Argumentation and Temporal Persistence

E. Hadjisoteriou* and A. Kakas

University of Cyprus, Dept. of Computer Science,
Kallipoleos 75, 1678 Nicosia, Cyprus
{csp7he2,antonis}@cs.ucy.ac.cy
http://www.cs.ucy.ac.cy/

Abstract. We study how the problem of temporal projection can be formalized in terms of argumentation. In particular, we extend earlier work of translating the language $\mathcal{E}$ for Reasoning about Actions and Change into a Logic Programming argumentation framework, by introducing new types of arguments for (i) backward persistence and (ii) persistence from observations. This forms a conservative extension of the language $\mathcal{E}$ that gives semantic meaning to domains that cannot be interpreted in the language $\mathcal{E}$.

Keywords: Argumentation, narrative information, observations, backwards and forwards persistence

1 Introduction and Motivation

Given some narrative information we can use argumentation to capture temporal projection from this and general knowledge about the causal laws of our problem domain. As shown in [4], where the language $\mathcal{E}$ [3] for reasoning about actions and change was formalized in terms of argumentation, default persistence over time is captured by assigning higher priority to arguments that are based on later events over the arguments based on earlier events.

In this paper we extend this argumentation based formulation of language $\mathcal{E}$ by introducing also arguments based on property observations. Thus, we approach the qualification problem[6]. We review how temporal persistence is captured and introduce new arguments for backward persistence. This will allow us to recover and also extend language $\mathcal{E}$, giving a semantic meaning to domains that cannot be interpreted in the language $\mathcal{E}$. With this form of backward persistence the extended interpretation of the language $\mathcal{E}$ comes closer to the original Event calculus [5] which also include notions for backward temporal conclusions.

As an example of how language $\mathcal{E}$ is extended consider a parking domain, with action constant $ParkingCar$ and property fluent $CarInParkingSpace$ and the narrative that we park the car at time 4 and that later at time 8 we observed that the car is not where it was parked:

$ParkingCar$ initiates $CarInParkingSpace$ $\quad (\Delta_1)$

* An earlier version of this paper appeared in the proceedings of the 7th PLS.

Andrew V. Jones (Ed.): Proceedings of the 1st Imperial College Student Workshop (ICCSW '11), pp. 31–38. September 29th–30th 2011, London UK.

$ParkingCar$ happens-at 4 ($\mathit{\Delta}_2$)
$\neg CarInParkingSpace$ holds at time 8 ($\mathit{\Delta}_3$)
For domains like this, where a fluent (e.g. $CarInParkingSpace$) changes its truth value without any known causal explanation, language $\mathcal{E}$ does not give a model. On the other hand, our extended argumentation framework of the language $\mathcal{E}$ that includes arguments for observations and for backwards persistence as well allows arguments for both truth values of the fluent within this time interval. Forwards persistence from the action $ParkingCar$ ($\mathit{\Delta}_2$) that indicates $CarInParkingSpace$ for every time point $t > 4$ ($\mathit{\Delta}_1$) come in conflict with backwards persistence from the observation argument $\neg CarInParkingSpace$ ($\mathit{\Delta}_3$). Allowing same priority to conflicting forward persistence over backwards persistence will give the natural interpretation of unknown value for the fluent $CarInParkingSpace$ for every $t \in (4, 8)$.

By introducing backwards persistence in our argumentation framework and assigning suitable priorities we can fully recover and also extend language $\mathcal{E}$. In our extended version we get models to domains that language $\mathcal{E}$ can not interpret. Furthermore, language $\mathcal{E}$ [4] handles domains without observations. We allow observations as part of our argumentation framework and assign priorities against all the other already existing arguments. As models must comply to all observations we treat observations as indisputable arguments. The reason we can do this is because of backwards persistence arguments and the priority assigned over forward persistence arguments.

The rest of the paper is organized as follows. Section 2 gives a brief review of the language $\mathcal{E}$. In section 3 we give the extended argumentation framework of $\mathcal{E}$. Section 4 presents our formal results and section 5 contains our conclusions.

2 A Brief Review of Language $\mathcal{E}$

Language $\mathcal{E}$ [3] is an action language that uses three kinds of propositions: **c-propositions**, of the form "A initiates F when C" or "A terminates F when C", **h-propositions** of the form "A happens-at T" and **t-propositions** of the form "L holds-at T", where A is an action constant, F is a fluent constant, T is a time point, L is a fluent literal and C is a set of fluent literals. Computational complexity is not the main concern of this paper. Lets note that the number of such models is exponentially high.

