Features and Fluents for Logic Programming: Non-simulative Algebraic Semantics

Brandano, Sergio

QUEEN MARY
AND WESTFIELD COLLEGE
UNIVERSITY OF LONDON

Department of Computer Science

Research Report No. RR-00-05 ISSN 1470-5559 September 2000

Features and Fluents for Logic Programming: Non-simulative Algebraic Semantics

Sergio Brandano

Features and Fluents for Logic Programming: Non-simulative Algebraic Semantics

Sergio Brandano

Department of Computer Science
Queen Mary and Westfield College
University of London
sb@dcs.qmw.ac.uk

Abstract

A Non-simulative Algebraic Semantics is defined and its range of applicability is proven to be the $\mathcal{K}$-**RACi** class of the *Features and Fluents* framework. The comparative assessment reveals the semantics epistemologically equivalent and ontologically stronger than the Abductive Logic Programming, the Action Description Language $\mathcal{A}$ and the $PMON$ entailment. The semantics is shown to be decidable.

Keywords: Features and Fluents, Logic Programming.

1 Introduction

The *reasoning method* of van Emden and Kowalski's algebraic semantics for the Horn Clause Logic is adequate for solving non-monotonic temporal-reasoning problems within the $\mathcal{K}$**sp-IAd** class of the *Features and Fluents* framework [2]. What is characteristic of this method, as shown in the assessment result, is the ability to simulate the ego-world game for $\mathcal{K}$-**IA** when the world player has a fixed strategy, leading to the proven restricted range of applicability.

In this paper, we design the converse non-simulative method and we study the range of applicability of the resulting definition. The comparative assessment reveals non-simulative algebraic semantics (NAS) both epistemologically and ontologically stronger than the simulative algebraic semantics. The comparison also reveals NAS epistemologically equivalent and ontologically stronger than the Abductive Logic Programming [5], the Action Description Language $\mathcal{A}$ [6] and the $PMON$ entailment [10]. The overall range of applicability is proven to be the full $\mathcal{K}$-**RACi** [1, 3] class of the *Features and Fluents* framework, a superclass of $\mathcal{K}$-**IA** [10] where strict inertia in continuous time applies, together with continuous change and alternative results of possibly concurrent and independent actions. The semantics is shown to be decidable. In order to get acquainted with all its aspects, the reader may find helpful experimenting with the given meta-interpreter. A companion web-site to this paper and its software can be accessed via the Internet address given at the end of the paper.

2 The Language

The language is identical to the language described in [2, section 4]. Let $\mathcal{T}$ be the timepoint domain, $\mathcal{F}$ the set of all feature symbols, $\mathcal{V}$ the domain of all feature values, and $\mathcal{E}$ the set of all action symbols. Let $\langle \mathcal{H}, \sqsubseteq \rangle$ be the lattice which elements, called *observations*, are members of $\mathcal{H} = \mathcal{T} \times \mathcal{F} \times \mathcal{V}$ and the order relation $\sqsubseteq$ applies as follows: $\langle t_1, f_1, v_1 \rangle \sqsubseteq \langle t_2, f_2, v_2 \rangle$ iff $t_1 \sqsubseteq t_2$ and $f_1 = f_2$. The tuple $\langle t, f, \mathit{unknown} \rangle$ is an abbreviation for $\bigvee_i \langle t, f, v_i \rangle$, varying i over all possible tuples $\langle t, f, v_i \rangle$ in $\mathcal{H}$. Let $\langle \mathcal{D}, \sqsubseteq \rangle$ be the lattice which elements, called *rigid occurrences of actions*, are members of $\mathcal{D} = \mathcal{T} \times \mathcal{T} \times \mathcal{E}$ and the order relation $\sqsubseteq$ applies as follows: $\langle s_1, t_1, A_1 \rangle \sqsubseteq \langle s_2, t_2, A_2 \rangle$ iff $s_1 \sqsubseteq s_2$. The order relation $\sqsubset$ is an abbreviation for $\sqsubseteq \wedge \neq$. The relation $\langle s_1, t_1, A_1 \rangle = \langle s_2, t_2, A_2 \rangle$ simply means that A_1 and A_2 start at the same timepoint, while $\langle s_1, t_1, A_1 \rangle \sqsubset \langle s_2, t_2, A_2 \rangle$ means that A_1 starts earlier than A_2. Finally, let Υ be a scenario description.

