

Reasoning about actions and change in argumentation

E. Hadjisoteriou* and A. Kakas

Department of Computer Science, University of Cyprus, Kallipoleos 75, 1678 Nicosia, Cyprus

(Received 31 December 2014; accepted 6 November 2015)

This paper studies how logic-based reasoning about actions and change (RAC) with its problems of temporal projection and qualification can be formalised in terms of argumentation. In particular, we extend earlier work of translating the language \mathcal{E} for RAC into a logic-based 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 a semantic meaning to domains that cannot be interpreted under \mathcal{E} thus addressing further the frame and (exogenous) qualification problems. As such the paper strengthens the link between argumentation theory and RAC in artificial intelligence.

Keywords: argumentation; reasoning about actions; narratives

1. Introduction and motivation

Reasoning about actions and change (RAC) has been an important problem in the development of artificial intelligence (AI). Starting with the early and foundational work of the situation and event calculi (Kowalski and Sergot, 1986; McCarthy and Hayes, 1969) there has been a wide interest in understanding and solving the central problems that underly RAC, namely the frame, ramification and qualification problems leading to several ‘action languages’ such as the \mathcal{A} , \mathcal{C} and \mathcal{E} languages (Gelfond and Lifschitz, 1993; Kakas and Miller, 1997; McCain and Turner, 1997) and the fluent calculus (Thielscher, 1999).

The relatively recent realisation that non-monotonic reasoning can be addressed using argumentation (e.g. Bench-Capon and Dunne, 2007; Bondarenko, Dung, Kowalski, and Toni, 1997; Dung, 1995; Kakas, Kowalski, and Toni, 1992) has led to the study of RAC through argumentation in works such as (Kakas, Miller, and Toni, 1999; Vo and Foo, 2005). Hence 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 Kakas et al. (1999), where the language \mathcal{E} (Kakas and Miller, 1997) for RAC was formalised in terms of argumentation, default persistence over time can be captured by assigning higher priority to arguments that are based on later events over arguments based on earlier events.

In this paper we extend this argumentation-based formulation of language \mathcal{E} by introducing arguments based on property observations that one typically finds in any given narrative. We will also introduce in the framework 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 under \mathcal{E} . With this form of backward persistence the extended interpretation of the language \mathcal{E} comes closer to the spirit of the original event calculus (EC) (Kowalski and Sergot, 1986) which also includes notions for backward temporal conclusions. Our work can

*Corresponding author. Emails: csp7he2@cs.ucy.ac.cy, evgenios.maths@gmail.com

thus also be seen as a way of increasing the expressiveness of the original EC by allowing for example positive but also negative observations.

We will be mainly concerned with domains that language \mathcal{E} cannot give models and exogenous qualification is required. From a purely technical point of view, we extend the argumentation formulation of language \mathcal{E} giving a semantics to domains where \mathcal{E} fails to do so. From a conceptual point of view, we aim to address the qualification problem in RAC (see, e.g. Thielscher, 2001) in a natural and modular way. This will rely on capturing a relative strength between arguments where, for example, arguments based on observations are stronger than other types of arguments.

RAC is an area that is concerned with the study of how (some of) the properties of our problem domain, normally called *Fluents*, change when new information is acquired such as the occurrence of *Actions* and how this view of the problem world is affected by the explicit observation of some properties holding or not at a particular stage (or time) in the flow of change. The change is governed by domain-specific *causal laws* that describe how the particular properties (or fluents) of interest are generated or terminated by actions and the domain-independent principle of *frame inertia* that properties do not change but rather persist, unless some action occurrence causes them to change via some causal law. The main aim of the reasoning is then to maintain, normally under the case of incomplete information, an accurate view of the problem world as things occur and/or are observed with the passage of time given in a specific *narrative* of interest.

To illustrate the types of problems that will concern us and the general approach that we will follow let us consider a standard example in RAC, that has the name the ‘Car Park Domain’ (Kautz, 1986). This domain when expressed in the language \mathcal{E} contains one action constant *ParkingCar* and one property fluent *CarInParkingSpace* together with the causal law that ‘parking the car causes the car to be in the parking space’. This causal law together with the specific narrative information that we park the car at time 4 and that later at time 8 we observe that the car is not where it was parked, is represented by the following *domain description*:

$$\text{ParkingCar initiates } \text{CarInParkingSpace} \quad (\Delta_1)$$

$$\text{ParkingCar happens-at } 4 \quad (\Delta_2)$$

$$\neg \text{CarInParkingSpace holds-at } 8 \quad (\Delta_3)$$

For domains like this, where a fluent (e.g. *CarInParkingSpace*) changes its truth value without any known causal explanation, language \mathcal{E} does not have a model. Our extended argumentation framework of the language \mathcal{E} that includes arguments for observations and for backwards persistence will allow arguments for both truth values of the fluent within the time interval (4, 8). Forwards persistence from the action *ParkingCar* (Δ_2) that initiates *CarInParkingSpace* for every time point $t > 4$ (Δ_1) comes into conflict with backwards persistence from the observation argument $\neg \text{CarInParkingSpace}$ (Δ_3). Allowing the same priority between conflicting forward persistence and backwards persistence arguments will give the natural interpretation of unknown value for the fluent *CarInParkingSpace* for every $t \in (4, 8)$.

Pictorially, the way our argumentation framework behaves as we update the domain description for the parking domain example, is illustrated in Figure 1 where *KB* denotes the domain description, *PC* the action *ParkingCar* and *CPS* the fluent *CarInParkingSpace*. The arrows in the pictures show the different arguments (for the fluent *CSP* and its negation) that exist in each case.

Suppose now that we extend the parking domain example by adding in the narrative another observation at time 12 of the form:

$$\text{CarInParkingSpace holds at time } 12 \quad (\Delta_4)$$

This will also result in an unknown value for the fluent *CarInParkingSpace* in the interval (8, 12). This arises analogously from conflicting arguments based on the two observation

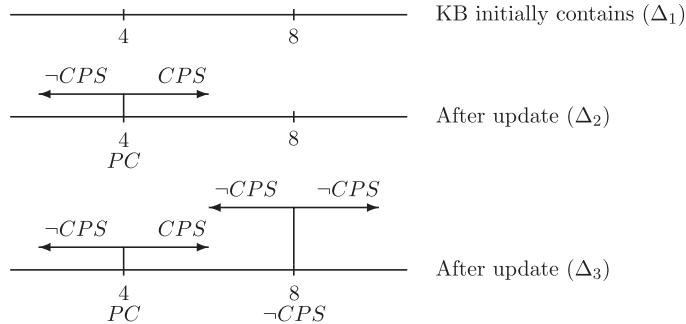


Figure 1. Parking domain.

arguments (Δ_3) and (Δ_4). Forward persistence from $\neg CarInParkingSpace$ (Δ_3) is equally strong to backwards persistence from $CarInParkingSpace$ (Δ_4) and hence we can build equally good (or strong) arguments both for the truth and falsity of the fluent at any time point in this interval. Finally, we note that by giving higher priority to conflicting forward persistence over backwards persistence we will see that our framework can fully recover language \mathcal{E} and hence in this respect it forms a conservative extension of \mathcal{E} .

The argumentation-based formalisation that we will develop in this paper is essentially a preference based (see, e.g. Modgil and Prakken, 2013) realisation of an abstract argumentation framework (Dung, 1995) under the admissibility semantics. Following the approach in Kakas et al. (1999), we will build the preference-based argumentation framework in terms of logic programming rules with a priority relation over these rules. Our work then can be seen as an example where the general theory of argumentation, that has been extensively and widely developed over the past two decades in AI (see, e.g. Bench-Capon and Dunne, 2007; Rahwan and Simari, 2009), finds concrete application in addressing the foundational problems of temporal persistence and knowledge qualification.

This synthesis of ideas opens up possibilities of extending the application of argumentation from ‘static problems’ to variations of these where the problem environment is dynamically changing as new information becomes available. Using argumentation for reasoning about changes in the problem’s world domain offers a principled way to manage the changes in the argumentation framework under which the application problem is expressed, thus extending the use of argumentation from static to ‘dynamic problems’.

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 language \mathcal{E} . Section 4 presents the formal results related to the semantics and the properties of the extended argumentation framework. Section 5 explores further the qualification problem based on this framework. Section 6 discusses related work and Section 7 concludes.

2. A brief review of language \mathcal{E}

Language \mathcal{E} (Kakas and Miller, 1997) is inspired by the EC (Kowalski and Sergot, 1986). As most other action languages, language \mathcal{E} has a set of *action constants*, \mathcal{AC} , that name actions that can occur and a set of *fluent constants*, \mathcal{F} , that name the different properties that can change in the problem domain. *Fluent literals* are either positive or negative fluent constants. It also has a set of *time points*, \mathcal{T} , such that for time points T_1 and T_2 , the notation $T_1 < T_2$ indicates that time point T_1 comes before time point T_2 and $T_1 \neq T_2$.

The language uses three kinds of sentences or propositions defined as follows, 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:

- (1) *c-propositions* (c stands for causes), of the form ‘ A initiates F when C ’ or ‘ A terminates F when C ’,
- (2) *h-propositions* (h stands for happens) of the form ‘ A happens-at T ’,
- (3) *t-propositions* (t stands for time point) of the form ‘ L holds-at T ’.

The first type of sentence is used to express the causal laws of our domain, i.e. how action occurrences generate a property or stop a property from holding. The other two types of sentences are used to describe the particular narratives of interest. The first is used to state what actions have occurred while the latter type of sentences, namely the t-propositions, which will also be called *observations*, are used to represent properties of the world known to hold at particular time points.

A *domain description* D is a set of *t*-, *h*- and *c*-propositions. The *c*-propositions form the background *world knowledge* of the domain and the *t*- and *h*-propositions the *narrative* of the domain.

The semantics of language \mathcal{E} is given in terms of its *models*. These assign a truth value, *true* or *false* to every fluent at every time point in the domain in a way that satisfies a set of constraints which reflect the axiom of *persistence* of the truth value of fluents. The following definitions (taken from Kakas and Miller, 1997) formalize the notion of a model.

DEFINITION 2.1 (interpretation) *Given a domain description D , an interpretation of D is a mapping*

$$H : \mathcal{F} \times \mathcal{T} \mapsto \{\text{true}, \text{false}\}$$

where \mathcal{F} is the set of fluent constants in D and \mathcal{T} is the set of time points in \mathcal{E} . A fluent literal F (resp. $\neg F$) is satisfied at T by H if and only if $H(F, T) = \text{true}$ (resp. $H(F, T) = \text{false}$).

An interpretation may or may not support the initiation or termination of a fluent property depending on whether the preconditions in the causal laws of the domain description are satisfied or not under the interpretation.

DEFINITION 2.2 (initiation-termination point) *Given an interpretation H of a domain description D , a time point T is an initiation (resp. termination) point of a fluent F in H in D when, D contains a combination of a c-proposition ‘ A initiates (resp. terminates) F when C ’ and an h-proposition ‘ A happens-at T ’, such that C at T is satisfied by the interpretation H .*

An interpretation is then a model when it satisfies the property of *persistence*, namely 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) *only* at an initiation (resp. termination) point for that fluent that is supported by some action occurrence and associate causal law under which the action changes the fluent.

