Foundations for a Probabilistic Event Calculus

Fabio Aurelio D'Asaro(✉), Antonis Bikakis, Luke Dickens, and Rob Miller

Department of Information Studies, University College London, London, UK
{uczcfad,a.bikakis,l.dickens,r.s.miller}@ucl.ac.uk

Abstract. We present PEC, an Event Calculus (EC) style action language for reasoning about probabilistic causal and narrative information. It has an action language style syntax similar to that of the EC variant $\mathcal{M}$odular-$\mathcal{E}$. Its semantics is given in terms of *possible worlds* which constitute possible evolutions of the domain, and builds on that of Epistemic Functional EC (EFEC). We also describe an ASP implementation of PEC and show the sense in which this is sound and complete.

1 Introduction

The Event Calculus (EC) [6] is a well-known approach to reasoning about the effects of a narrative of action occurrences (events) along a time line. This paper briefly summarises [3], which describes PEC, an adaptation of EC able to reason with probabilistic causal knowledge. There are numerous applications for this kind of probabilistic reasoning, e.g. in modelling medical, environmental, legal and commonsense domains, and in complex activity recognition and security monitoring. Full technical details of PEC are in [3]. Its main characteristics are: (i) it supports EC-style narrative reasoning, (ii) it uses a tailored action language syntax and semantics, (iii) it uses a *possible worlds* semantics to naturally allow for *epistemic* extensions, (iv) for a wide subset of domains it has a sound and complete ASP implementation, and (v) its generality allows in principle for the use of other models of uncertainty, e.g. truth-functional belief or Dempster-Schafer theory. Although other formalisms exist for probabilistic reasoning about actions (see e.g. [1,2,5,11,12]), PEC is, to our knowledge, the only framework to combine together these features. As shown in [3] it can be used to model scenarios such as:

Scenario 1 (Coin Toss). *A coin initially (instant* 0*) shows Heads. A robot can attempt to toss the coin, but there is a small chance that it will fail to pick it up, leaving the coin unchanged. The robot attempts to toss the coin (instant* 1*).*

Scenario 2 (Antibiotic). *A patient has a rash often associated with a bacterial infection, and can take an antibiotic known to be reasonably effective. Treatment is not always successful, and if not may still clear the rash. Failed treatment leaves the bacteria resistant. The patient is treated twice (instants* 1 *and* 3*).*

Other than the primary author, authors are listed alphabetically.

M. Balduccini and T. Janhunen (Eds.): LPNMR 2017, LNAI 10377, pp. 57–63, 2017.
DOI: 10.1007/978-3-319-61660-5_7

2 Overview of PEC's Syntax and Semantics

A PEC *domain language* consists of a a finite non-empty set $\mathcal{F}$ of *fluents*, a finite set $\mathcal{A}$ of *actions*, a finite non-empty set $\mathcal{V}$ of *values* such that $\{\top, \bot\} \subseteq \mathcal{V}$, a function $vals: \mathcal{F} \cup \mathcal{A} \rightarrow 2^{\mathcal{V}} \setminus \emptyset$, and a non-empty set $\mathcal{I}$ of *instants* with minimum element $\bar{0}$ w.r.t. total ordering $\leq$. For $A \in \mathcal{A}$ we impose $vals(A) = \{\top, \bot\}$. Our approach is to model a given domain with action-language-like propositions that specify (probabilistic) causal and narrative information. For example, Scenario 1 is modelled using the following domain description $\mathcal{D}_C$:

$$Coin\,\textbf{takes-values}\,\{Heads, Tails\} \tag{C1}$$

$$\textbf{initially-one-of}\{(Coin{=}Heads, 1)\} \tag{C2}$$

$$Toss\,\textbf{causes-one-of}\; \{(\{Coin{=}Heads\}, 0.49), (\{Coin{=}Tails\}, 0.49), (\emptyset, 0.02)\} \tag{C3}$$

