finite, then the set of non-isomorphic worlds is also finite.

The value of `maxinstant` depends on the queries, i.e. how far ahead the agent needs to reason about. But finding the right (smallest but sufficient) value for `maxworld` is not obvious given an EFEC specification and query. On the one hand we must ensure that there are enough possible worlds, but on the other hand we do not want duplicate possible worlds that do not affect the answers but may increase the computational complexity. The former requirement is guaranteed by (EFEC20) and (EFEC26), and the latter can be achieved by adding an extra axiom (an integrity constraint in ASP) to disallow duplicate worlds:

$$:- \text{world}(W1), \text{world}(W2), W1 \neq W2, \text{equalUpTo}(W1, W2, \text{maxinstant}).$$

Thus in the implementation, we can incrementally increase `maxworld` until `Clingo` reports “satisfiable”. Note that although the value of `maxworld` found in this way is unique to a given EFEC specification, `Clingo` may return multiple stable models because of permutations<sup>4</sup> of possible world labeling (e.g., we can swap the names of worlds “1” and “2” in one stable model to obtain another). But any one of these isomorphic stable models can be used for answering EFEC queries.

In particular, we have sound and complete reasoning as regards the knowledge predicates defined in (EFEC9)-(EFEC14). For example, let  $Th = (\text{FEC1}) \wedge \dots \wedge (\text{FEC5}) \wedge (\text{EFEC1}) \wedge \dots \wedge (\text{EFEC26}) \wedge (\text{S1}) \wedge \dots \wedge (\text{S13})$ , let  $sm(Th)$  be a stable model returned by `Clingo`, let  $\theta_a, \theta_n \in \{0, \dots, \text{maxinstant}\}$ , and let  $\zeta$  and  $\nu$  be ground fluent and value terms. Then  $Th \models \text{KnowsValueIs}(\langle \omega, \theta_a \rangle, \zeta, \theta_n, \nu)$  if and only if  $\text{knowsValueIs}(\langle \omega, \theta_a \rangle, \zeta, \theta_n, \nu)$  is in  $sm(Th)$  for every world  $\omega \in \{1, \dots, \text{maxworld}\}$ . Similarly,  $Th \models \neg \text{KnowsValueIs}(\langle \theta_a \rangle, \zeta, \theta_n, \nu)$  if and only if  $\neg \text{knowsValueIs}(\langle \omega, \theta_a \rangle, \zeta, \theta_n, \nu)$  is in  $sm(Th)$  for every  $\omega \in \{1, \dots, \text{maxworld}\}$ , where “ $\neg$ ” represents classical negation in `Clingo`. This is because the actual world  $W_a$  must be isomorphic to exactly one such  $\omega$  in any model. A UNIX script is given at <http://www.ucl.ac.uk/infostudies/efec> for testing a representative sample of 88 such entailments in `Clingo`, including all those in Proposition 1.

## 4 Summary, Related and Future Work

The contributions of this paper are (i) to generalize the EC of [23] to multi-valued (non-binary) fluents, (ii) to build upon this generalization to provide an EC framework for combined narrative, epistemic and causal reasoning under a possible-worlds approach, and (iii) to illustrate how, for discrete time, our representation may be given a sound and complete ASP encoding. EFEC is able to deal with triggered, concurrent, non-deterministic and conflicting action occurrences in a uniform manner under both discrete and continuous models of time. It facilitates reasoning about knowledge of both action occurrences and fluent values at past, present and future times, as well as epistemically feasible plan validation, where conditional actions’ conditions are guaranteed to be known by the time of potential execution. To the best of our knowledge, no other existing epistemic action formalism is able to deal with this combination of features. In particular, triggered events (and knowledge about them) have not previously been incorporated in epistemic reasoning frameworks. This is in spite of their recognised importance in modelling many domains, e.g. involving complex ramifications [26], or reasoning about biological, physical or mechanical systems [35, 22], and many modes of reasoning, e.g. evidence gathering and diagnosis.

Our work is related to, and inspired by, the work of Scherl and Levesque [30, 29], who used possible situations to specify how the mental state of an agent should change with ordinary

<sup>4</sup>In the shopping example, the unique value of `maxworld` is 9 and `Clingo` computes  $9! = 362880$  models.

and sense actions, providing a solution to the frame problem for knowledge. It evolved from Moore’s [24] Kripke-like formulation of epistemic notions of modal logic in action theories by reifying possible worlds as situations. Since then several other studies have extended this model with new features: [34] adapted the model in the context of the Fluent Calculus, [31] covered concurrent actions, [13] introduced epistemic modalities for groups of agents, and [32] extended the model to account for belief revision. The notions used in Section 3.4 to compare narratives across different possible worlds are somewhat related to the notion of a *history structure* in [3].