DEFINITION 2.3 (model) *Let D be a domain description. An interpretation H is a model of D if and only if for every fluent constant F and time points T_1, T_2, T_3 the following properties hold:*

- (1) *If there is no initiation-point or termination-point T_2 for F in H in D such that $T_1 \leq T_2 < T_3$, then $H(F, T_1) = H(F, T_3)$.*
- (2) *If T_1 is an initiation-point for F in H in D , and there is no termination-point T_2 for F in H in D such that $T_1 \leq T_2 < T_3$, then $H(F, T_3) = \text{true}$.*

- (3) If T_1 is a termination-point for F in H in D , and there is no initiation-point T_2 for F in H in D such that $T_1 \leq T_2 < T_3$, then $H(F, T_3) = \text{false}$.
- (4) For all t -propositions in D of the form ‘ F holds-at T_1 ’, $H(F, T_1) = \text{true}$, and for all t -propositions in D of the form ‘ $\neg F$ holds-at T_2 ’, $H(F, T_2) = \text{false}$.

The property of persistence is expressed by the first three conditions with the second and third conditions also capturing the semantic meaning of the causal laws expressed by the c -propositions. The fourth condition imposes the requirement that models must respect all the observations (t -propositions) of fluents literals contained in the narrative of the given domain description.

The above definition of a model is a slightly extended definition of the original definition in Kakas and Miller (1997) that allows *non-deterministic models* when we have domains where there exist time points that can be simultaneously initiation and termination points for the same fluent. For this we have replaced, in the second and third conditions of the original definition, the restriction on the intermediate time point, T_2 , to be strictly greater than T_1 , i.e. the restriction $T_1 < T_2$, by $T_1 \leq T_2$.

Given the notion of a model, 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. Hence ‘ L holds-at T ’ is *consistent in D* iff there exists a model, M , of D such that $M(F, T) = \text{true}$ when $L = F$ and $M(F, T) = \text{false}$ when $L = \neg F$. Similarly, ‘ L holds-at T ’ is *entailed by D* when the above holds for every model, M , of D .

The following simple examples illustrate the (above variant of) language \mathcal{E} . Let us consider the domain descriptions D , D' and D'' shown in Figure 2, where A and B are action constants and $T_1 < T_2$ are two time points.

Example 2.4 (domain D)

A initiates F

A happens-at T_1

F holds-at T_2

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. Suppose we replace the observation that F holds at T_2 with the observation that it does not hold, i.e. we have the domain D' given by

Example 2.5 (domain D') A initiates F

A happens-at T_1

$\neg F$ holds-at T_2 .

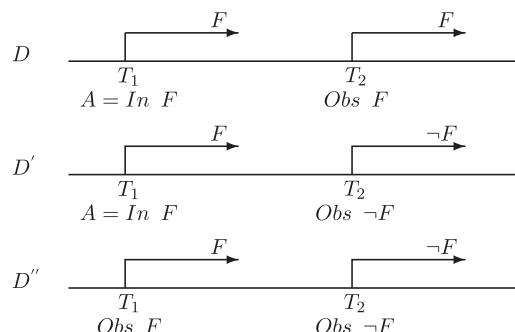


Figure 2. Example domains.

Then this domain, which is similar to the car parking domain discussed in the introduction, does not have any models. The generation of F holding onwards from T_1 cannot be reconciled with the observation of $\neg F$ at T_2 . Similarly, for the domain D'' , given by

Example 2.6 (domain D'')

- F holds-at T_1
- $\neg F$ holds-at T_2

language \mathcal{E} is inconsistent and has no models. The persistence of F holding onwards from T_1 cannot be reconciled with the observation of $\neg F$ at T_2 . Finally, consider the domain D''' given by

Example 2.7 (domain D''')

- A initiates F
- B terminates F
- A happens-at T_1
- B happens-at T_1

We have two possible models for times after time T_1 . In one F holds true for all such time points and in the other F is false for all time points after time T_1 . Hence if we extend this domain with either of the observations that F holds-at T_2 or $\neg F$ holds-at T_2 the extended domain remains consistent.

3. Argumentation formulation

The language \mathcal{E} has been reformulated in terms of argumentation (Kakas et al., 1999) within the preference-based argumentation framework of logic programs with priorities, *LPwNF*, under its admissibility semantics (Dimopoulos and Kakas, 1995; Kakas, Mancarella, and Dung, 1994). In this the information from t-propositions (observations) is not used in the argumentation process but rather these observations are 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 introducing new arguments that are rooted or based on observations. In addition, we will use backward temporal persistence arguments as well as forward ones to allow us to address the qualification problem.

To translate a given domain description D into an argumentation program in *LPwNF*, all individual h - and c -propositions translations are included in the background theory as follows. We will consider time to be discrete scalar.

DEFINITION 3.1 (background theory for D) *The background theory for D is the theory $B(D)$ given by:*

- for all time points T and T' and action constants A :
 - $B(D) \vdash T \leq T'$ if and only if $T \leq T'$,
 - $B(D) \vdash T < T'$ if and only if $T < T'$,
 - $HappensAt(A, T) \in B(D)$ if and only if ‘ A happens-at T ’ is in D , and
 - $Observation(L, T) \in B(D)$ if and only if ‘ L holds-at T ’ is in D .
- for each c -proposition ‘ A initiates F when $\{L_1, \dots, L_n\}$ ’ in D , $B(D)$ contains the rule
 $Initiation(F, t) \leftarrow HappensAt(A, t), \Lambda(L_1), \dots, \Lambda(L_n)$, and

- for each c-proposition ‘A terminates F when $\{L_1, \dots, L_n\}$ ’ in D, $B(D)$ contains the rule
 $\text{Termination}(F, t) \leftarrow \text{HappensAt}(A, t), \Delta(L_1), \dots, \Delta(L_n)$
where $\Delta(L_i) = \text{HoldsAt}(F_i, t)$ if $L_i = F_i$, and $\Delta(L_i) = \neg \text{HoldsAt}(F_i, t)$ if $L_i = \neg F_i$, for any fluent constant F_i ,
- $B(D)$ contains no other rules.

The above definition is essentially the same as in Kakas et al. (1999) with the exception that the given observations are also added in $B(D)$ and hence the whole narrative forms part of the background knowledge. Any such background knowledge is then extended by the following logic program rules and priorities between these to give the full *argumentation logic program with priorities* corresponding to a given domain description.

DEFINITION 3.2 (argumentation program of D) *The argumentation program corresponding to a domain D is $AD \equiv (B(D), \mathcal{A}, <)$ where:*

- $B(D)$, is the corresponding background knowledge of D.
- The set, \mathcal{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.*¹

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

Local Generation Arguments:

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

Local Observation Arguments:

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

Initial Assumptions:

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

- The priority (or strength of argument) relation, $<$, between these arguments is given as follows. Let t, t^*, t_1 and t_2 be time points:

Set 1: If $t_1 < t_2$

$$\begin{aligned} PFP[f, t^*; t_1] &< NFP[f, t^*; t_2] \text{ and } NFP[f, t^*; t_1] < PFP[f, t^*; t_2], \\ PBP[f, t^*; t_2] &< NBP[f, t^*; t_1] \text{ and } NBP[f, t^*; t_2] < PBP[f, t^*; t_1]. \end{aligned}$$

Set 2: If $t_1 \leq t_2$

$$NFP[f, t_2 + 1; t_1] < PG_F[f, t_2] \text{ and } PFP[f, t_2 + 1; t_1] < NG_F[f, t_2].$$

Set 3 At 0,

$$PA[f, 0] < NO[f, 0] \text{ and } NA[f, 0] < PO[f, 0].$$

Set 4: For any t,

$$NG_B[f, t] < PO[f, t] \text{ and } PG_B[f, t] < NO[f, t].$$

$$PG_F[f, t] < NO[f, t + 1] \text{ and } NG_F[f, t] < PO[f, t + 1].$$

Set 5: If $t_1 < t_2$

$$NFP[f, t_2; t_1] < PO[f, t_2] \text{ and } PFP[f, t_2; t_1] < NO[f, t_2],$$

$$NBP[f, t_1; t_2] < PO[f, t_1] \text{ and } PBP[f, t_1; t_2] < NO[f, t_1].$$

This definition introduces four different types of arguments and the relative strength between them. These types of arguments express in a natural way the defeasible conclusions that we can draw or support from the non-defeasible information in $B(D)$.

Let us explain the various parts of this definition. Consider first the various arguments rules.

- *Persistence rules* : These rules express the possibility of fluent properties persisting forward and or backward in time. The first rule captures the forward persistence from a property holding at t to holding at a latter point t_2 , while the second rule captures the backward persistence from a property holding at t to holding at an earlier point t_1 . The other two rules are similar but for the persistence of properties not holding.
- *Local generation arguments* : These are argument rules for the causation of a property by an event. The first rule says that the effect of an initiation point at time t starts at the very next time point, while the second rules say that from this initiation point we have an argument that its effect does not hold at the time of the event that generates the property. Analogously for the following two rules based on termination points.
- *Local observation arguments* : Observations give directly arguments for fluents holding or not at the time of observation.
- *Initial assumptions* : Assumptions can only be introduced at time 0 and can be for a property holding or not holding.

Let us now comment on the *priority relation*. We first note that the priority relation, $<$, in an argumentation program is *independent* of the domain description, D . It is a universal relation that will us capture the general principles of persistence and qualification.

- *Set 1*: The first set of priorities in the above definition make forward persistence arguments that are based on a later time point stronger than conflicting forward persistence arguments that are based on an earlier time point. Analogously, backward persistence arguments that are based on earlier time points are stronger than conflicting backward persistence arguments that are based on later time points.
- *Set 2*: This set of priorities expresses the fact that the generation of effects is stronger than opposing persistence.
- *Set 3–Set 5*: The final three sets of priorities give, as expected, higher strength to observation arguments over any other type of conflicting arguments.

We note that included in the types of arguments are the two local generation arguments, $NG_B[f, t]$ and $PG_B[f, t]$, which are not usually found in formulations of many RAC frameworks, with a notable exception of the original EC. It is important though to stress that these arguments only support the possibility for the occurrence of events and their initiation or termination of effects to affect the past. The argumentation formulation allows us to capture this possibility in a weak form, by saying that there exists an argument for something holding in the past due to the occurrence of events. Then depending on the priority we give to this argument over other arguments, its effect on the semantics would vary. For example, as we will see below if we assign forward persistence arguments to be stronger than these local generation arguments (or more generally stronger than backwards persistence arguments) then these arguments do not have any effect on the semantics, i.e. no maximal admissible extension can contain these local generation arguments.

In the priority relation 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. Finally, we note that the assumption arguments are not chosen to be weaker

than the conflicting backwards persistence arguments to allow the freedom of incomplete initial states.

The semantics of these *LPwNF* argumentation programs is given through the standard argumentation notion (see Dung, 1995; Kakas et al., 1994) of *maximally admissible* subsets of the given argumentation program, called *admissible extensions*, as follows.

DEFINITION 3.3 (background logic) *The background monotonic logic is the tuple (\mathcal{L}, \vdash) , where the language \mathcal{L} consists of all ground rules of the form $l_0 \leftarrow l_1, l_2, \dots, l_n$ ($n \geq 0$) that belong to $B(D)$ or \mathcal{A} with each l_i a classical literal. l_0 is called the head of the rule and l_1, l_2, \dots, l_n is called the body of the rule. \vdash is obtained by the repeated application of the modus ponens rule.*

The above definition expresses formally the logical reasoning that allows the link of the background knowledge with the argument rules so that arguments supporting a fluent literal conclusion can be constructed. These arguments are grounded on the narrative given in the domain description D and translated in the corresponding background knowledge $B(D)$.