Models of the language $\mathcal{E}$ assign a truth value, $\{true\, or\, false\}$ at every fluent and every time point in the domain such that within any time interval the truth value assigned by a model to any fluent remains the same or **persists**, changing from false to true (resp. from true to false) at an initiation (resp. termination) time point. A time point T is an initiation (resp. termination) point when the problem domain description contains a combination of a c-proposition "A initiates (resp. terminates) F when C" and an h-proposition "A happens-at T", such that the model satisfies C at T. Furthermore, a model must confirm all the t-propositions given in the problem domain description resulting from fluent observations of the state of the world at various time points. Entailment and consistency of formulae of the form "L holds-at T", where L is a fluent literal are then defined in the usual way. For formal definitions and results the reader is referred to [3].

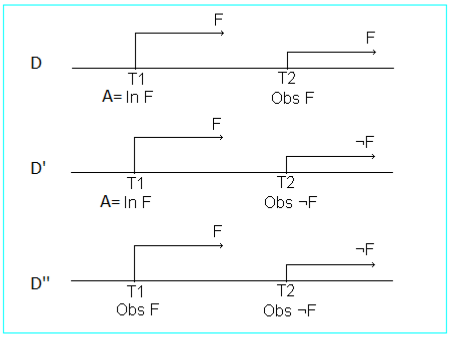

Fig. 1. Example Domains

As examples let us consider the domain descriptions $D, D^{'}$ and $D^{''}$ illustrated in (Figure 1), where A is an action and $T_1 < T_2$ are two time points. In domain D we have an initiation point at time T_1 and at time T_2 we observe F. Models of language $\mathcal{E}$, for domain D, require F to be true for all $T > T_1$ whereas, for $T^{'} \leq T_1$ a model can assign F to be either true or false at all such time points. In the domain $D^{'}$, where we have an initiation point at time T_1 and observation $\neg F$ at time T_2, and the domain $D^{''}$, where we have an observation F at time T_1 and an observation $\neg F$ at time T_2, the language $\mathcal{E}$ is inconsistent and has no models. The persistence of the F holding onwards from T_1 cannot be reconciled with the observation of $\neg F$ at T_2. Lets note that domain $D^{'}$ is similar to the parking domain example.

3 Argumentation Formulation

Language $\mathcal{E}$ has been reformulated in terms of argumentation [4]. In this the information from t-propositions (observations) is imposed as a-posteriori constraints on the argumentation formulation. We will extend this reformulation so that t-propositions are taken into account directly within the argumentation. To do so we will generalize the original formulation by allowing backward temporal persistence arguments as well as forward ones.

Following the earlier approach in [4], we define an argumentation logic program with priorities corresponding to a given domain description as follows.

Definition 1. *[argumentation program of D] The argumentation program corresponding to a domain D is $\Delta \equiv (B(D), A, <)$ where:*

- *The background knowledge, $B(D)$, contains the rule definitions of $Initiation(F, t)$ and $Termination(F, t)$ from c-propositions in D, facts of the form $Observation(L, T)$ for every t-proposition "L holds at T" in D and actions of the form A for every h-propositions "A happens-at T" in D.*
- *A consists of the following argument rules: For all time points t_1, t_2 and t such that $t_1 < t < t_2$,*

  Persistence:

$HoldsAt(f, t_2) \leftarrow HoldsAt(f, t)$	$PFP[f, t_2; t]$
$HoldsAt(f, t_1) \leftarrow HoldsAt(f, t)$	$PBP[f, t_1; t]$
$\neg HoldsAt(f, t_2) \leftarrow \neg HoldsAt(f, t)$	$NFP[f, t_2; t]$
$\neg HoldsAt(f, t_1) \leftarrow \neg HoldsAt(f, t)$	$NBP[f, t_1; t]$

  Local Generation Arguments:

$HoldsAt(f, t+1) \leftarrow Initiation(f, t)$	$PG_F[f, t]$
$\neg HoldsAt(f, t) \leftarrow Initiation(f, t)$	$PG_B[f, t]$
$\neg HoldsAt(f, t+1) \leftarrow Termination(f, t)$	$NG_F[f, t]$
$HoldsAt(f, t) \leftarrow Termination(f, t)$	$NG_B[f, t]$

  Local Observation Arguments:

$HoldsAt(f, t) \leftarrow Observation(f, t)$	$PO[f, t]$
$\neg HoldsAt(f, t) \leftarrow Observation(\neg f, t)$	$NO[f, t]$

  Assumption at 0:

$HoldsAt(f, 0)$	$PA[f, 0]$
$\neg HoldsAt(f, 0)$	$NA[f, 0]$

- *The priority (or strength of argument) relation, $<$, between these arguments is given below (t, t^*, t_1 and t_2 are time points):*