$$Toss\,\textbf{performed-at}\,1 \tag{C4}$$

More generally *v-propositions*, such as (C1), have the form

$$F\,\textbf{takes-values}\,\{V_1, \ldots, V_m\} \tag{1}$$

for $F \in \mathcal{F}$, $m \geq 1$, $V_i \in \mathcal{V}$ for all $1 \leq i \leq m$, and $\{V_1, \ldots, V_m\} = vals(F)$. *c-propositions* such as (C3) modeling causal relationships are of the form

$$\theta\,\textbf{causes-one-of}\,\{O_1, O_2, \ldots, O_m\} \tag{2}$$

where formula θ captures preconditions, and $O_1, \ldots, O_m$ are alternative *outcomes* – partial assignments of fluent values paired with probabilities that sum to 1. Initial conditions are declared via *i-propositions* of the form

$$\textbf{initially-one-of}\{O_1, O_2, \ldots, O_m\} \tag{3}$$

and action occurrences are identified through *p-propositions* of the form

$$A\,\textbf{performed-at}\,I\,\textbf{with-prob}\,P^+ \tag{4}$$

for $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $P^+ \in (0, 1]$. When $P^+ = 1$ it is omitted as in (C4).

A *domain description* is a finite set $\mathcal{D}$ of v-, c-, p- and i-propositions such that: (i) for any two distinct c-propositions in $\mathcal{D}$ with bodies θ and θ', there is no state compatible[1] with both θ and θ', (ii) $\mathcal{D}$ contains exactly one i-proposition, and (iii) $\mathcal{D}$ contains exactly one v-proposition for each $F \in \mathcal{F}$.

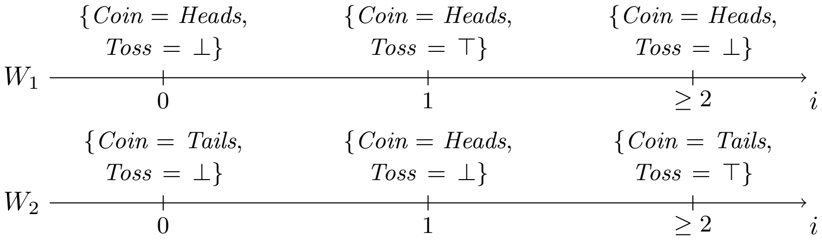
PEC's semantics describes how domain descriptions entail *h-propositions* of the form

$$\varphi\,\textbf{holds-with-prob}\,P \tag{5}$$

where $P \in [0, 1]$, and φ is an *i-formula* (time-stamped formula). For example, $\mathcal{D}_C$ entails '$[Coin{=}Heads]@2$ **holds-with-prob** 0.51'.

[1] i.e. Taking literals as propositions, there is no state that is a classical Herbrand model of both θ and θ'.

PEC has a *possible-worlds* semantics. A *world* is an evolution of the environment, i.e. a function $W : \mathcal{I} \rightarrow \mathcal{S}$, where $\mathcal{S}$ is the set of all *states* (complete assignments of values to fluents and actions). $\mathcal{W}$ denotes the set of all worlds. Worlds can be pictured as timelines with information about the current state attached at each instant. E.g. two worlds for Scenario 1 can be visualised as:



Intuitively, W_1 is consistent with domain description $\mathcal{D}_C$ as it represents a coherent history of what could have happened in Scenario 1, whereas W_2 does not (e.g., changes occur when no action is performed, an infinite number of actions are performed, etc...). For this reason, world W_1 said to be *well-behaved w.r.t.* $\mathcal{D}_C$, whereas W_2 is not. The semantics captures this notion with the concept of a *trace* – a chain of effects matching both a unique world W and the domain description $\mathcal{D}$, through consistency with propositions in $\mathcal{D}$ and a persistence condition. In other words, a world W represents an evolution of *state*, whereas a trace of W represents a legal causal history w.r.t. W and a corresponding domain description. For example, two traces for W_1 w.r.t. $\mathcal{D}_C$ are:

$$t_1 = \langle(\{Coin{=}Heads\}, 1)@⧖,\ (\{Coin{=}Heads\}, 0.49)@1\rangle$$

$$t_2 = \langle(\{Coin{=}Heads\}, 1)@⧖,\ (\emptyset, 0.02)@1\rangle$$

where the special symbol ⧖ is used to deal with the initial condition. The *evaluation* of a trace *tr*, written $\epsilon(tr)$, is the product of all real values appearing in it. In our example, $\epsilon(t_1) = 0.49$ and $\epsilon(t_2) = 0.02$. W_2 has no trace w.r.t. $\mathcal{D}_C$ and so is not well-behaved w.r.t. $\mathcal{D}_C$.