To our knowledge little work has previously been done in extending possible-worlds epistemic action theories to deal with non-deterministic actions (resulting in knowledge loss). (An exception is  $\mathcal{A}_k$  [20] that also accounts for conditional sensing, but not functional fluents, concurrent actions, narrative reasoning or triggered events.) This may in part be explained by the technical difficulty of ensuring that in each model there are sufficient possible worlds to model the lack of knowledge that ensues after a non-deterministic event. In our framework this has been made possible partly because non-deterministic effects are represented as conjuncts (of *Causes Value*) rather than disjuncts, so that in each model each conjunct can be associated with an accessible world. We see no reason why this solution should not be translated into Situation Calculus-related approaches.

Non-determinism has been studied in non-possible-worlds approaches [1, 8]. Epistemic action frameworks that use alternative knowledge models also include [25, 5, 33, 28, 19, 36, 18, 27]. This last EC-based work is limited to discrete time and binary fluents, but can model ramifications, a feature we have not yet looked at.

Circumscription and stable model semantics have both been used widely in action theories. Of particular relevance here is [7], which discusses computational aspects of the frame, ramification and qualification problems. Recent studies such as [11, 17] have identified syntactic classes where circumscription and stable model semantics coincide, permitting the reformulation of circumscriptive action theories, such as the Situation Calculus and the Event Calculus, into ASP [16]. Their implementations are both fast and expressive in comparison to previous SAT- or Prolog-based encodings, as exemplified by the ASP encoding introduced in [16, 14] and built upon in the present study.

Denecker et al. [6] distinguish two general styles in programming ASP applications, the ASP-belief methodology and the ASP-world methodology. In the former, a stable model represents the belief state of an epistemic agent, which can be viewed to some extent as capturing the beliefs obtained from a collection of possible worlds. In the latter, the different stable models correspond to possible worlds, resembling the KR methodology of general logic programming. The majority of ASP applications follow the ASP-world methodology. For instance [2] uses ASP to encode the traditional Kripke-based modelling of the epistemic notions of an agent for the multi-agent case, presenting a way to implement a variety of commonly met scenarios in the field, such as the muddy children problem.

Most recently [9] gives an ASP implementation of a theory *HPX* that can refer to knowledge about past world states. It uses an approximation of possible-worlds semantics for solving postdiction problems. *HPX* has reduced computational complexity in comparison to previous theories, for deterministic domains where sensing can also occur concurrently with acting.

Our future plans include more formal or general methods of showing the correctness and limitations of our approach (other than case studies) and its correspondence with other frameworks. Additionally, we aim to generalise our framework in various ways. For example, [21] gives some preliminary ideas on how to include complex fluent conditions within conditional actions. We will also work further on modeling both hypothetical and explicit knowledge about the future (e.g. drawing on [4]), as well as on considerations of belief (and belief revision) rather

than knowledge, nested knowledge/belief structures, and multi-agent domains.

## References

- [1] Matteo Baldoni, Alberto Martelli, Viviana Patti, and Laura Giordano. Programming rational agents in a modal action logic. *Annals of Mathematics and Artificial Intelligence*, 41(2-4):207–257, 2004.
- [2] Chitta Baral, Gregory Gelfond, Tran Cao Son, and Enrico Pontelli. Using answer set programming to model multi-agent scenarios involving agents’ knowledge about other’s knowledge. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS ’10, pages 259–266, 2010.
- [3] B. Bennett and A. Galton. A unifying semantics for time and events. *Artificial Intelligence*, 153:13–48, 2004.
- [4] Ernest Davis and Leora Morgenstern. A first-order theory of communication and multi-agent plans. *J. Log. and Comput.*, 15(5):701–749, October 2005.
- [5] R. Demolombe and M. P. Pozos-Parra. A simple and tractable extension of situation calculus to epistemic logic. In *Twelfth International Symposium on Methodologies for Intelligent Systems (ISMIS-00)*, pages 515–524, 2000.
- [6] Marc Denecker, Joost Vennekens, Hanne Vlaeminck, Johan Wittocx, and Maurice Bruynooghe. Answer Set Programming’s contributions to classical logic. In Marcello Balduccini and Tran Son, editors, *Lecture Notes in Computer Science*. Springer, 2011.
- [7] Yannis Dimopoulos, Antonis C. Kakas, and Loizos Michael. Reasoning about actions and change in answer set programming. In Vladimir Lifschitz and Ilkka Niemel, editors, *LPNMR*, volume 2923 of *Lecture Notes in Computer Science*, pages 61–73. Springer, 2004.
- [8] Thomas Eiter, Wolfgang Faber, Nicola Leone, Gerald Pfeifer, and Axel Polleres. A logic programming approach to knowledge-state planning: Semantics and complexity. *ACM Trans. Comput. Logic*, 5(2):206–263, 2004.
- [9] M. Eppe, M. Bhatt, and F. Dylla. Approximate epistemic planning with postdiction as answer-set programming. In Pedro Cabalar and TranCao Son, editors, *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 8148 of *Lecture Notes in Computer Science*, pages 290–303. Springer Berlin Heidelberg, 2013.
- [10] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, Massachusetts, 1995.
- [11] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz. Stable models and circumscription. *Artif. Intell.*, 175(1):236–263, 2011.
- [12] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *aicom*, 24(2):105–124, 2011.
- [13] Ryan F. Kelly and Adrian R. Pearce. Complex epistemic modalities in the situation calculus. In *KR*, pages 611–620, 2008.
- [14] Tae-Won Kim, Joohyung Lee, and Ravi Palla. Circumscriptive event calculus as answer set programming. In *Proceedings of the 21st international joint conference on Artificial intelligence, IJCAI’09*, pages 823–829, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
- [15] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.
- [16] Joohyung Lee and Ravi Palla. Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming. *J. Artif. Int. Res.*, 43(1):571–620, 2012.
- [17] Vladimir Lifschitz and Fangkai Yang. Translating first-order causal theories into answer set programming. In *Proceedings of the 12th European conference on Logics in artificial intelligence*,