The attacking relation is then defined amongst sets of argument rules that have conflicting conclusions by taking into account the priorities on the argument rules as given in Definition 3.2.

DEFINITION 3.4 (attack) *Let G, W be two non empty sets of argument rules. G attacks W if and only if there exists a literal l of the form $HoldsAt(f, t)$ or $\neg HoldsAt(f, t)$ and sets $G_1 \subseteq G$ and $W_1 \subseteq W$ such that:*

- (i) $B(D) \cup G_1 \vdash_{\min} l$ and $B(D) \cup W_1 \vdash_{\min} \neg l$
- (ii) if there exist $r' \in G_1, r \in W_1$ such that $r' < r$ then there exist $r' \in G_1, r \in W_1$ such that $r < r'$.

Hence the attacking relation is defined by lifting the priorities from the level of single argument rules to sets of these. For a set G of argument rules to attack another set W it must contain at least one rule of higher priority than a rule in the attacked set W or otherwise the set W must (also) not contain any rule of higher priority than some rule in the attacking set G . In this way we ensure that the attacking set is not ‘weaker’ than the attacked set as either it contains a stronger rule or the two sets are non-comparable. Note that if a set is inconsistent, i.e. it derives both a literal and its negation then it attacks itself as the second condition in the definition of the attacking relation is trivially satisfied.

The following definition introduces some technical conditions on sets of argument rules that are more specific to the use of argumentation to the particular application of RAC.

DEFINITION 3.5 (closed, complete and compact) *A set Δ of argument rules is closed if all the bodies of its rules are derived via \vdash from the background theory $B(D)$ extended by Δ . Δ is complete if for any fluent f and time point t , $\Delta \vdash HoldsAt(f, t)$ or $\Delta \vdash \neg HoldsAt(f, t)$. Finally, Δ is compact if Δ is closed and does not contain a pair of argument rules, $PFP[f, t; _]$ and $PBP[f, t; _]$ or $NFP[f, t; _]$ and $NBP[f, t; _]$, for any fluent f and any time point t .*

The condition of closeness is a technical condition to avoid considering unnecessarily argument sets that contain rules which cannot be used to support conclusions. Similarly, the compactness property imposes a uniqueness of support for the conclusions drawn through persistence arguments requiring that any such conclusion (timed fluent literal) is supported in a uni-directional way, i.e. either by a forwards persistence or by backwards persistence but not (redundantly) by both such arguments. These technical conditions are conditions that

avoid redundancy and help to simplify the framework and the proofs of its formal properties, particularly the link with the model theory of language \mathcal{E} when the two approaches coincide.

Finally, the property of completeness is a desired property for the semantics of the approach so that the admissible extensions, that will define the semantics, are able to decide for any fluent at any time point its status, i.e. they can populate the whole time line with a decision on whether any fluent holds or not. This condition is also motivated by the technical consideration of relating the semantics to the model theory semantics of the language \mathcal{E} .

DEFINITION 3.6 (admissibility) *Let D be a domain description and $AD \equiv (B(D), \mathcal{A}, <)$ its corresponding argumentation program. Let S be a closed subset of \mathcal{A} . Then S is admissible if and only if:*

- (i) $B(D) \cup S$ does not derive a literal l and its complement $\neg l$ and
- (ii) for any $S' \subseteq \mathcal{A}$ if S' attacks S then S attacks S' .

This is the standard definition of admissibility. A subset of arguments is admissible if it does not derive both $HoldsAt(f, t)$ and $\neg HoldsAt(f, t)$ for any fluent and time point and therefore it does not attack itself and it can counter-attack any subset of arguments that attacks it.

The semantics of domain descriptions is then given by putting these notions together to define admissible extensions of the corresponding argumentation program.

DEFINITION 3.7 (admissible extension) *Let D be a domain description and $AD \equiv (B(D), \mathcal{A}, <)$ its corresponding argumentation program. Let S be a closed subset of \mathcal{A} . Then $\Delta = B(D) \cup S$ is an admissible extension of AD if and only if S is a complete and compact admissible set.*

We note that alternatively we could have omitted the explicit requirement of completeness and replaced this with the maximality condition as we normally have with the preferred semantics of argumentation. Nevertheless, remaining within the spirit of RAC and as complete extensions are maximal it is natural to define the semantics only in terms of the complete admissible extensions provided that complete extensions exist, which as we will see in the next section, is indeed the case. Complete admissible extensions then correspond more closely to the stable semantics of argumentation as they cannot be extended consistently (i.e. without self-conflict) by any other subset of argument rules that support a conclusion that is not already supported by the complete extension.

Conclusions from an admissible extension are then drawn using the derivability relation, \vdash , given in Definition 3.3, with *credulous conclusions* those that can be derived from at least one admissible extension and *sceptical conclusions* as those that are derived in every admissible extension.

To illustrate the above definition of the argumentation semantics let us consider our earlier example domains D (Example 2.4), D' (Example 2.5) and D'' (Example 2.6) within the argumentation framework defined above (Figure 3). In domain D , for all $T > T_1$ the strongest (and hence admissible) argument is for F to hold through a local generation argument at $T_1 + 1$ and then by forwards persistence arguments at any other time after T_1 . Otherwise, from T_2 onwards we can use the observation argument at T_2 and forwards persistence arguments from this to support strongly F for times greater than T_2 . 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. Note also that the backwards persistence arguments for F from T_2 to some time point equal to or before T_1 would not be stronger than the forwards persistence of $\neg F$ starting from time 0 through an assumption argument for $\neg F$. Nor would it be stronger than the backwards negative generation argument for $\neg F$ based on the initiation point T_1 for F .

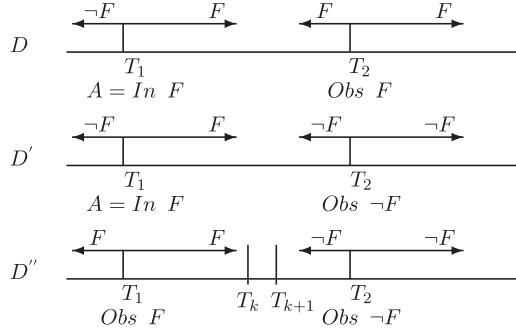


Figure 3. Example domains and arguments.

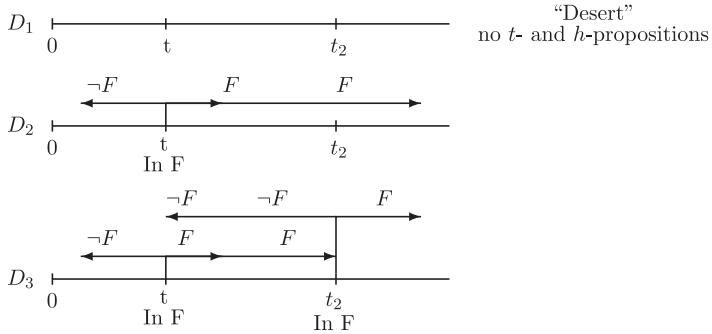


Figure 4. Examples.

For the domain D'' , the strongest argument for all time points $T \geq T_2$ is the observation argument for $\neg F$ at T_2 combined with a forward persistence argument from T_2 to any such time point T . For times between T_1 and T_2 , we have admissible arguments for either F or $\neg F$: at some time point T_k , $T_1 \leq T_k < T_2$, 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 through some exogenous qualification.

The examples that follow illustrate further the argumentation formulation. Let f be any fluent and t, T time points:

Example 3.8 Let D_1 be a domain with no h -propositions and t -propositions. The only possible admissible extensions is for f (resp. $\neg f$) to hold for all times t as there does not exist a stronger argument that attack the set $S = \{PA[f, 0], PFP[f, t; 0]\}$, for all $t > 0$. Possible attacks on S contain the argument $NA[f, 0]$ and S counterattacks it through $PA[f, 0]$. Admissible extensions in this example domain are the same as the (maximally) admissible extensions of language \mathcal{E} .

Example 3.9 Let D_2 be a domain with no t -propositions containing h -propositions so that it necessarily has two initiation points of f at t_1, t_2 such that $t_2 > t_1$. In this case, language \mathcal{E} models require f to hold for all $T > t_1$. In our extended argumentation framework, f will hold for all $T > t_2$ in all admissible extensions. For time points $T' \in (t_1, t_2]$ we can build an argument for f , $(PG_F[f, t_1], PFP[f, T'; t_1 + 1])$ and an argument for its negation $\neg f$, $(NG_B[f, t_2], NBP[f, T'; t_2])$

which are both admissible. Hence between the two initiation points we cannot conclude that f is entailed, like language \mathcal{E} , as f does not hold in all admissible extensions. We will see that our framework is closer to the original EC in this respect. Also we will see that if we wish to fully recover language \mathcal{E} we could do this by assigning higher priority to forward persistence arguments over conflicting backwards persistence arguments.

Note that if we additionally have in the domain D_2 a termination point for f at time $T'' \in (t_1, t_2)$ then the second type of argument sets that derives $\neg f$ at points in between the two initiation points before T'' would not be admissible anymore as it would be attacked by the stronger backward persistence argument for f from T'' , namely from $(PG_B[f, T''], PBP[f, T'; T''])$. This can be understood as an indication that the given narrative in D_2 has some missing information between the time points t_1 and t_2 of the two initiation points.

4. Formal results

In this section we present a set of formal results that show the various properties that the argumentation formulation given above has with respect to persistence and qualification, how this formulation gives meaning to any domain description theory and how it relates to and extends the original model theoretic formulation of language \mathcal{E} .

Firstly, as expected, we show that admissible extensions are consistent with the observations in the given narrative. From now on we will consider only *point-wise consistent* domain descriptions, i.e. domain descriptions that do not contain any pair of t -propositions of the form ‘ f holds-at t ’ and ‘ $\neg f$ holds-at t ’.

PROPOSITION 4.1 Let D be a point-wise consistent domain description and E an admissible extension of the corresponding argumentation program $AD \equiv (B(D), \mathcal{A}, <)$. Then E is consistent with D , i.e. there does not exist a t -proposition, ‘ f holds-at t ’ (resp. ‘ $\neg f$ holds-at t ’) in D such that $E \vdash \neg \text{HoldsAt}(f, t)$ (resp. $E \vdash \text{HoldsAt}(f, t)$). ■

Proof see Appendix 2. ■

This is a simple basic result that is needed for the main result of showing the existence of admissible extensions for any description domain and for linking admissible extensions with the models of language \mathcal{E} , whenever such models exist. Indeed, it corresponds to the fourth condition in the definition of a model of language \mathcal{E} (see Definition 2.3).

The next result shows that the admissible extensions satisfy the important property of *persistence of derivation* of fluent properties along the time line. Informally, it says that if between two time points there is no information, either causal or observational in nature, that a fluent has changed value then an admissible extension will derive a constant value for the fluent in this time interval.

LEMMA 4.2 Let D be a point-wise consistent domain description and E an admissible extension of D . Let f be a fluent and $t_n < t_m$ two time points such that there does not exist an initiation point nor a termination point for the fluent f in E^3 at any time $t_1 \in [t_n, t_m)$ nor there exists an observation point for the fluent f in E at any time $t_2 \in (t_n, t_m]$. Then if $E \vdash \text{HoldsAt}(f, t_n)$ and $E \vdash \text{HoldsAt}(f, t_m)$ hold or (respectively, $E \vdash \neg \text{HoldsAt}(f, t_n)$ and $E \vdash \neg \text{HoldsAt}(f, t_m)$ hold) then $E \vdash \text{HoldsAt}(f, t)$ (respectively, $E \vdash \neg \text{HoldsAt}(f, t)$) holds, for every $t \in [t_n, t_m]$. ■

Proof see Appendix 1. ■

Therefore the notion of admissibility in argumentation helps us to capture the requirement of persistence. Admissible extensions must have this property of persistence as otherwise they will have attacks that they cannot counter-attack. This plays an important role in showing the next theorem which gives the central result of the argumentation formulation, namely that any domain description D (with a countable narrative) is always consistent, i.e. there always exists an admissible extension of the corresponding argumentation program of D .