PEC's semantics defines a probability distribution over worlds. To show this, we first define a *[0,1]-interpretation* as a function from $\mathcal{W}$ to $[0, 1]$, and, given a domain description $\mathcal{D}$, single out a unique [0,1]-interpretation $M_\mathcal{D} : \mathcal{W} \mapsto [0, 1]$ called the *model* of $\mathcal{D}$. For world W, well-behaved w.r.t. $\mathcal{D}$, $M_\mathcal{D}(W)$ is the sum of values $\epsilon(tr)$ for all corresponding traces *tr* of W. If W is not well-behaved, then $M_\mathcal{D}(W) = 0$. $M_\mathcal{D}$ is extended to a function $M^*_\mathcal{D}$ over i-formulas as follows:

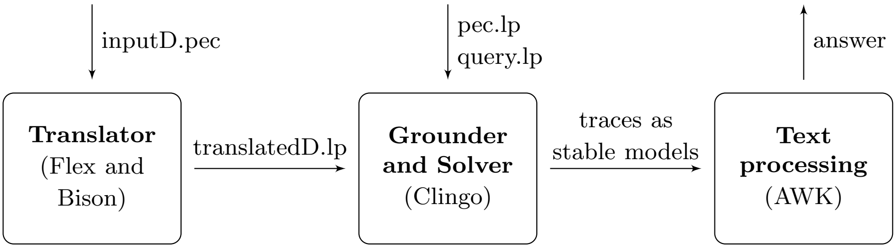
$$M^*_\mathcal{D}(\varphi) = \sum_{W |\!\models \varphi} M_\mathcal{D}(W) \tag{6}$$

where $W \mathrel{|\!\models} \varphi$ indicates that φ is *satisfied* in W (in the obvious sense, see [3]). We say that 'φ **holds-with-prob** P' *is entailed by* $\mathcal{D}$ iff $M^*_\mathcal{D}(\varphi) = P$.

In [3], we prove that M^* is a *probability function* (see assumptions (P1) and (P2) from [9, Chap. 1]). Note that M^* could be alternatively defined to satisfy different axioms (see e.g. (DS1–3) from [9, Chap. 1] for Dempster-Schafer belief functions or [9, Chap. 5] for truth-functional belief functions).

3 ASP Implementation

We have implemented PEC for the class of domains in which the bodies θ of c-propositions are conjunctive formulas. The implementation and example domain descriptions can be found at https://github.com/dasaro/pec. For this, a translator turns a PEC domain description $\mathcal{D}$ into an ASP program using standard lexical analyser Flex and parser generator Bison. A grounder and solver, Clingo [4], then processes the translator's output together with the domain-independent part of the semantics and a query (both in ASP). This returns a collection of answer sets, each representing a trace and the corresponding well-behaved world. A standard text processing tool, AWK, then evaluates $\epsilon(tr)$ for each answer set trace tr and sums these to give a probability for the query using Eq. (6). The following diagram illustrates this procedure:



A general translation procedure from this class of PEC domain descriptions to ASP programs is given in [3]. To illustrate, $\mathcal{D}_C$ is translated to:

$$\begin{aligned}&\mathit{fluent}(\mathit{coin}).\\ &\mathit{action}(\mathit{toss}).\\ &\mathit{instant}(0..\mathit{maxinst}).\end{aligned} \tag{TC0}$$

$$\begin{aligned}&\mathit{possVal}(\mathit{coin}, \mathit{heads}).\\ &\mathit{possVal}(\mathit{coin}, \mathit{tails}).\end{aligned} \tag{TC1}$$

$$\begin{aligned}&\mathit{belongsTo}((\mathit{coin}, \mathit{heads}), \mathit{id}_1^0).\\ &\mathit{initialCondition}((\mathit{id}_1^0, 1)).\end{aligned} \tag{TC2}$$