- The OBS part of Υ is a sub-lattice of $\langle \mathcal{H}_\Upsilon, \sqsubseteq \rangle$, which elements are members of $\mathcal{H}_\Upsilon = \mathcal{T} \times \mathcal{F}_\Upsilon \times \mathcal{V} \subseteq \mathcal{H}$, where F_Υ is the set of all features occurring in Υ.
- The SCD part of Υ is a sub-lattice of $\langle \mathcal{D}, \sqsubseteq \rangle$. Each tuple in SCD specifies the starting time, the ending time and the action symbol of an action scheduled for execution.
- The function $\Rrightarrow : \mathcal{D} \rightarrow 2^{\mathcal{H}_\Upsilon}$ maps each schedule's occurrence in a set of non-empty lattices of observations. The function $\Rrightarrow$ is parametric on the action type, and the LAW part of Υ consists in the definition of $\Rrightarrow$ as a set of action laws in Full Trajectory Normal Form, one law for each action type.

  The Full Trajectory Normal Form for the action laws is a mapping $\langle s, t, A \rangle \Rrightarrow \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} S_{ij}$ for which the action occurrence $\langle s, t, A \rangle$ is *expanded* into a formula in Full Disjunctive Normal Form, that is into a disjunction of conjunctions of trajectory formulas S_{ij}, each of which corresponds to the feature f_j in the alternative i. A trajectory formula for a given feature f_j in $\mathcal{F}$ is the first-order formula $\forall \tau \in [s, t] \subset \mathcal{T} \,.\, [\tau] f_j \doteq \varphi_j(\tau)$ where the descriptor φ_j is a partial fluent defined over $D \subseteq [s, t] \subset \mathcal{T}$, and $s \neq t$.

A scenario description is a recursive definition for problem domain specification, where observations in OBS and occurrences of actions in SCD are regarded as being *true propositions*, and action laws in LAW are regarded as being *rules* which from a number of true propositions they permit to infer other true propositions. In the specific case of action laws, inferred propositions can only be observations, i.e. tuples in $\mathcal{H}$.

3 The Semantics

Let Υ be a scenario description and let $\langle \tau, f, v \rangle$ be a tuple in $\mathcal{H}$. The decisional problem we are about to address consists in assigning a proper truth value to the tuple $\langle \tau, f, v \rangle$ using the information Υ. The problem is equivalent to asking whether Υ, augmented with $\langle \tau, f, v \rangle$ in its OBS part, is a consistent set. As more than one intended model is possible, we also want to identify that single set V of values in $\mathcal{V}$ such that for any $v \in V$ the tuple $\langle \tau, f, v \rangle$ is *true* in at least one intended model. The ultimate task is to conclude everything about the values of features at different points in time.

According to the fixpoint semantics [2] an interpretation for Υ is any subset of $\mathcal{H}_\Upsilon$ and the model for Υ is the least fixpoint of a continuous transformation $T_\Upsilon : 2^{\mathcal{H}_\Upsilon} \rightarrow 2^{\mathcal{H}_\Upsilon}$, where $2^{\mathcal{H}_\Upsilon}$ is a complete lattice under the partial order of set inclusion. The reasoning method is model-theoretic. It consists of a step-by-step approach to temporal inertia, where the ordinals are used as timepoints and the upward inductive process is used as master-clock. The model is built by means of successive approximations, iterating from the origo of timepoints up to the limit ordinal. The decisional problem is then solved by deciding whether the tuple $\langle \tau, f, v\rangle$ belongs to the set $T_\Upsilon \uparrow \omega$. What is characteristic of this method, as shown in the assessment result, is the ability to build the preferred history of the ego-world game developments for $\mathcal{K}$-**IA** when the world player has a fixed strategy, leading to the proven restricted range of applicability.