THEOREM 4.3 Let D be a point-wise consistent domain description with time structure that of the natural numbers and a countable number of h - and t -propositions. Then there exists an admissible extension E of the corresponding argumentation program $AD \equiv (B(D), \mathcal{A}, <)$.

Proof see Appendix 3. ■

Hence any domain description D is given a meaning under the above argumentation semantics. For example, as we have seen above, the example domains D' (Example 2.5) and D'' (Example 2.6) are consistent in our extended framework. These domains do not have any model in the language \mathcal{E} or an admissible extension in the earlier argumentation-based reformulation of \mathcal{E} in Kakas et al. (1999).

The formal link of our argumentation formulation of the language \mathcal{E} with its original model theoretic semantics is given by the following theorem.

THEOREM 4.4 Let D be a point-wise consistent domain description with time structure that of the natural numbers 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 $AD \equiv (B(D), \mathcal{A}, <)$ such that E corresponds to M , i.e. $E \vdash \text{HoldsAt}(f, T)$ if and only if $M(f, T) = \text{true}$ and $E \vdash \neg \text{HoldsAt}(f, T)$ if and only if $M(f, T) = \text{false}$.

Proof see Appendix 3. ■

This theorem shows that when language \mathcal{E} models exist then our argumentation framework always has a corresponding admissible extension. Our formulation therefore forms a conservative extension of \mathcal{E} .

The next theorem provides an interpretation of how the extended argumentation formulation handles the qualification problem when it provides a semantics to any domain description. It shows that an admissible extension E of any domain D can be interpreted as a language \mathcal{E} model of D or of D extended by some h -propositions of unknown events.

THEOREM 4.5 Let D be a point-wise consistent domain description with a finite number of h -propositions and t -propositions. For every admissible extension E of D there exists a domain D' obtained from D by adding a (possibly empty) set of new h -propositions such that there exist a language \mathcal{E} model, M , of D' that corresponds to E (i.e. $E \vdash \text{HoldsAt}(f, t)$ (resp. $E \vdash \neg \text{HoldsAt}(f, t)$) if and only if $M(f, t) = \text{true}$ (resp. $M(f, t) = \text{false}$)).

Proof see Appendix 3. ■

For example consider the domain D'' (Example 2.6) where we observe F at time T_1 and $\neg F$ at time T_2 . Then given any time point $T_k \in (T_1, T_2)$ this domain has admissible extensions E where F is derived for any time point between T_1 up to T_k and $\neg F$ is derived for all time points after T_k up to T_2 . The domain D'' does not have any language \mathcal{E} models. But by interpreting time T_k as a termination point for F through an unknown (in D) new action that can terminate F we can ‘explain’ within \mathcal{E} the meaning given to D'' by the above admissible extensions in the extended argumentation formulation of the language.

5. Qualification extensions

In this section we study in more detail how the argumentation semantics that we have provided for action theories addresses the qualification problem. In particular, we will examine how it helps us understand qualifications of action occurrences and how we can find, within the given domain description, explanations for exogenous qualifications.

The argumentation formulation allows us to identify exogenous qualifications imposed by the lack of information in the given narrative of the domain description.

DEFINITION 5.1 *Let D be a domain description and E an admissible extension of D such that there exist time points t_1, t_2 with $t_1 < t_2$ and a fluent F so that the following conditions hold:*

- The t -proposition ‘ $\neg F$ holds-at t_2 ’ (respectively ‘ F holds-at t_2 ’) belongs to D and there is no t -proposition ‘ F holds-at t ’ (respectively ‘ $\neg F$ holds-at t ’) in D for any t such that $t_1 < t < t_2$.
- There exists an h -proposition ‘ A happens-at t_1 ’ and a c -proposition ‘ A initiates F when C ’ (respectively ‘ A terminates F when C ’) in D such that t_1 is an initiation (respectively termination) point for F in E , and there is no other initiation (respectively termination) point t for F in E such that $t_1 < t < t_2$.⁴
- There exists no termination point for F at t in E such that $t_1 \leq t < t_2$.

Then E is called an exogenous qualification extension on F at $(t_1, t_2]$ of D . The exogenous qualification is said to be based on the proposition ‘ $\neg F$ holds-at t_2 ’ (respectively ‘ F holds-at t_2 ’).

Furthermore, if there does not exist a time point $t \in (t_1, t_2)$ such that $E \vdash \text{HoldsAt}(F, t)$ (respectively $E \vdash \neg \text{HoldsAt}(F, t)$) then E is called an exogenous qualification of the action occurrence of A at t_1 . Otherwise, E is an exogenous qualification of the persistence of F in $(t_1, t_2]$.

Qualification extensions are therefore separated into two types. In the first case, the action occurrence of A at t_1 is exogenously qualified failing to generate its effect and consequently in the whole of the interval, $(t_1, t_2]$, $\neg \text{HoldsAt}(F, t)$ (respectively $\text{HoldsAt}(F, t)$) holds in E . In the second case, the action occurrence of A has succeeded to initiate its effect but its persistence was terminated before t_2 by some unknown exogenous event.

The above definition of exogenous qualification captures instances of *direct qualification* in the sense that the qualification is linked directly to the fluent through which the disparity between what is observed and what is described by the domain and its semantic extensions, manifests itself. The domain description implies that F (respectively $\neg F$) should hold at t_2 but the opposite, i.e. $\neg F$ (respectively F), is observed at t_2 . In the first case of an exogenous qualification of an action, the failure of the action to produce its effect could be due to some exogenous reason for (at least) one of the preconditions of the c -proposition through which F would be initiated (respectively terminated), not to hold at the time t_1 of the action occurrence. For example, consider the following domain description, also depicted in Figure 5.

Example 5.2 (domain QD_1)

A initiates F when L

A happens-at T_1

L holds-at T_0

$\neg F$ holds-at T_2

All the admissible extensions of this domain are exogenous qualification extensions on F . In particular, one of these extensions is an exogenous qualification of the action occurrence of A at



Figure 5. Domain QD_1 .

T_1 . But the failure of the action A to produce its effect F might be due to some other exogenous reason for the precondition L , of the c -proposition that initiates F , not to hold at the time of occurrence of A .

We can thus translate or *explain the exogenous qualification* of the action A at T_1 *endogenously* by the hypothesis that $\neg L$ holds-at T_1 and transfer the exogenous qualification of the domain to the earlier persistence of L from T_0 to T_1 . Indeed, if we add in QD_1 this t -proposition then the extended domain description does not anymore have an exogenous qualification extension for F between T_1 and T_2 . Such extensions of a given domain description with additional t -propositions so that the exogenous qualification of actions is pushed back into the past in terms of exogenous qualifications of preconditions of c -propositions will be called qualification explanation domains and are defined as follows.

DEFINITION 5.3 *Given a domain description D and E an exogenous qualification extension of an action occurrence A on F at t_1 a qualification explanation domain for A at t_1 is a domain D' obtained from D by adding a set H of t -propositions on the narrative time points of D such that*

- H is point-wise consistent with the t -propositions already in D and H does not contain a t -proposition of the form ‘ F holds-at t ’ for $t > t_1$.
- E is no longer an exogenous qualification extension of the action occurrence A at t_1 in D' in the sense that starting from the same initial state as in E there is no admissible exogenous qualification extension of the action occurrence A at t_1 in D' .

H is called a qualification explanation (with respect to E) (or the initial state of E) of the exogenous failure of action A at t_1 .

In addition, as is usual with abductive explanations, we can require that qualification explanations are minimal with respect to set inclusion.

Hence in the previous Example 5.2, $H = \{\neg L \text{ holds-at } T_1\}$, is a qualification explanation of the exogenous failure of action A at T_1 . Suppose now that we extend the previous Example 5.2 as follows:

Example 5.4 (domain QD_2)

A initiates F when L

A happens-at T_1

B initiates L when K

B' terminates L when K'

B happens-at T'

$\neg F$ holds-at T_2

where $T_0 < T' < T_1 < T_2$. Depending on the initial value of the fluents K and L in an admissible extension, i.e. whether a positive or a negative assumption argument at time T_0 for these fluent is included in an extension, this will determine whether the extension is an exogenous qualification of the action occurrence of A at T_1 . This can happen in two cases (a) ‘ L holds-at T_0 ’ or (b) ‘ K holds-at T_0 ’ under which the action occurrence of A at T_1 gives a generation (initiation) argument for F . Figure 6 depicts this example and three possible qualification explanations, H, H', H'' , for the exogenous qualification of the action occurrence of A at T_1 , as will be discussed below.

Hence under either of these cases, as in the previous example, a qualification explanation for the exogenous qualification of the action occurrence of A at T_1 is given by ‘ $\neg L$ holds-at T_1 ’. The extended domain obtained from QD_2 by adding this t -proposition does not anymore have an exogenous qualification extension of A at T_1 because the potential initiation argument of F at T_1 cannot be applied. But this extended domain, in the case where we have ‘ K holds-at T_0 ’, will now have an exogenous qualification extension for L at T' based on the new t -proposition that we have now added to the domain⁵ (see Figure 7).

We can then explain this new exogenous qualification by adding the extra t -proposition, ‘ $\neg K$ holds-at T' ’, to the extended domain. This pushes the exogenous qualification back to the persistence of K from T_0 to T' .

Then if also the initial state is such that $\neg L$ is derived at T_0 , e.g. when ‘ $\neg L$ holds-at T_0 ’ is in the domain, there will be no exogenous qualification of A at T_1 in the domain extended by both ‘ $\neg L$ holds-at T_1 ’ and ‘ $\neg K$ holds-at T' ’ (see Figure 8). Note that Figure 8 shows ‘ $\neg L$ holds-at T_1 ’ as an additional observation.

Furthermore, the new t -proposition, ‘ $\neg K$ holds-at T' ’, is sufficient alone, without the need for the earlier explanation ‘ $\neg L$ holds-at T_1 ’, to explain the exogenous qualification of A at T_1 . In other words, starting from the explanation $H = \{\neg L\}$ for the exogenous qualification of A at T_1 we can replace (one of) its element(s), i.e. ‘ $\neg L$ holds-at T_1 ’, by an explanation, $H' = \{\neg K\}$

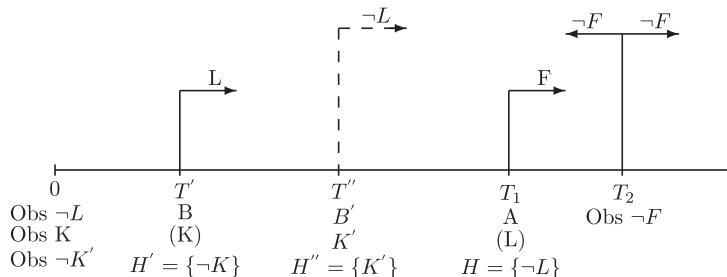


Figure 6. Qualification explanations for domain QD_2 .