$$\begin{aligned}&\mathit{belongsTo}((\mathit{coin}, \mathit{heads}), \mathit{id}_1^1).\\ &\mathit{causesOutcome}((\mathit{id}_1^1, 49/100), I) \leftarrow \mathit{holds}(((\mathit{toss}, \mathit{true}), I)).\end{aligned} \tag{TC3.1}$$

$$\begin{aligned}&\mathit{belongsTo}((\mathit{coin}, \mathit{tails}), \mathit{id}_2^1).\\ &\mathit{causesOutcome}((\mathit{id}_2^1, 49/100), I) \leftarrow \mathit{holds}(((\mathit{toss}, \mathit{true}), I)).\end{aligned} \tag{TC3.2}$$

$$\mathit{causesOutcome}((\mathit{id}_3^1, 2/100), I) \leftarrow \mathit{holds}(((\mathit{toss}, \mathit{true}), I)). \tag{TC3.3}$$

$$\mathit{performed}(\mathit{toss}, 1). \tag{TC4}$$

The domain-independent part of the implementation is as follows:

$$possVal(A, true) \leftarrow action(A).$$
$$possVal(A, false) \leftarrow action(A). \quad \text{(PEC1)}$$

$$fluentOrAction(X) \leftarrow fluent(X);\ action(X). \quad \text{(PEC2)}$$

$$literal((X, V)) \leftarrow possVal(X, V). \quad \text{(PEC3)}$$

$$iLiteral((L, I)) \leftarrow literal(L), instant(I). \quad \text{(PEC4)}$$

$$1\{\ holds(((X, V), I)) : iLiteral(((X, V), I))\ \}1 \leftarrow instant(I),\ fluentOrAction(X). \quad \text{(PEC5)}$$

$$inOcc(I) \leftarrow instant(I),\ causesOutcome(O, I). \quad \text{(PEC6)}$$

$$1\{\ effectChoice(O, I) : causesOutcome(O, I)\ \}1 \leftarrow inOcc(I). \quad \text{(PEC7)}$$

$$1\{\ initialChoice(O) : initialCondition(O)\ \}1. \quad \text{(PEC8)}$$

$$\bot \leftarrow action(A), instant(I), holds(((A, true), I)),\ not\ performed(A, I). \quad \text{(PEC9)}$$

$$\bot \leftarrow action(A), instant(I), not\ holds(((A, true), I)),\ performed(A, I). \quad \text{(PEC10)}$$

$$\bot \leftarrow initialChoice((S, P)),\ literal(L), belongsTo(L, S),\ not\ holds((L, 0)). \quad \text{(PEC11)}$$

$$\bot \leftarrow instant(I),\ effectChoice((X, P), I), fluent(F),\ belongsTo((F, V), X), not\ holds(((F, V), I + 1)), I < maxinst. \quad \text{(PEC12)}$$

$$\bot \leftarrow instant(I),\ fluent(F),\ not\ holds(((F, V), I)), effectChoice((X, P), I),\ not\ belongsTo((F, V), X), holds(((F, V), I + 1)), I < maxinst. \quad \text{(PEC13)}$$

$$\bot \leftarrow fluent(F),\ instant(I),\ holds(((F, V), I)),\ not\ inOcc(I), not\ holds(((F, V), I + 1)),\ I < maxinst. \quad \text{(PEC14)}$$

Correctness of the translation and implementation are guaranteed by the following proposition, more details of which (including a proof) are given in [3].

Proposition 1 (Soundness and Completeness). *Z is a stable model of the translated domain description $\mathcal{D}$ together with the domain-independent part of PEC iff Z* represents *a well-behaved world W and one of its traces w.r.t. $\mathcal{D}$.*

Intuitively, this proposition states that if we interpret the elements of a stable model Z of the translated domain description and the domain-independent part of PEC as their natural semantic counterpart (e.g., $holds(((F, V), I))$ is interpreted as $F{=}V \in W(I)$), then this interpretation is a trace together with its corresponding well-behaved world W w.r.t. $\mathcal{D}$. The trace and world are then said to be *represented* by Z. Conversely, for every well-behaved world W w.r.t. $\mathcal{D}$ and one of its traces *tr* there exists a stable model Z of the program such that

Z represents W and tr. It is in this sense that our implementation is *sound* and *complete*. Our proof in [3] relies on the Splitting Theorem [7].