In this section, the converse non-simulative method is designed. The reasoning is proof-theoretic. The decisional problem is answered immediately, by focusing the reasoning on the set of all action alternatives that are strictly relevant to the feature of interest. The method relies heavily on the order relation of the basic time structure [1], so that the time is assumed as "given" and no master-clock is necessary. The notion of relevant action alternative is as follows. Let A be an action scheduled for execution, that is exists $\langle \mathbf{s}, \mathbf{t}, A\rangle$ in SCD for some temporal expressions $\mathbf{s}$ and $\mathbf{t}$. Let $\langle s, t, A\rangle \Rrightarrow \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} S_{ij}$ be the corresponding action law in LAW, and let θ be a valuation such that $\theta = \{s/M(\mathbf{s}), t/M(\mathbf{t})\}$. An instantiated alternative of the action A (iaa), for some alternative i, is the tuple

$$\langle s\theta, t\theta, \bigcup_{j=1}^{m} S_{ij}(\theta)\rangle \in \mathcal{T} \times \mathcal{T} \times 2^{\mathcal{H}}$$

For a given feature f_j we say that $\langle s\theta, t\theta, S_{ij}(\theta)\rangle$ is relevant per f_j if and only if $\langle \tau, f_j, v\rangle \in \langle s\theta, t\theta, S_{ij}(\theta)\rangle$, that is the iaa may change the value of f_j either deductively or abductively during the time interval $[s\theta, t\theta]$. The change takes place only if the action alternative is successfully executed. Let $\mathcal{I} = \mathcal{T} \times \mathcal{T} \times 2^{\mathcal{H}}$ be the set of all iaas. The order relation $\sqsubseteq$ associated with the basic time structure applies as follows on members of $\mathcal{I}$: $\langle s_1, t_1, S_1\rangle \sqsubseteq \langle s_2, t_2, S_2\rangle$ iff $s_1 \sqsubseteq s_2$, for any $\langle s_1, t_1, S_1\rangle$ and $\langle s_2, t_2, S_2\rangle$ in $\mathcal{I}$. The order relation $\sqsubseteq$ is a partial order on $2^{\mathcal{I}}$, in fact for any I_1 and I_2 in $2^{\mathcal{I}}$ is $I_1 \sqsubseteq I_2$ iff for any occurrence $i_1 \in I_1$ and $i_2 \in I_2$ is $i_1 \sqsubseteq i_2$. Also, $2^{\mathcal{I}}$ is a complete lattice under $\sqsubseteq$, in fact the least upper bound ($\sqcup$) of a collection of subsets of $\mathcal{I}$ is their minimum element and the greatest lower bound ($\sqcap$) is their maximum element [1]. Furthermore, the function $Min\colon (2^{\mathcal{I}}; \sqsubseteq) \to (2^{\mathcal{I}}; \sqsubseteq)$ is a monotone (order-preserving) function since $I_1 \sqsubseteq I_2 \Rightarrow Min(I_1) \sqsubseteq Min(I_2)$ for any $I_1, I_2 \in 2^{\mathcal{I}}$. The Min function is also complete, since $Min(\sqcup I) = \sqcup(Min(I)) = Min(I) = \sqcup(I)$ for every directed subset I of $2^{\mathcal{I}}$. The similar property clearly holds for $Max\colon (2^{\mathcal{I}}; \sqsubseteq) \to (2^{\mathcal{I}}; \sqsubseteq)$ too.

The following is the bottom-up scheme underlying the overall top-down construction: (1) for every $\langle t, f, v\rangle$ in OBS, $\langle t, f, v\rangle$ is true in every model; (2) for every $\langle t, f, v\rangle$ that may be generated as the effect of at least one alternative of an action law without antecedents, $\langle t, f, v\rangle$ is true in at least one model; (3) for every $\langle t, f, v\rangle$ that may be generated as the effect of at least one alternative of an action law for which every antecedent is satisfied, $\langle t, f, v\rangle$ is true in at least one model; (4) for every $\langle t, f, v\rangle$ that may be generated as the abductive effect of at least one alternative of an action law for which every consequent is satisfied, $\langle t, f, v\rangle$ is true in at least one model; (5) for every $\langle t, f, v\rangle$ for which none of the above hold, $\langle t, f, v\rangle$ is true in every model. Requirements at points 3 and 4 are then weakened by allowing not every antecedent (consequent) satisfied, but a non empty subset of them, while others must correspond to unknown feature values. The scheme is actually an extension of the fixpoint scheme to the case of backward and forward reasoning about actions with alternative results.