- JELIA'10, pages 247–259, 2010.
- [18] Yongmei Liu and Gerhard Lakemeyer. On first-order definability and computability of progression for local-effect actions and beyond. In *Proceedings of the 21st international joint conference on Artificial intelligence*, IJCAI'09, pages 860–866, 2009.
  - [19] Yongmei Liu and Hector J. Levesque. Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In *19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 522–527, 2005.
  - [20] J. Lobo, G. Mendez, and S. R. Taylor. Knowledge and the action description language a. *Theory and Practice of Logic Programming*, 1(2):129–184, 2001.
  - [21] R. Miller, L. Morgenstern, and T. Patkos. Reasoning about knowledge and action in an epistemic event calculus. In *proceedings of the Eleventh International Symposium on Logical Formalizations of Commonsense Reasoning*, [www.commonsense2013.cs.ucy.ac.cy/docs/commonsense2013\\_submission\\_3.pdf](http://www.commonsense2013.cs.ucy.ac.cy/docs/commonsense2013_submission_3.pdf), Cyprus, May 27-29, 2013.
  - [22] R. Miller and M. Shanahan. Reasoning about discontinuities in the event calculus. In *5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 63–74. Morgan Kaufmann, 1996.
  - [23] R. Miller and M. Shanahan. Some alternative formulations of the event calculus. *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert Kowalski Part 2, Lecture Notes in Artificial Intelligence*, 2408(1):452–490, 2002. A. Kakas and F. Sadri (ed.s).
  - [24] R. C. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*, pages 319–358. J. Hobbs, R. Moore (Eds.), 1985.
  - [25] Leora Morgenstern. Knowledge preconditions for actions and plans. In *IJCAI*, pages 867–874, 1987.
  - [26] E. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, 2006.
  - [27] T. Patkos and D. Plexousakis. Reasoning with knowledge, action and time in dynamic and uncertain domains. In *21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 885–890, Pasadena, California, USA, 2009.
  - [28] R. Petrick and H. Levesque. Knowledge equivalence in combined action theories. In *KR'02*, pages 303–314, 2002.
  - [29] R. Scherl and H. Levesque. The frame problem and knowledge-producing actions. In *Eleventh National Conference on Artificial Intelligence (AAAI'93)*, pages 689–695, Washington D.C, USA, August 18-20 1993. AAAI Press, Menlo Park, California, USA. R. Fikes and W. Lehnert (eds.).
  - [30] R. Scherl and H. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39, 2003.
  - [31] Richard B. Scherl. Reasoning about the interaction of knowledge, time and concurrent actions in the situation calculus. In *18th international joint conference on Artificial intelligence*, IJCAI'03, pages 1091–1096, 2003.
  - [32] Steven Shapiro, Maurice Pagnucco, Yves Lespérance, and Hector J. Levesque. Iterated belief change in the situation calculus. *Artif. Intell.*, 175(1):165–192, 2011.
  - [33] T. C. Son and C. Baral. Formalizing sensing actions – a transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, 2001.
  - [34] M. Thielscher. Representing the knowledge of a robot. In *KR'00*, pages 109–120, 2000.
  - [35] Nam Tran and Chitta Baral. Reasoning about Triggered Actions in AnsProlog and Its Application to Molecular Interactions in Cells. In *KR2004*, pages 554–564, 2004.
  - [36] Stavros Vassos and Hector Levesque. Progression of situation calculus action theories with incomplete information. In *20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 2024–2029, 2007.