4 Future Work

PEC semantics is defined in terms of (possible) *worlds* with a view to adding epistemic features in the future (see e.g. [10]). Our initial investigations in this respect focus on representing *imperfect sensing actions* and *actions conditioned on knowledge* acquired in previous instants (similar to the approach in the EFEC extension of FEC [8]). We envisage including *s-propositions* such as

$$\mathit{See}\,\textbf{senses}\;\mathit{Coin}\,\textbf{with-accuracies}\;\begin{pmatrix}0.9 & 0.1\\ 0.3 & 0.7\end{pmatrix}$$

which represents that our coin-tossing robot can imperfectly sense the current face showing on the coin, and *conditional p-propositions* such as

$$\mathit{Toss}\,\textbf{performed-at}\;2\;\textbf{if-believes}\;(\mathit{Coin}{=}\,\mathit{Tails},(0.65,1])$$

which represents that the robot will toss again if it believes with a greater than 65% probability that the first toss resulted in *Tails*. Preliminary results indicate that our possible worlds semantics can be readily extended to cover these notions.

There are several other ways in which the present work can be continued. For instance, the problem of *elaboration tolerance*, which plays an important role in classical reasoning about actions, needs to be reviewed and solved in our setting. A related point is that of *underspecification*, i.e. what an agent can reasonably infer from a domain in which the initial conditions and the effects of actions are not entirely specified (even probabilistically). Finally, a crucial point is that of *computational efficiency*. Indeed, the intractability of several computational problems arising in this setting (such as temporal projection) suggests that techniques (e.g. Monte Carlo Markov Chain) are needed to efficiently approximate the correct answer to a given query with an appropriate degree of confidence.

Related Work: For a discussion of related work see [3].

References

1. Bacchus, F., Halpern, J.Y., Levesque, H.J.: Reasoning about noisy sensors and effectors in the situation calculus. Artif. Intell. **111**(1), 171–208 (1999)
2. Baral, C., Tran, N., Tuan, L.C.: Reasoning about actions in a probabilistic setting. In: AAAI/IAAI, pp. 507–512 (2002)
3. D'Asaro, F.A., Bikakis, A., Dickens, L., Miller, R.: Foundations for a probabilistic event calculus: Technical report (2017). http://arxiv.org/abs/1703.06815
4. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Clingo = ASP + control: Preliminary report (2014). http://arxiv.org/abs/1405.3694
5. Iocchi, L., Lukasiewicz, T., Nardi, D., Rosati, R.: Reasoning about actions with sensing under qualitative and probabilistic uncertainty. TOCL **10**(1), 1–39 (2009)

6. Kowalski, R., Sergot, M.: A logic-based calculus of events. In: Schmidt, J.W., Thanos, C. (eds.) Foundations of Knowledge Base Management, pp. 23–55. Springer, Heidelberg (1989)
7. Lifschitz, V., Turner, H.: Splitting a logic program. In: ICLP, pp. 23–37 (1994)
8. Ma, J., Miller, R., Morgenstern, L., Patkos, T.: An epistemic event calculus for asp-based reasoning about knowledge of the past, present and future. In: LPAR-19, vol. 26, pp. 75–87 (2014)
9. Paris, J.B.: The Uncertain Reasoner's Companion: A Mathematical Perspective, vol. 39. Cambridge University Press, Cambridge (2006)
10. Scherl, R.B., Levesque, H.J.: Knowledge, action, and the frame problem. Artif. Intell. **144**(1–2), 1–39 (2003)
11. Skarlatidis, A., Artikis, A., Filippou, J., Paliouras, G.: A probabilistic logic programming event calculus. TPLP **15**, 213–245 (2015)
12. Skarlatidis, A., Paliouras, G., Artikis, A., Vouros, G.A.: Probabilistic event calculus for event recognition. ACM Trans. Comput. Logic **16**(2), 11:1–11:37 (2015)