The converse top-down construction consists in (1) analyzing the contribution of feature values at τ and at past timepoints with respect to τ, gathered under the

[1] In our specific case, namely $\mathcal{K}$-**RACi**, the time structure is linear and $(\mathcal{T}, \sqsubseteq)$ is a complete lattice too. In general, however, the basic time structure as defined in [1] is not a complete lattice; this is easy to see, as the structure can be branching and the greatest lower bound of a collection of subsets of $\mathcal{T}$ may consist of more than a single maximum element.

name of *greatest lower observations* (glo) for f at τ; (2) analyzing the contribution of feature values at future timepoints with respect to τ, gathered under the name of *least upper observations* (luo) for f at τ; (3) determining the solution to the decisional problem $\langle \tau, f, v \rangle$ as the *consistent union* of $glo(\tau, f)$ and $luo(\tau, f)$, which is properly the non-simulative $History$. A number of complete lattices are associated to Υ, sub-lattices of $(2^{\mathcal{I}}; \sqsubseteq)$ and hence referred to as *causal chains*, so that temporal priorities will determine the candidate answers $glo(\tau, f)$ and $luo(\tau, f)$.

Let O_1 and O_2 be sets of candidate partial-states for a world at time t. For example, let $\sigma_{11} = \{\langle \_, f, true \rangle, \langle \_, g, true \rangle\}$, $\sigma_{12} = \{\langle \_, f, true \rangle, \langle \_, g, false \rangle\}$, $\sigma_{21} = \{\langle \_, f, true \rangle\}$ and $\sigma_{22} = \{\langle \_, f, false \rangle\}$ be candidate partial states. If $O_1 = \{\sigma_{11}, \sigma_{12}\}$ and $O_2 = \{\sigma_{21}, \sigma_{22}\}$, the consistent union of O_1 and O_2 is the set $\{\sigma_{11} \cup \sigma_{21}, \sigma_{12} \cup \sigma_{21}\}$. The operator $\overset{*}{\cup} : 2^{\mathcal{H}} \times 2^{\mathcal{H}} \rightarrow 2^{\mathcal{H}}$ of consistent union is then defined as follows: for any O_1 and O_2 non-empty sets of candidate partial states, $O_1 \overset{*}{\cup} O_2$ is the set of all $\sigma_1 \cup \sigma_2 \in 2^{\mathcal{H}}$ such that $\sigma_1 \in O_1$, $\sigma_2 \in O_2$ and $\sigma_1 \cup \sigma_2$ is consistent. If both O_1 and O_2 are empty sets, we impose $O_1 \overset{*}{\cup} O_2$ as being $\mathcal{H}$ itself. The reason for having $\mathcal{H}$ rather than the empty set, as one would otherwise expect, is due to the underlying semantics itself, as shown by proposition 4.3 of the assessment result. The operator of non-simulative history $History : \mathcal{T} \times \mathcal{F} \rightarrow 2^{\mathcal{V}}$ is then defined as follows:

$$History(\tau, f) = \{v \in \mathcal{V} : \langle t, f, v \rangle \in glo(\tau, f) \ \overset{*}{\cup} \ luo(\tau, f)\}$$

For any scenario description Υ and tuple $\langle \tau, f, v \rangle \in \mathcal{H}$, we say that $\langle \tau, f, v \rangle$ is true if and only if $v \in History(\tau, f)$. We observe that $History$ has its values in $2^{\mathcal{V}}$, while the function *history* of the underlying semantics [2] has its values in $\mathcal{V}$. For a given feature f, in fact, the aim of NAS is to collect all the values that are intended for the feature f at the given timepoint τ. For example, in the tossing-coin scenario, the underlying semantics reports two possible developments as the effect of tossing, namely 1 and 0 for the head and the cross respectively, so that if τ is the timepoint at which the action of tossing the coin terminates, the function $history(\tau, face)$ reports the feature value 1 for one ego-world game simulation, and 0 for the other. According to NAS is $History(\tau, face) = \{1, 0\}$, for which there exists at least one intended development where the coin shows the head and another where the coin shows the cross. The problem of specifying in which development the certain feature has the certain value at the certain timepoint, it is addressed by *glo* and *luo* themselves, which definition is given below, after some due preliminaries. Before getting to the actual definition of *glo* and *luo*, the following three important functions can be given, as they are strictly defined in terms of $History$, the third of them formally defining NAS itself.