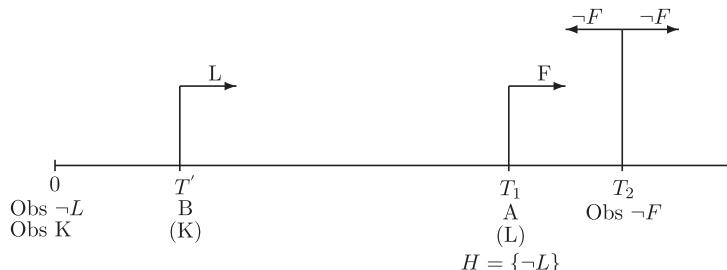


Figure 7. Qualification explanation H for domain QD_2 .

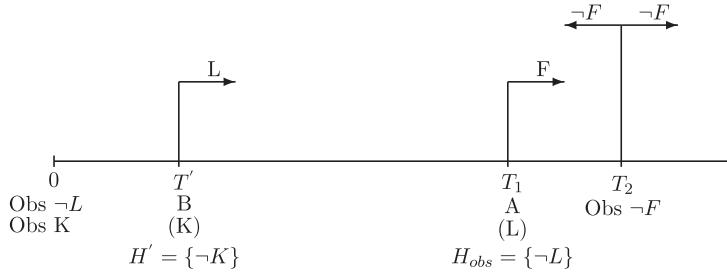


Figure 8. Qualification explanation H' for domain QD_2 .

holds-at T' }, of the exogenous qualification of B at T' based on this element, resulting in a new (minimal) explanation of the exogenous qualification of A at T_1 .

In general, there are two types of qualification explanation domains: ones that block a generation point as we have seen in the examples above and explanations that activate an intermediate generation point to block persistence. Let us consider an example of the latter type. In the example domain QD_2 another explanation of the exogenous qualification of A at T_1 would be possible if we had in the domain a known occurrence of the action B' at some time point $T'' \in (T', T_1)$ (shown in Figure 6 with a dotted line). So let us assume that the domain QC_2 also contains the h -proposition B' happens-at T'' . Then $H'' = \{K' \text{ holds-at } T''\}$ is a qualification explanation of the exogenous qualification of B at T' as it enables a termination point for L at T'' . In turn this means that H'' is also a qualification explanation of the exogenous qualification of A at T_1 . In other words, again starting from the explanation $H = \{\neg L \text{ holds-at } T_1\}$ for the exogenous qualification of A at T_1 we can generate a new explanation H'' for this by finding an explanation for an earlier action that is exogenous qualified by the addition of H in the domain.

Hence as we have seen in the example above, qualification explanations can be built iteratively by pushing the exogenous qualification they explain to an earlier action and replacing (elements of) the initial explanation by an explanation of the exogenous qualification of the earlier action. The following property linking the qualification explanations of actions holds in general.

Property: Let D be a domain, E an exogenous qualification extension of an action occurrence A at t_1 and H a qualification explanation of A with respect to E . Let also ‘ L holds-at T ’ belong to H and H' be a qualification explanation with respect to E of an action occurrence A' at T_1 ($T_1 < T$) based on L holds-at T . Then given $H'' = (H - \{L \text{ holds-at } T\}) \cup H'$ one of the following holds:

- H'' is also a qualification explanation of A at t_1 with respect to E
- or $D \cup H''$ has an admissible extension E'' with the same initial state as that of E such that E'' is an exogenous qualification extension of an action occurrence A'' at T_2 with $T_2 < T_1$.
- or $E \vdash \neg \text{HoldsAt}(L, 0)$ and the persistence of the fluent literal $\neg L$ from 0 to T is exogenously qualified in $D \cup H$ with respect to E .

Hence as we have seen above in the example domain QD_2 $H = \{\neg L \text{ holds-at } T_1\}$ can be replaced by $H'' = \{\neg K \text{ holds-at } T'\}$ to either give a new qualification explanation of A at T_1 in the case where $E \vdash \neg \text{HoldsAt}(L, 0)$ or when $E \vdash \text{HoldsAt}(L, 0)$ the persistence of L from 0 to T_1 is exogenously qualified in $D \cup H$.

Based on this property we can then develop algorithms to find qualification explanations of action failures by systematically following backwards the time line from the time of an unexpected observation, such as ‘ $\neg F$ holds-at T_2 ’ in the above examples, and examining the preconditions of the possible initiation and termination points for F . These explanations can be further compared according to some criterium, e.g. that of minimality of the number of actions

that are exogenously qualified. This issue of the minimisation of exogenous qualification (and its argumentation interpretation) is beyond the scope of this paper. It is important though to note that the computation of qualification explanations is made possible by the generality of the argumentation semantics and its ability to give meaning to any domain description.

6. Related work

Our work follows the EC approach for RAC and the realisation of this through argumentation. It extends the earlier argumentation-based approach for the EC language \mathcal{E} (Kakas and Miller, 1997) by allowing the observations in a domain description to be used directly in the argumentation process rather than to be imposed as a posteriori constraints. As a result any domain description in \mathcal{E} can be given a meaning under the extended argumentation framework.

Interestingly, our formulation comes closer to the original EC (Kowalski and Sergot, 1986). Like the original EC it has a symmetrical treatment of forward and backward reasoning from events, unlike the later formulations of the simplified EC (Miller and Shanahan, 1999; Shanahan, 1997) where the emphasis is mainly on reasoning forward from events. The original EC is based upon the notion of events and time periods generated by the events. There are two types of time periods, *after(e u)* and *before(e u)* generated by an event *e* and during which a fluent *u* holds. The first of these names a time period after the occurrence of the event *e* while the second names a time period before the occurrence of the event *e*. The ends of these periods is initially undefined until extra information that can help us derive this is provided. These time periods are derived through the rules in the EC of the following form:

Holds(after(e u)) if Initiates(e u)

Holds(before(e u)) if Terminates(e u)

Such rules are analogous to the arguments $\{PG_F[u, e], PFP[u, t; e]\}$, where *t* is any time point after the time of the event *e* and $NG_B[u, e]$ and $NBP[u, t; e]$, where *t* is any time point before or equal to the time of the event *e*.

This symmetrical nature of reasoning with time in the EC is mirrored with the symmetry of forward and backwards persistence arguments in our formulation. Hence both formalisms have similar effects especially when reasoning in the past. To illustrate this consider the simple example:

Example 6.1 (domain D''')

A initiates F

B initiates F

A happens-at T₁

B happens-at T₂

In this domain the original EC concludes that there exists an end point *i* of the time period *after(A F)* such that $A < i \leq B$. After *i* the value of *F* is unknown. The effect of this is that the value of *F* cannot be concluded at any time point between the time of *A*, i.e. *T₁*, and the time of *B*, i.e. *T₂* (as the exact time of *i* is unknown). Similarly, in our formulation there exists admissible extensions of this domain such that for any time point $T_i \in (T_1, T_2)$ the fluent *F* changes its truth value from true to false and hence the value of *F* within this time period cannot be sceptically determined.

Hence both formulations essentially conclude the existence of at least one unknown event at some intermediate time point *T_i* between *T₁* and *T₂* that has terminated *F* and hence we cannot decide sceptically on *F* between *T₁* and *T₂*. Through different mechanisms, our argumentation and the original EC give the same end result in terms of properties holding. We also note that both

formalisms allow that in these situations of ‘uncertainty’ between two time points, the possibility of several changes in the value of the fluent.

The above example may be a little specialised. The more natural case of this effect of incorporating backwards persistence is seen in Examples 2.5 and 2.6 where observations are involved. The observations indicate a change in the value of a fluent at some unknown time within a certain time period. This is captured by our formulation thus showing how the original EC could be extended to allow observations in its domain descriptions.

Another approach to RAC using argumentation is that of Vo and Foo (2005). This work addresses the main problems of RAC, in particular the frame, and qualification problems, using a formulation that is based on argumentation and in particular the ABA argumentation framework of Bondarenko et al. (1997). As in Bondarenko et al. (1997), it uses argumentation assumptions that complement the logical knowledge to draw inferences from a domain description. There are two types of assumptions associated with each fluent literal: *frame* and *qualification* assumptions. The frame assumptions allow the persistence of a fluent literal as these assumptions can be assumed at any time point. Unlike our approach which relies directly on argumentation to capture persistence through the use of persistence arguments, the effect of persistence in Vo and Foo (2005) is obtained indirectly by the freedom of their formalism to make these frame assumptions. This though imposes the need to control this freedom by additional requirements imposed on top of a first structure they obtain via the argumentation requirement of preferred frame assumptions, called pre-models, in order to narrow down the possibilities of frame assumptions. This extra requirement is outside the standard argumentation notions imposing a ‘minimisation of change’ on the pre-models.

Subsequently, in order to handle also the qualification problem another type of assumptions is introduced, the qualification assumptions. These are ‘externally’ linked with each of the causal rules that generate fluent literals and are thought off as encompassing all the exogenous (unknown) qualification conditions that can block the generation of the fluent literal through the causal law. Then again a further technical machinery is developed on top of this with notions such as, plausible, unsound and rejected sets of assumptions, which need to be imposed on the initial pre-models, i.e. outside the argumentation formulation. In this filtering it is also necessary to impose carefully a preference of qualification assumptions over frame assumptions in order to solve both the frame and qualification problems together.

Hence although the two approaches have similarities as they are both based on argumentation they have crucial differences which can have an effect both at the representation and computational level. For example, the assumption arguments in Vo and Foo (2005) are of various types and are operational in nature rather than expressing some declarative knowledge in the domain. In contrast, our approach works directly on the given declarative knowledge of the problem: assumptions can only be made at the initial time point and importantly they are declarative statements.

Furthermore, our approach addresses the problem of RAC entirely within standard argumentation notions with no reliance on any other formal structure. One of the reasons why this is not the case for the case of Vo and Foo (2005) is the overly liberal freedom to make assumptions in the first place. In our case the arguments considered need to be grounded (or supported) by the given domain description and particularly the narrative it contains. We can thus apply the argumentation process directly on the given knowledge capturing the required reasoning in a natural and simple way. This difference in the two approaches is highlighted by the way each approach uses priorities between arguments. In Vo and Foo (2005), the priorities are used on top to filter pre-structures whose definition depends on notions of argumentation only at a first level whereas in our case the priority is integrated within the argumentation process allowing the notions of argumentation to be sufficient to directly capture the semantics and the reasoning required.

As Vo and Foo (2005) argue, one of the main advantages of an argumentation-based approach to the central challenges of RAC and default reasoning more generally, is that the reasoning can be made more explicit compared to other non-monotonic approaches. This is a general feature shared by most argumentation-based application frameworks including our own in this paper, whose relative simplicity and direct use of argumentation makes it easy to exploit this advantage.

7. Conclusions and further work

We have re-examined the argumentation reformulation of language \mathcal{E} and introduced backwards persistence as well as forward persistence arguments to deal uniformly with the frame and qualification problems. This has enabled us to extend in a meaningful way domains that the language \mathcal{E} could not interpret when observations are included in the narrative. This argumentation interpretation corresponds to the unknown occurrences of events that could resolve potential inconsistencies between properties at different time points. Our extended argumentation framework comes closer in spirit to the original EC. It shows how this could be extended to include positive and negative observations and addressing the qualification problem that these observations may bring about, thus increasing significantly the expressiveness of the original EC.