  If $t_1 < t_2$
  $PFP[f, t^*; t_1] < NFP[f, t^*; t_2]$, $NFP[f, t^*; t_1] < PFP[f, t^*; t_2]$,
  $PBP[f, t^*; t_2] < NBP[f, t^*; t_1]$, $NBP[f, t^*; t_2] < PBP[f, t^*; t_1]$,
  $NFP[f, t_2; t_1] < PO[f, t_2]$, $PFP[f, t_2; t_1] < NO[f, t_2]$,
  $NBP[f, t_1; t_2] < PO[f, t_1]$ *and* $PBP[f, t_1; t_2] < NO[f, t_1]$.
  At 0,
  $PA[f, 0] < NO[f, 0]$ *and* $NA[f, 0] < PO[f, 0]$.
  At t,
  $PG_B[f, t] < PO[f, t]$ *and* $NG_B[f, t] < NO[f, t]$.
  At $t+1$,
  $PG_F[f, t] < NO[f, t+1]$ *and* $NG_F[f, t] < PO[f, t+1]$.

Informally, the above priority makes forward persistence arguments that are based on later narrative information stronger and similarly for backward persistence arguments that are based on earlier narrative information. Also we assign higher priority to t-propositions over forward and backwards persistence. With this assignment observations become part of the argumentation rules and are treated as constrains that must be satisfied. However, note that there is no priority between conflicting forward and backward arguments. Such priorities can be additionally set when we wish to impose further properties on the temporal reasoning.

The semantics of these programs is given through the standard argumentation notion (see [1, 2]) of maximally admissible subsets of the given argumentation program, called **admissible extensions**. A subset of arguments is admissible if it does not derive $HoldsAt(f,t)$ and $\neg HoldsAt(f,t)$ for any fluent and time point and it can counter-attack any subset of arguments that attacks it. This attacking relation is defined such that a set of arguments would attack another if it derives a contrary conclusion and its argument rules in doing so are not weaker than the opposing argument rules. For the formal details please refer to [2, 4].

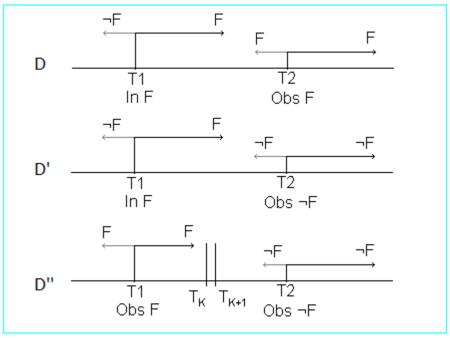

Fig. 2. Example Domains and Arguments

Comparing our earlier example domains D, D' and $D^{''}$ (see Figure 1) within this new argumentation framework (Figure 2) we see that in the new domain D, for all $T > T_1$ the strongest (and hence admissible) argument is for F to hold. For $T^{'} \leq T_1$ we can have admissible arguments for F or its negation $\neg F$ depending on the assumption we make at the initial time point. For the domain $D^{''}$ the strongest argument for all time points $T \geq T_2$ is for $\neg F$. For times between T_1 and T_2 we have admissible arguments for either F or $\neg F$: at some time point T_k, $T_1 \leq T_k$, the fluent F changes from true to false at T_{k+1}. This indicates that the given narrative has some missing information within this time interval that would explain the change in F. Similar results hold for $D^{'}$ where also in this case there exists an admissible extension where $\neg F$ holds for all times T, such that $T_1 \leq T \leq T_2$. This captures the possibility that the generation of F at T_1 has failed.

4 Formal Results

In this section we present a set of formal results that show how our proposal for an argumentation semantics gives a meaning to any theory even when domains have t-propositions. By allowing forward persistence to be non comparable to conflicting backwards persistence we can recover and also extend language $\mathcal{E}$ when this can not give a semantic meaning to a domain.

Property 1. Let D be a domain description and E an admissible extension of D. E is consistent (i.e. there does not exist a t-proposition "holds-at(f,t)" in D such that $D \models \neg$holds-at(f,t)).

Theorem 1. *Let D be a language $\mathcal{E}$ domain description and a countable number of h-propositions. Then:*

- *For every language $\mathcal{E}$ model, M, of D there exists an admissible extension, E, of the corresponding argumentation program $\Delta \equiv (B(D), A, <)$ such that E corresponds to M, i.e. $E \models$ holds-at(f,T) if and only if $M(f,T) = true$ and $E \models \neg$holds-at(f,T) if and only if $M(f,T) = false$.*
- *There exists a complete admissible extension D of the corresponding argumentation program $\Delta \equiv (B(D), A, <)$.*

For example, consider domain $D^{'}$ and $D^{''}$. With the new argumentation framework all maximally admissible extensions are consistent while in language $\mathcal{E}$ maximally admissible extensions are inconsistent.