$$Fluent(f) = \{\langle t, v \rangle \in \mathcal{T} \times \mathcal{V} : v \in History(t, f)\}$$

Given a scenario description Υ and a feature $f \in \mathcal{F}$, $Fluent(f)$ is the set of all values per f on flowing time, according to Υ.

$$State(\tau) = \{\langle f, v \rangle \in \mathcal{F} \times \mathcal{V} : v \in History(\tau, f)\}$$

Given a scenario description Υ and a timepoint $\tau \in \mathcal{T}$, $State(\tau)$ is the state of the world at timepoint τ, according to Υ.

$$Comp(\Upsilon) = \{\langle t, f, v \rangle \in \mathcal{H} : \langle f, v \rangle \in State(t)\}$$

[2]We remind that *history* is the mapping H from $\mathcal{T}$ to a set of tuples in $\mathcal{F} \times \mathcal{V}$ as state, so that if t is in $\mathcal{T}$, $H(t)$ is the state of the world at timepoint t, and if f is in $\mathcal{F}$, $H(t, f)$ is the value of the feature f at timepoint t.

Given a scenario description Υ, $Comp(\Upsilon)$ is the set of all states of the world on flowing time, according to Υ. $Comp(\Upsilon)$ defines the Non-simulative Algebraic Semantics for Υ. $Comp(\Upsilon)$ is referred to as the *Completion Set* of Υ.

In order for the definition of NAS being complete, and gain that insight on the reasoning method which is needed for the subsequent assessment result, we shall now construct the $glo(\tau, f)$ and $luo(\tau, f)$ sets for the feature f at timepoint τ.

3.1 Successfully Executed Actions

We shall firstly define the method which determines whether an action alternative is *successfully* executed, that is whether features may be influenced by its execution. Let remind the method adopted for the fix-point semantics:

$$
\begin{aligned}
T_\Upsilon(I) = \{\langle \tau, f, v\rangle \in \mathcal{H}_\Upsilon :\ & \\
& \langle \tau, f, v\rangle \in OBS \text{ and } \tau = 0, \text{ or} \\
& \text{exists } \langle \mathbf{s}, \mathbf{t}, A\rangle \in SCD, \\
& \text{exists } \langle s, t, A\rangle \Rrightarrow \textstyle\bigwedge_{j=1}^{m} S_j \in LAW \text{ and} \\
& \text{exists a valuation } \theta = \{s/M(\mathbf{s}), t/M(\mathbf{t})\} \\
& \text{such that:} \\
& \langle \tau, f, v\rangle \in Consequents(s\theta, \tau, t\theta, S_j(\theta)) \\
& \text{and } Antecedents(s\theta, \tau, t\theta, S_j(\theta)) \subseteq I\ \}
\end{aligned}
$$

where

$$
\begin{aligned}
Antecedents(s, \tau, t, S) &= \{\langle s, f, \varphi(s, s, t)\rangle \in S \ :\ s \sqsubseteq \tau \sqsubseteq t\ \} \\
Consequents(s, \tau, t, S) &= \{\langle \tau, f, \varphi(s, \tau, t)\rangle \in S \ :\ s \sqsubset \tau \sqsubseteq t\ \}
\end{aligned}
$$

According to the T_Υ operator, the action A is successfully executed if and only if $Antecedents(\tau, A) \subseteq I$. Differently from T_Υ, NAS does not perform a purely deductive temporal reasoning (from premises to consequences), but a combination of abductive and deductive temporal reasoning. Furthermore the OBS part of the scenario description is intended to be a subset of the model set (point 1 of the bottom-up scheme, and point 4 of the definition of Complete Development Set [10]), so that some additional tests are required.

$$Sat\_Obs : \mathcal{I} \rightarrow \{true, false\}$$

$Sat\_Obs(\langle s, t, S\rangle)$ succeeds if and only if for any $\langle \tau, f, v\rangle \in OBS$ such that $s \sqsubseteq \tau \sqsubseteq t$ is $\langle \tau, f, v\rangle \in S$. The effect of Sat_Obs is analogous to the intersection of $[\![OBS]\!]$ with $Min(\ll, [\![LAW[SCD]]\!])$ in filter preferential entailment.