For future work we will examine how we can integrate our framework for reasoning about properties over time with argumentation-based approaches for decision-making so that these decisions can be context sensitive over time. This will allow us to develop applications of advisory or recommendation systems, as in Chesñevar, Maguitman, and Simari (2004), that can adapt themselves as their external environment evolves. Similarly, we want to apply our framework to planning problems and in particular to the revision of plans as new unexpected information is acquired.

In particular, as the dialectic nature of argumentation is close to human reasoning, with recent studies from cognitive psychology (Mercier and Sperber, 2011) reinforcing this view, our argumentation-based approach for RAC can help its integration with wider forms of human reasoning such as that of discourse comprehension, dialogue and debate. Recent work on story comprehension (Diakidoy, Kakas, Michael, and Miller, 2014) has shown how argumentation can play a significant role in formulating and automating the human process of comprehension. Similarly, in multi-agent systems interaction and communication several works, e.g. McBurney and Parsons (2009) and Kakas, Maudet, and Moraitis (2004), have shown the suitability of using argumentation to model agent dialogues by exploiting the dialectic and game theoretic form of argumentation.

The dynamic nature of dialogues then lends itself to a uniform argumentation-based formalisation for integrating reasoning about changes in the environment of communication with the various dialogue protocols. In particular, the incremental process of a dialogue can be modelled in terms of dynamic argumentation by generalising the proof and game theories of computation (Dung, Kowalski, and Toni, 2006; Modgil and Caminada, 2009) for static argumentation. Allowing arguments to be time dependent we can adapt over time by tracking the changes of arguments and the attacking relation between them. Real-life problems can then be mapped to argumentation frameworks, constructed incrementally from the dynamic knowledge of the problem, e.g. the information exchanged at different time points of a dialogue.

Another way to address these problems of dynamic development is to examine the link between our work and that of dynamic argumentation (see, e.g. Baumann and Brewka, 2010; Liao, Jin, and Koons, 2011) where RAC with new information in the time line of an application problem realizes a dynamic argumentation framework. New arguments and attacks between arguments are enabled as the information unfolds, particularly by observations causing exogenous qualification. We can also explore how various problems that have been studied in the general

setting of dynamic abstract argumentation can help address problems in RAC such as overcoming an exogenous qualification by finding what actions need to occur so that some conclusion would necessarily follow.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes

1. PFP (resp. PBP) stands for Positive Forward (resp. Backward) Persistence, NFP (resp. NBP) stands for Negative Forward (resp. Backward) Persistence, PG_F (resp. PG_B) stands for Positive Forward (resp. Backward) Generation, NG_F (resp. NG_B) stands for Negative Forward (resp. Backward) Generation, PO (resp. NO) stands for Positive (resp. Negative) Observation and finally PA (resp. NA) stands for Positive (resp. Negative) Assumption.
2. $B(D) \cup X \vdash_{\min} l$ if and only if $B(D) \cup X \vdash l$ and there does not exist $X' \subset X$ such that $B(D) \cup X' \vdash l$.
3. An initiation or termination point for a fluent f in an admissible extension E is defined as in Definition 2.2 where now the preconditions C of the c -proposition are satisfied at T in E when the corresponding *HoldsAT* conclusions are derived by E .
4. For simplicity of presentation we are assuming that there can only be one generating (initiating or terminating) action of a given fluent at any time point.
5. When we have the other case of ‘ L holds-at T_0 ’ the extended domain will have admissible extensions with an exogenous qualification of persistence of L in $(T_0, T_1]$.
6. Note that t_{k+1} can be both an initiation and a termination point for f in M but any given model the fluent f will have either the value true or false (see case (iv)).

References

- Baumann R., & Brewka G. (2010). *Expanding argumentation frameworks: enforcing and monotonicity results*. Computational models of argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8–10, 2010, pp. 75–86.
- Bench-Capon T. J. M., & Dunne P. E. (2007). Argumentation in artificial intelligence. *Artificial Intelligence*, 171, 619–641.
- Bondarenko A., Dung P. M., Kowalski R. A., & Toni F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93, 63–101.
- Chesñevar C. I., Maguitman A. G., & Simari G. R. (2004). *A first approach to argument-based recommender systems based on defeasible logic programming*. Proceedings of the 10th international workshop on non-monotonic reasoning (NMR 2004), Whistler, Canada, June 6–8, 2004, pp. 109–117.
- Diakidoy I. A., Kakas A. C., Michael L., & Miller R. (2014). Story comprehension through argumentation. *Proceedings of the 5th international conference on computational models of argument (COMMA 2014)*, September, Frontiers in artificial intelligence and applications (Vol. 266, pp. 31–42). Scottish Highlands: IOS Press.
- Dimopoulos Y., & Kakas A. C. (1995). *Logic programming without negation as failure*. Logic programming, proceedings of the 1995 international symposium, Portland, OR, USA, December 4–7, 1995, pp. 369–383.
- Dung P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77, 321–357.
- P. M. Dung, R. A. Kowalski, & Toni F. (2006). Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170, 114–159.
- Gelfond M., & Lifschitz V. (1993). Representing action and change by logic programs. *The Journal of Logic Programming*, 17, 301–321.

- Kakas A. C., Kowalski R. A., & Toni F. (1992). Abductive logic programming. *Journal of Logic and Computing*, 2, 719–770.
- Kakas A. C., Mancarella P., & Dung P. M. (1994). *The acceptability semantics for logic programs*. Logic programming, proceedings of the eleventh international conference on logic programming, Santa Marherita Ligure, Italy, June 13–18, 1994, pp. 504–519.
- Kakas A. C., Maudet N., & Moraitis P. (2004). *Flexible agent dialogue strategies and societal communication protocols*. Paper presented at the 3rd international joint conference on autonomous agents and multiagent systems (AAMAS 2004), New York, NY, USA, August 19–23, 2004, pp. 1434–1435.
- A. Kakas, & Miller R. (1997). A simple declarative language for describing narratives with actions. *Journal of Logic and Algebraic Programming*, 31, 157–200.
- Kakas A. C., Miller R., & Toni F. (1999). An argumentation framework of reasoning about actions and change. *LPNMR*, El Paso, Texas, USA, pp. 78–91.
- Kautz H. A. (1986). *The logic of persistence*. Proceedings of the 5th national conference on artificial intelligence: Volume 1: Science, Philadelphia, PA, August 11–15, 1986, pp. 401–405.
- R. Kowalski, & M. Sergot (1986). A logic-based calculus of events. *New Generation Computing*, 4, 67–95.
- Liao B., Jin L., & Koons R. C. (2011). Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175, 1790–1814.
- McBurney P., & Parsons S. (2009). Dialogue games for agent argumentation. In G. Simari & I. Rahwan (Eds.), *Argumentation in artificial intelligence* (pp. 261–280). Springer.
- McCain N., & Turner H. (1997). *Causal theories of action and change*. Proceedings of the AAAI-97, Providence, Rhode Island, pp. 460–465.
- McCarthy J., & Hayes P. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4, 463–502.
- Mercier H., & Sperber D. (2011). Why do humans reason? Arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34, 57–74.
- Miller R., & Shanahan M. (1999). The event calculus in classical logic – alternative axiomatisations. *Electronic Transaction of Artificial Intelligence*, 3, 77–105.
- Modgil S., & Caminada M. (2009). Proof theories and algorithms for abstract argumentation frameworks. In G. Simari & I. Rahwan (Eds.), *Argumentation in artificial intelligence* (pp. 105–129). Springer.
- Modgil S., & Prakken H. (2013). A general account of argumentation with preferences. *Artificial Intelligence*, 195, 361–397.
- Rahwan I., & Simari G. (2009). *Argumentation in artificial intelligence*. Springer Publishing Company, Incorporated.
- Shanahan M. (1997). *Solving the frame problem – a mathematical investigation of the common sense law of inertia*. MIT Press.
- Thielscher M. (1999). From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem. *Artificial Intelligence*, 111, 277–299.
- Thielscher M. (2001). The qualification problem: A solution to the problem of anomalous models. *AIJ*, 131, 1–37.
- Vo Q. B., & Foo N. Y. (2005). Reasoning about action: An argumentation-theoretic approach. *Journal of Artificial Intelligence Research*, 24, 465–518.

Appendix 1. Lemma proofs

Proof of Lemma 4.2. Let f be a fluent, $t_n < t_m$ two time points such that $E \vdash \text{HoldsAt}(f, t_n)$ and $E \vdash \text{HoldsAt}(f, t_m)$ (the case where $E \vdash \neg \text{HoldsAt}(f, t_n)$ and $E \vdash \neg \text{HoldsAt}(f, t_m)$ is analogous). Assume that $E \vdash \neg \text{HoldsAt}(f, t)$ for some time $t \in (t_n, t_m)$. We will show that this leads to a contradiction of the admissibility of E . As time is discrete and so there are finitely many time points in the interval (t_n, t_m) we can consider the first time point T in $[t_n, t_m]$ at which we have a change of the value of f by E , i.e. $E \vdash \text{HoldsAt}(f, T)$, $E \vdash \neg \text{HoldsAt}(f, T + 1)$. Consider now the set $S = E_1 \cup \{\text{PFP}(f, T + 1; T)\}$ where E_1 is a minimal subset of E that derives f at T . Then $S \vdash \text{HoldsAt}(f, T + 1)$ and so it is a potential attack on E . In fact, since there are no termination points for f in $[t_n, t_m]$ and observations of $\neg f$ in $[t_n, t_m]$ the extension E cannot derive $\neg f$ at $T + 1$ with an argument stronger than $\text{PFP}(f, T + 1; T)$ and so S attacks E . The only

possible way for E to attack back S is via a conflict at $T + 1$ by deriving $\neg f$ at $T + 1$, as it cannot attack S on its E_1 subset since this would mean that E is inconsistent.

Again, since there are no termination points for f in $[t_n, t_m)$ and observations of $\neg f$ in $[t_n, t_m)$ the extension E can derive $\neg f$ at $T + 1$ either by a forward persistence of $\neg f$ starting from a time point before T or by backward persistence of $\neg f$ from a time point after $T + 1$, but not both due to the compactness property of an admissible extension (see Definition 3.7) that explicitly excludes this. Note that if there is an initiation point for f at t_m and hence we have the possibility to have in the argumentation framework a $NG_B[f, t_m]$ argument for $\neg f$ at t_m this cannot belong to E as then E would be inconsistent (at t_m since E derives f at t_m). In the first case this cannot constitute an attack as $PFP(f, T + 1; T)$ is stronger than the forward persistence of $\neg f$ starting from a time point before T .