Theorem 2 gives an interpretation of the extended semantic of the argumentation formulation in terms of the original language $\mathcal{E}$. We first need the following two lemmas:

Lemma 1. *Let D be a consistent domain and E a complete admissible extension of D. Let f be a fluent and $t_n < t_m$ two time points. If there does not exist a generation point for the fluent f in E at $t_1 \in [t_n, t_m)$ nor an observation point for the fluent f in E at $t_2 \in (t_n, t_m)$ and if*
$E \models$ holds-at(f, t_n) and $E \models$ holds-at(f, t_m) or
$E \models \neg$holds-at(f, t_n) and $E \models \neg$holds-at(f, t_m) then, there does not exist a time point $T \in [t_n, t_m)$ where the given fluent f changes its truth value in E, i.e. $E \models$ holds-at(f, T), for every $T \in [t_n, t_m]$ or $E \models \neg$holds-at(f, T), for every $T \in [t_n, t_m]$.

Informally, when no information is given in the narratives between two time periods that assign the same truth value for every fluent then a complete admissible extension gives a constant truth value for every fluent over this time period.

Lemma 2. *Let D be a consistent domain and E a complete admissible extension of D. Let f be a fluent and $t_n < t_m$ two time points. If there does not exists a generation point for the fluent f in E at $t_1 \in [t_n, t_m)$ nor an observation point for the fluent f in E at $t_2 \in (t_n, t_m)$ and if*
$E \models$ holds-at(f, t_n) and $E \models \neg$holds-at(f, t_m) or
$E \models \neg$holds-at(f, t_n) and $E \models$ holds-at(f, t_m) then, there exist at least one time points $T \in [t_n, t_m)$ where f can change its truth value in E, i.e.
$E \models$ holds-at(f, T) and $E \models \neg$holds-at$(f, T+1)$ or $E \models \neg$holds-at(f, T) and $E \models$ holds-at$(f, T+1)$. If the number of such time points is $k > 1$ then k is an odd number.

When two time periods assign opposite values for a fluent f and no information is given in the narratives then in a complete admissible extension there must exist k many (k is an odd number) time points between these two time points that change the truth value of f.

Theorem 2. *Let D be a domain description. For every maximally admissible extension E there exist a domain $D^{'}$ obtained from D by adding new events such that there exist a language $\mathcal{E}$ model, M, of $D^{'}$ that corresponds to E (i.e. $E \models$ holds-at(f, t) if and only if $M \models$ holds-at(f, t)).*

For example consider domain $D^{''}$ where from F at time T_1 we jump to $\neg F$ at time T_2. Let time point $T_k \in (T_1, T_2)$. By accepting that time T_k is a termination point for F we explain the semantic meaning given by argumentation to the domain.

To recover exactly the language $\mathcal{E}$ semantics we need to add extra priorities and specifically, to give preference to forward arguments over conflicting backwards arguments. The formal result for this is given in theorem 3.

Theorem 3. *In addition to the priorities given in definition 1 let also the following when $t_1 < t_2$:*
$PFP[f, t; t_1] > NBP[f, t; t_2]$ *and* $NFP[f, t; t_1] > PBP[f, t; t_2]$.

Then, every maximally admissible extension E, for any domain D corresponds to a model M of the language $\mathcal{E}$, of D.

5 Conclusions and Further Work

We have reexamined the argumentation reformulation of language $\mathcal{E}$ and introduced backwards persistence as well as forward persistence arguments. This enabled us to extend in a meaningful way domains that language $\mathcal{E}$ could not interpret. When language $\mathcal{E}$ is inconsistent within two time points, the argumentation interpretation corresponds to the unknown occurrences of events that could resolve this inconsistency.

As a future work we recommend a planning for more complicated domains with action A, fluents F and G such that A causes $\neg F$ and $\neg G$. In addition, even though there are many ways to deal with ramification problems (F_1 causes F_2 when L) we leave an open door that one can work on this issue further. Finally, all sets in this theory are countable. We are very interested to learn if there a way to extend this theory to uncountable sets and variables.

References

1. Phan Minh Dung, *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*, Artif. Intell. **77** (1995), no. 2, 321–357.
2. Antonis C. Kakas, Paolo Mancarella, and Phan Minh Dung, *The acceptability semantics for logic programs*, ICLP, 1994, pp. 504–519.

3. Antonis C. Kakas and Rob Miller, *A simple declarative language for describing narratives with actions*, J. Log. Program. **31** (1997), no. 1-3, 157–200.
4. Antonis C. Kakas, Rob Miller, and Francesca Toni, *An argumentation framework of reasoning about actions and change*, LPNMR, 1999, pp. 78–91.
5. R Kowalski and M Sergot, *A logic-based calculus of events*, New Gen. Comput. **4** (1986), no. 1, 67–95.
6. M. Thielscher, *The Qualification Problem: A Solution to the Problem of Anomalous Models*, AIJ **131** (2001), no. 1–2, 1–37.