$$Sat\_Pre : \mathcal{I} \rightarrow \{true, false\}$$

$Sat\_Pre(\langle s, t, S\rangle)$ succeeds if and only if $Sat\_Obs(\langle s, t, S\rangle)$ succeeds and exists at least one $\langle s, f, v\rangle \in S$ such that $\langle \_, f, v\rangle \in glo(s, f)$. An empty set of preconditions is allowed. The remaining preconditions in S are either in glo or correspond to unknown feature values.

$$Sat\_Post : \mathcal{I} \rightarrow \{true, false\}$$

$Sat\_Post(\langle s, t, S\rangle)$ succeeds if and only if $Sat\_Obs(\langle s, t, S\rangle)$ succeeds and exists at least one $\langle t, f, v\rangle \in S$ such that $\langle \_, f, v\rangle \in luo(t, f)$. An empty set of postconditions is not allowed. The remaining postconditions in S are either in luo or correspond to unknown feature values.

3.2 Causal Chains

This part of the reasoning is inspired to the ordinal powers of the T_Υ operator. The resulting definition is affected by a number of extensions which purpose is to allow for a possibly unspecified initial state of the world, observations at later states than the initial one, continuous time, continuous change and disjunction in the action laws of possibly concurrent and independent actions. A number of recursive calls to that part of the definition of NAS which is entitled for backward temporal reasoning are also involved (via Sat), as it has been recognized [10] that *both* backward *and* forward reasoning are needed in order to handle the full class. The method neatly differs from the fixpoint operator as it does not build the partial model from the origo of timepoints up to τ. The method in fact exploits the order relation with the given temporal structure, rather than beating time with the master clock. This is done by selecting all the relevant action alternatives that have successfully occurred before ($\sqsubseteq$) τ, and therefore influenced the value of the feature of interest. The collection of all such action alternatives is called the set of all the *lower causal chains* for f at τ, or $lcc(\tau, f)$.

$$lcc : \mathcal{T} \times \mathcal{F} \rightarrow 2^{\mathcal{I}}$$

Given a timepoint τ and a feature f, $lcc(\tau, f)$ is the set of all iaas $\langle s, t, S\rangle$ such that $\langle s, t, S\rangle \in Max(\mathcal{P}(\tau, f); \sqsubseteq)$ union the set of all iaas $\langle s_2, t_2, S_2\rangle$ such that $\langle\langle s_1, t_1, S_1\rangle, \langle s_2, t_2, S_2\rangle\rangle \in \mathcal{P}(\tau, f) \times \mathcal{P}(\tau, f)$, is $t_2 \sqsubseteq s_1$ and for all $\langle s_1, f, v\rangle \in S_1$ is $\langle t_2, f, v\rangle \notin S_2$. Given a timepoint τ and a feature f, $\mathcal{P}(\tau, f)$ is the set of all iaas $\langle s\theta, t\theta, S\theta\rangle$ of actions A such that (1) $\langle \mathbf{s}, \mathbf{t}, A\rangle \in SCD$ and either $t\theta \sqsubset \tau$ or $s\theta \sqsubset \tau \sqsubseteq t\theta$, (2) $\langle s\theta, t\theta, S\theta\rangle$ is relevant per f and (3) $\langle s\theta, t\theta, S\theta\rangle$ is successfully executed. If conditions 1b,2 and 3 hold, we say that $f \in Influence(\tau, \langle s\theta, t\theta, S(\theta)\rangle)$.

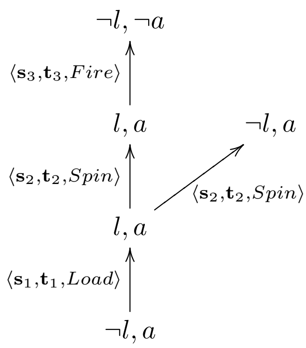
$$glo : \mathcal{T} \times \mathcal{F} \rightarrow 2^{\mathcal{H}}$$