In the second case this backward persistence of $\neg f$ must have its origin from a point after t_m as there are no termination points or observations before t_m . But then the set $S' = E' \cup \{PBP(f, T + 1; t_m)\}$ where E' is a minimal subset of E that derives f at t_m attacks E . This can only be defended by E with a backwards persistence argument starting before t_m which is not possible as there are no termination points for f or observation points for $\neg f$. Note that if we have a chain of backward persistence starting from a time after t_m to an intermediate time point, t_i , before t_m but after $T + 1$ and then another backwards persistence from this to $T + 1$, the extension E still needs to contain a persistence argument for $\neg f$ starting after t_m and reaching this intermediate time point before t_m . Then the attack, $S'' = E' \cup \{PBP(f, t_i; t_m)\}$, analogously to S' cannot be attacked back by E . Similarly, if the way that E derives $\neg f$ at $T + 1$ is through a forward persistence from a time point before t_n to a time point, t' , after $T + 1$ and then a backwards persistence from t' to $T + 1$, then E would not be able to attack back the attack, $S''' = E_1 \cup \{PFP(f, t'; T)\}$, where E_1 is a minimal subset of E that derives f at T . Hence one of the attacks, S, S', S'' or S''' , cannot be attacked back by E and so E would not be admissible. Contradiction. ■

Appendix 2. Propositions proofs

Proof of Proposition 4.1. By contradiction. Let D be a domain and E an admissible extension that is not consistent with the t -propositions in D . Therefore, there exists a fluent f and a time point t such that ‘ f holds-at t ’ belongs to D and $E \vdash \neg \text{HoldsAt}(f, t)$ (resp. ‘ $\neg f$ holds-at t ’ belongs to D and $E \vdash \text{HoldsAt}(f, t)$). Consider the case $E \vdash \neg \text{HoldsAt}(f, t)$ (the respective case is completely analogous). Let B be the set of argument rules $B = \{PO[f, t]\}$. We will show that B is an attack against E that is not attacked back by E , thus contradicting the admissibility of E . B attacks E because $B(D) \cup B \vdash \text{HoldsAt}(f, t)$ minimally and $E \vdash \neg \text{HoldsAt}(f, t)$ and there is no argument that has higher priority than $PO[f, t]$ at t . This attack can only be attacked back by deriving $\neg \text{HoldsAt}(f, t)$. The (minimal) derivation of $\neg \text{HoldsAt}(f, t)$ by E will require either a persistence argument or a local generation argument whose conclusion will be $\neg \text{HoldsAt}(f, t)$. The priority relation in Definition 3.2 assigns to these lower priority than $PO[f, t]$ and hence since B does not contain any other argument rule such derivations cannot form an attack against B . The only other possibility for E to derive $\neg \text{HoldsAt}(f, t)$ is through $PO[f, t]$ but this is not possible due to the point-wise consistency of D . ■

Appendix 3. Theorems proofs

Proof of Theorem 4.3. Let D be given with a countable number of h - and t -propositions. We will show the existence of an admissible extension E by induction on the number of h -propositions in D .

Basecase : Number of h -propositions is zero.

To prove this case we will use a second induction on the number of t -propositions, which we have also assumed to be countable.

Base case: Number of t -propositions is zero.

Let $E = \bigcup_f \{PA[f, 0], PFP[f, t; 0] \text{ for every fluent } f \text{ and every time point } t > 0\}$. By construction E is complete, consistent and clearly satisfies the compactness property of an admissible extension. It remains to show that E attacks all its attacks. To have an attack A on E , A needs to derive $\neg \text{HoldsAt}(f, t)$ for some fluent f and time point t , so as to have a contrary conclusion with E . This can only happen if A contains the argument $NA(f, 0)$. But then E attacks back A through its argument $PA(f, 0)$.

Induction step : Suppose that there exist a (complete) admissible extension, E' , of the corresponding argumentation framework $(B(D'), \mathcal{A}, <)$ for any domain D' that contains up to k t -propositions and no h -propositions. We will prove that the statement holds when a domain D contains $k + 1$ t -propositions (and no h -propositions).

Consider the last t -proposition and let its time point be t_{k+1} . If we take it out of D the resulting domain, D' , will only contain k many t -propositions. By the induction hypothesis, let E' be a (complete) admissible extension of D' . We will use E' to construct an extension E for D as follows. Let f be the fluent to which the t -proposition taken out of D refers to and consider the last t -proposition in D' on f . Let its time be t_l , $t_l \leq t_{k+1}$. We will construct an admissible extension E for D as the union, $E = E_f^- \cup E_f^+$, of two sets of arguments obtained and extended from E' . We have the following cases:

Case (a) there does not exist such a t -proposition, i.e. the observation at t_{k+1} is the only one in D that refers to f . Assume that this observation at t_{k+1} , is that f holds (the case where $\neg f$ holds at t_{k+1} is analogous). Then let

$$\begin{aligned} E_f^- &= E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f \text{ for any time point} \} \text{ and} \\ E_f^+ &= \{PO[f, t_{k+1}], PFP[f, t_1; t_{k+1}], PBP[f, t_2; t_{k+1}] \mid t_1 > t_{k+1}, t_2 < t_{k+1}\}. \end{aligned}$$

Case (b) the (last) observation at time t_l also states that f holds (i.e. as does the observation for the fluent f at time t_{k+1}). Let $E_f^- = E'$ and $E_f^+ = \emptyset$.

Case (c) the (last) observation at time t_l is opposite to the observation for the fluent f at time t_{k+1} , i.e. at time $t_l f$ is observed not to hold whereas at t_{k+1} it is observed to hold. Then let

$$\begin{aligned} E_f^- &= E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f \text{ for any time point, } t \geq t_l \}. \text{ Choose a time point } T_i \in [t_l, t_{k+1}) \text{ and let} \\ E_f^+ &= \{NO[f, t_l], NFP[f, t; t_l] \mid t \in (t_l, T_i]\} \cup \{PO[f, t_{k+1}], PFP[f, t_1; t_{k+1}], PBP[f, t_2; t_{k+1}] \mid t_1 > t_{k+1}, t_2 \\ &\quad \in (T_i, t_{k+1})\}. \end{aligned}$$

By construction, $E = E_f^- \cup E_f^+$ in every case is consistent, complete and compact. To show that E is admissible let us consider the possible new attacks on E that result from the changes to E' for the three cases above. Case (a): new attacks need to prove $\neg \text{HoldsAt}(f, t)$ for some time point t . This can only happen by a potential attack A that contains the argument $NA[f, 0]$ as in D there does not exist any h -propositions and the observation at t_{k+1} is the only one referring to f . But any attack of this form on $E_f^+ = \{PO[f, t_{k+1}], PFP[f, t_1; t_{k+1}], PBP[f, t_2; t_{k+1}] \mid t_1 > t_{k+1}, t_2 < t_{k+1}\}$ can be counterattacked by E_f^+ on the argument $NA[f, 0]$ of A . Case (b): E does not differ from E' and hence it remains admissible. Case (c): let us consider new attacks on E that are not attacks on E' after and before t_{k+1} .

- (i) after t_{k+1} : any potential new attack on E on the fluent f can only come through forward persistence of $\neg f$ from some time point $T \leq t_l < t_{k+1}$ as there are not any h -propositions in D and the last observation for f before t_{k+1} is at time t_l . But this is not an attack on $E_f^+ = \{NO[f, t_l], NFP[f, t; t_l] \mid t \in (t_l, T_i]\} \cup \{PO[f, t_{k+1}], PFP[f, t_1; t_{k+1}], PBP[f, t_2; t_{k+1}] \mid t_1 > t_{k+1}, t_2 \in (T_i, t_{k+1})\}$ as forward persistence arguments from latter time points are stronger.
- (ii) before t_{k+1} and t_l new potential attacks on E can only occur by the observation at t_{k+1} . Let us then consider a potential new attack, A' on E given by $PO[f, t_{k+1}], PBP[f, T; t_{k+1}]$ for some $T < t_l$. We then consider how E' and E derive $\neg f$ at T . There are two possibilities, either by $NO[f, t_l], NBP[f, T; t_l]$ in which case A' does not attack E' , or through some forward persistence argument from some other observation for $\neg f$ before t_l or from a negative assumption for $\neg f$ at 0 . In both of these latter cases, E' would counter attack the attack A' as forward and backwards persistence arguments are non-comparable.
- (iii) before t_{k+1} but after t_l attacks on E are based on a conflict between forward and backwards persistence in the interval (t_l, t_{k+1}) . These are counter attacked by $E_f^+ = \{NO[f, t_l], NFP[f, t; t_l] \mid t \in (t_l, T_i]\} \cup \{PO[f, t_{k+1}], PFP[f, t_1; t_{k+1}], PBP[f, t_2; t_{k+1}] \mid t_1 > t_{k+1}, t_2 \in (T_i, t_{k+1})\}$ in E since conflicting forward and backwards persistence arguments have the same priority.

Induction step: Suppose that there exist an (complete) admissible extension, E' , of the corresponding argumentation framework $(B(D'), \mathcal{A}, <)$, for any domain D' that contains k or less h -propositions. We will prove that the statement holds when a domain D contains $k + 1$ h -propositions.

Consider the last h -proposition that occurred at time point t_{k+1} . If we take it out of D the resulting domain, D' , will only contain k many h -propositions. By the induction hypothesis, let E' be a (complete) admissible extension of D' . We will use E' to construct an extension E for D as follows. Let E be given by

$$E = \bigcup_f E_f^- \cup \bigcup_f E_f^+,$$

where E_f^- is defined from E' by taking out some arguments for the fluent f and E_f^+ are some new arguments for the fluent f . For any fluent f we have the following cases:

(Case 1) t_{k+1} is an initiation point for f in E' .

(Case 2) t_{k+1} is a termination point for f in E' . This case is similar to case 1. Note that t_{k+1} can be both an initiation and a termination point for the same fluent f . In such a case we can pick either case in the construction.

(Case 3) t_{k+1} is neither an initiation nor a termination point for f in E' . In this case $E_f^- = E_f'$ and $E_f^+ = \{\}$, i.e. nothing (related to the fluent f) is removed or added.

Hence let us assume that t_{k+1} is an initiation point for f in E' and consider the following changes in E' in order to build an admissible extension E .

These changes to E' will all be after time point t_{k+1} . Note that there is no need to do any changes before t_{k+1} as the existence of the initiation point for f at t_{k+1} can enable new possible attacks on E' (and the resulting E that we are constructing) only through the argument $NG_B[f, t_{k+1}]$ and either backwards persistence of $\neg f$ from this or forwards persistence. In the case of backwards persistence, this attack will always be counter attacked by E' as backwards persistence arguments are non-comparable to other arguments or they will be weaker than earlier backwards persistence in E' . For the new forwards persistence of $\neg f$ from $NG_B[f, t_{k+1}]$ possible attacks, we will see below that these persistence arguments will be weaker than the new arguments E_f^+ that we add in E' .

Consider therefore the first t -proposition referring to f after t_{k+1} at time, $t_n > t_{k+1}$. We have the following cases:

(a₁) If there does not exist such a t -proposition let

$$E_f^- = E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f, \text{ for any } t > t_{k+1} \} \text{ and}$$

$$E_f^+ = \{PG_F[f, t_{k+1}], PFP[f, t; t_{k+1} + 1] \mid t > t_{k+1} + 1\}.$$

(a₂) The first t -proposition after t_{k+1} , at time t_n , confirms (i.e. we observe that f holds at t_n) the initiation point for the fluent f at time t_{k+1} . Let

$$E_f^- = E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f, \text{ for any } t \in (t_{k+1}, t_n) \} \text{ and}$$

$$E_f^+ = \{PG_F[f, t_{k+1}], PFP[f, t; t_{k+1} + 1] \mid t \in (t_{k+1} + 1, t_n)\}.$$

(a₃) The first t -proposition after t_{k+1} at time t_n is opposite (i.e. we observe that f does not hold at t_n) to the initiation point for the fluent f at t_{k+1} . Choose $T_i \in [t_{k+1}, t_n]$. Let

$$E_f^- = E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f, \text{ for any } t \in (t_{k+1}, T_i) \} \text{ and}$$

$$E_f^+ = \{PG_F[f, t_{k+1}], PFP[f, t; t_{k+1} + 1] \mid t \in (T_i, t_n) \} \cup \{NO(f, t_n), NBP[f, t'; t_n] \mid t' \in (T_i, t_n)\}.$$

(b) Changes to E' before or at time point t_{k+1} : Consider the last t -proposition referring to f or h -proposition that can generate f or $\neg f$ in E' at time $t_m \leq t_{k+1}$.

(b₁) If there does not exist such a t -proposition or an h -proposition, i.e. the h -proposition at t_{k+1} is the only one in D that refers to f and there does not exist a t -proposition in D at $t_m \leq t_{k+1}$ that refers to f . Let

$$E_f^- = E' \setminus \{ \text{All arguments in } E' \text{ that refer to } f, \text{ for any } t \leq t_{k+1} \} \text{ and}$$

$$E_f^+ = \{NA[f, 0], NFP[f, t; 0] \mid t < t_{k+1}\}.$$

(b₂) If there exists a t -proposition or an h -proposition at $t_m \leq t_{k+1}$ opposite to the initiation point for the fluent f at time t_{k+1} (i.e. the observation at t_m is that f does not hold at t_m or at t_m we have a termination point for f), then no change before t_{k+1} of E' is needed.

(b₃) Similarly, if there exists a t -proposition at $t_m \leq t_{k+1}$ where f is observed to hold at t_m or there exists an h -proposition at $t_m \leq t_{k+1}$ such that t_m is also an initiation point for the fluent f , then no change before t_{k+1} of E' is needed.

By construction E is complete, consistent and compact. We need further to show that E attacks back any of its attacks. Consider all possible new attacks on E through the different cases given above. Case a₁. Any new potential attack on E after t_{k+1} needs to prove $\neg HoldsAt(f, t)$ for some $t > t_{k+1}$. Such a minimal proof can only be built from either an observation for the fluent $\neg f$ before or equal to t_{k+1} , a termination for the fluent f before or equal to t_{k+1} or an assumption for $\neg f$ at 0. But all three cases do not constitute an attack on the $E_f^+ = \{PG_F[f, t_{k+1}], PFP[f, t; t_{k+1} + 1] \mid t > t_{k+1} + 1\}$ as between conflicting forward persistence arguments higher priority have the arguments occurring at a latter time point. Hence there are no new attacks on this new part of E .

Case a_2 . A potential new attack on E needs to prove $\neg \text{HoldsAt}(f, t)$ for $t \in (t_{k+1}, t_n]$. Such a proof can only be built from an observation or a generation point before or equal to t_{k+1} or by an assumption at 0. Similar to case a_1 these are not attacks. Another possibility are proofs starting after t_n by observations of $\neg f$ by backward persistence. This attacks E but it is counter attacked by E because no priority is given between conflicting forward persistence arguments over backward persistence arguments.

Case a_3 . Any new potential attack on E at $(t_{k+1}, T_i]$ needs to prove $\neg \text{HoldsAt}(f, t)$ for any $t \in (t_{k+1}, T_i]$. Such a minimal proof can only be built from either an assumption for the fluent $\neg f$ at 0, an observation for the fluent $\neg f$ before or equal to t_{k+1} or by a termination for the fluent f before or equal to t_{k+1} . But similarly to case a_1 all three are not attacks. Other new attacks require a proof of $\neg f$ starting after t_n . But these attacks are counter attacked (similarly to case a_2). Also new attacks on E at $(T_i, t_n]$ need to prove $\text{HoldsAt}(f, t)$ for any $t \in (T_i, t_n]$. Such a minimal proof can only be built from either an assumption for the fluent f at 0, an observation for the fluent f before T_i or by an initiation for the fluent f before T_i . All three are attacks but can be counter attacked by the constructed E . ■

Proof of Theorem 4.4. Let D be given and let M be a language \mathcal{E} model of D . We will show the existence of a corresponding admissible extension E by induction on the number of h -propositions in D .

Base case: Number of h -propositions is zero.

Case (1): Number of t -propositions (observations) in D is zero.

By definition of the Models of \mathcal{E} , for every fluent f either $M(f, t) = \text{true}$ for every t or $M(f, t) = \text{false}$ for every t . For any fluent f such that $M(f, t) = \text{true}$ for every t , let $E_f = \text{PA}(f, 0) \cup \{\text{PFP}(f, t; 0) \mid \text{for every } t\}$ and for any fluent f' such that $M(f', t) = \text{false}$ for every t , let $E_{f'} = \text{NA}(f', 0) \cup \{\text{NFP}(f', t; 0) \mid \text{for every } t\}$. Let us then define E as follows:

$$E = \bigcup_{f \mid M(f, t) = \text{true}} E_f \cup \bigcup_{f' \mid M(f', t) = \text{false}} E_{f'}.$$

By construction E corresponds to the model M . We will show that E is an admissible extension. By construction E is consistent, complete and compact. In order for an argument set A , to attack E , for some fluent f and time point t a contrary conclusion should exist, i.e. $A \vdash \neg \text{HoldsAt}(f, t)$ when $M(f, t) = \text{true}$ (resp. $A \vdash \text{HoldsAt}(f', t)$ when $M(f', t) = \text{false}$). This is only possible if A contains $\text{NA}(f, 0)$ (resp. contains $\text{PA}(f', 0)$). E therefore attacks back A through its argument $\text{PA}(f, 0)$ (resp. $\text{NA}(f', 0)$). Furthermore, by construction E corresponds to the given model M .

Case (2): The domain D contains t -propositions (observations).

Given the existence of the model, M , of D and the fact that there are no h -propositions in D all observations in D for a given fluent f are either all for f to hold or all for $\neg f$ to hold at the different time points where we have observations. Therefore as in Case (1), all fluents have a constant truth value in M at all time points. Thus we can define E as before:

$$E = \bigcup_{f \mid M(f, t) = \text{true}} E_f \cup \bigcup_{f' \mid M(f', t) = \text{false}} E_{f'}.$$

New attacks on E based on observations are not possible since by construction of E such an attack would require an observation for a fluent f which is contrary to the constant truth value for f in M .

Induction step : Suppose that the statement holds for any domain such that the number of h -propositions is less or equal to k .

Let D be a domain with $k + 1$ h -propositions and let M be a model of D . Consider the last h -proposition and assume that this occurred at the time point t_{k+1} . We then consider the new domain D' resulting by taking out of D this last h -proposition as well as all the t -propositions after this time point. The domain D' contains k many h -propositions. By the induction hypothesis there exist an admissible extension E' for $(B(D'), \mathcal{A}', \prec)$ corresponding to the model, M' , of D' obtained from the model M by replacing the truth value of every fluent, for every time after t_{k+1} , to be the same as its truth value in M at t_{k+1} . We then define E , from E' by considering for each fluent the initiation or termination status of the time point t_{k+1} and the possible t -propositions in D after this time point.

Let f be any fluent:

- (i) Let t_{k+1} be an initiation point for the fluent f in M and that M assigns true to f after t_{k+1} .⁶ Let $E_f^- = E'_f \setminus \{\text{All arguments in } E' \text{ with conclusion } \neg \text{HoldsAt}(f, T) \text{ or } \text{HoldsAt}(f, T), \text{ for any } T > t_{k+1}\}$ and $E_f^+ = \{\text{PG}(f, t_{k+1})\} \cup \{\text{PFP}(f, t_1; t_{k+1} + 1) \mid t_1 > t_{k+1} + 1\}$. Here and below E'_f denotes the subset of E' of all the arguments that refer to the fluent f .

- (ii) Let t_{k+1} be a termination point for the fluent f in M . Let $E_f^- = E'_f \setminus \{\text{All arguments in } E'\text{ with conclusion } \text{HoldsAt}(f, T) \text{ or } \neg\text{HoldsAt}(f, T), \text{ for any } T > t_{k+1}\}$ and $E_f^+ = \{\text{NG}(f, t_{k+1})\} \cup \{\text{NFP}(f, t_1; t_{k+1} + 1) \mid t_1 > t_{k+1} + 1\}$.
- (iii) Let t_{k+1} be neither an initiation nor a termination point for the fluent f in M . Let $E_f^- = E'_f$ and $E_f^+ = \emptyset$.
- (iv) Let t_{k+1} is both an initiation and a termination point for f in M . The model M of \mathcal{E} will nondeterministically have chosen the value *true* or *false* for f after t_{k+1} . If this value is *true* then E_f^- and E_f^+ are defined as in case (i). Otherwise, if it is *false* then E_f^- and E_f^+ are defined as in case (ii).

We then define E as follows:

$$E = \bigcup_f E_f^- \cup \bigcup_f E_f^+.$$

By construction and the inductive hypothesis on E' , E is consistent, complete and compact. Also by construction E corresponds to M . We also note that if there exists t -propositions on a fluent f after time t_{k+1} then they must all give the same truth value for f since otherwise the domain D would not have a model. This truth value of f in M will coincide with the derivation of f or $\neg f$ after t_{k+1} by the constructed E .

It remains to show that E attacks all its attacks. New attacks on E that were not attacks on E' can occur by the possible generation point at t_{k+1} . Since t_{k+1} is the last generation point new attacks on E can be built by backwards persistence from this, conflicting with forward persistence arguments from $t < t_{k+1}$. But then, E can counter attack any of these attacks since forward persistence has same priority than conflicting backwards persistence. Note also that observations of f after t_{k+1} cannot generate any new attacks on E since if t_{k+1} is an initiation point in M for a fluent f then these observations must all be for f to be true (otherwise M would not be a model) and hence by the construction of E the observations cannot generate an argument conflicting with the arguments in E . Similarly, when t_{k+1} is a termination point in M . ■

Proof of Theorem 4.5. Let E be a given admissible extension of D and consider the interpretation H^E that correspond to E . H^E is well defined as admissible extensions are complete. Suppose that H^E is not a model of D in language \mathcal{E} . Then one of the four conditions of Definition 2.3 of a model must be violated. Given the result of Proposition 4.1, H^E cannot violate property (4) of Definition 2.3, since admissible extensions are consistent with respect to the t -propositions in the domain. Thus H^E violates one of properties (1–3).

Violation of property 1: There exist a fluent f and time points t_1, t with $t_1 < t$ such that $E \vdash \text{HoldsAt}(f, t_1)$ and $E \vdash \neg\text{HoldsAt}(f, t)$ (or vice versa) and there exists no termination point in $[t_1, t]$ relative to E (or H^E). Since time is discrete we can choose t_1, t to be consecutive times.

We then add in D a new h -proposition ‘A happens-at t_1 ’ such that ‘A terminates F ’. In the new domain D_1 obtained from this addition this violation of the model property is removed.

Violation of property 2: There exist a fluent f and time points $t_1 < t$ such that t_1 is an initiation point of f relative to H^E , $E \vdash \neg\text{HoldsAt}(f, t)$ and there exists no termination point in (t_1, t) for f in H^E . Due to the discrete nature of time, we can choose t to be the closest time point to t_1 where there is such a violation, i.e. t is the closest time point after t_1 where $E \vdash \neg\text{HoldsAt}(f, t)$.

We then add in D a new h -proposition ‘A happens-at T' ’ at $T' = t - 1$ such that ‘A terminates F ’. In the new domain D_2 obtained from this addition the violation of the model property is removed. Note that T' can be equal to t_1 in which case language \mathcal{E} has models for either f or $\neg f$ to hold after t_1 .

Violation of property 3: The treatment of this violation is analogous to the violation of property (2) and results in a new domain D_3 that removes one such violation.

We then consider the domain $D_{123} = D_1 \cup D_2 \cup D_3$. If H^E is not a model of D_{123} then we repeat the above construction. Due to the finiteness of the h -propositions and t -propositions in D and the discrete nature of the time line this process terminates and results in the required new domain D' . ■