Given a timepoint τ and a feature f, $glo(\tau, f)$ is the union of the following three disjoint sets: $glo_{obs}(\tau, f)$, $glo_{pas}(\tau, f)$ and $glo_{cas}(\tau, f)$. $glo_{pas}(\tau, f)$ is the set of all tuples $\langle t, f, v\rangle \in \mathcal{H}$ such that $\langle s, t, S\rangle \in lcc(\tau, f)$, $t \sqsubset \tau$ and $\langle t, f, v\rangle \in S$. $glo_{cas}(\tau, f)$ is the set of all tuples $\langle \tau, f, v\rangle \in \mathcal{H}$ such that $\langle s, t, S\rangle \in lcc(\tau, f)$, $s \sqsubset \tau \sqsubseteq t$ and $\langle \tau, f, v\rangle \in S$. Finally, $glo_{obs}(\tau, f)$ is the set of all tuples $\langle t, f, v\rangle \in \mathcal{H}$ such that $\langle t, f, v\rangle \in OBS$, $t \sqsubset \tau$ and the following restrictions apply. For all $\langle s, f', v'\rangle \in S$ such that $\langle s, \_, S\rangle$ is relevant per f, it must be the case that $glo(s, f') \setminus \{\langle \_, f', v'\rangle\} \neq \varnothing$ where $f' \neq f$, i.e. the relevant iaa has failed. This ensures that no relevant iaas were successfully executed during $[t, \tau]$ that may have influenced f. If no relevant iaas are available, then no influencing actions occur during $[t, \tau]$, so that inertia applies and $\langle t, f, v\rangle \in glo_{obs}(\tau, f)$. If relevant iaas are available and they all have an empty set of preconditions (i.e. the set of all $\langle s, f', v'\rangle \in S$ is empty), then they all were successfully executed (point 2 of the bottom-up scheme), the value for f as from the tuple $\langle t, f, v\rangle$ is then influenced during $[t, \tau]$ and $\langle t, f, v\rangle \notin glo_{obs}(\tau, f)$.

This is all there is to it. The definition of ucc ad luo are symmetrical to lcc and glo. The only difference worth mentioning is that current activities are not involved, so that the set of all upper causal chains (ucc) for f at τ consists of all and only those relevant action alternatives $\langle s, t, S\rangle$ that have successfully occurred after τ ($\tau \sqsubseteq s\theta$) and therefore exert backward abductive influence on the feature of interest. Figure 1 shows the result of applying the given definition to the celebrated Russian Shooting Scenario. The definition applies to uncountably many different scenarios, to which the assessment result in the next section provides exhaustive insight.

Figure 1: The Russian Shooting Scenario

A turkey is initially *alive* and the gun is not *loaded*. Successively, a *load*, a *spin* of the chamber and a *shoot* event occur. One expects the turkey to be dead afterwards, if the gun was loaded at the time of firing, alive otherwise.



$glo_{obs}(\mathbf{t}_3, l) = \varnothing$
$glo_{pas}(\mathbf{t}_3, l) = \{\langle \mathbf{t_2}, l, 0 \rangle\}$
$glo_{cas}(\mathbf{t}_3, l) = \{\langle \mathbf{t}_3, l, 0 \rangle\}$
$glo(\mathbf{t}_3, l) = \{\langle \mathbf{t}_3, l, 0 \rangle, \langle \mathbf{t_2}, l, 0 \rangle\}$

$glo_{obs}(\mathbf{t}_2, l) = \varnothing$
$glo_{pas}(\mathbf{t}_2, l) = \varnothing$
$glo_{cas}(\mathbf{t}_2, l) = \{\langle \mathbf{t_2}, l, 0 \rangle, \langle \mathbf{t_2}, l, 1 \rangle\}$
$glo(\mathbf{t}_2, l) = \{\langle \mathbf{t}_2, l, 1 \rangle, \langle \mathbf{t_2}, l, 0 \rangle\}$

$glo_{obs}(\mathbf{t}_1, l) = \varnothing$
$glo_{pas}(\mathbf{t}_1, l) = \varnothing$
$glo_{cas}(\mathbf{t}_1, l) = \{\langle \mathbf{t_1}, l, 1 \rangle\}$
$glo(\mathbf{t}_1, l) = \{\langle \mathbf{t_1}, l, 1 \rangle\}$

$glo_{obs}(0, l) = \{\langle 0, l, 0 \rangle\}$
$glo_{pas}(0, l) = \varnothing$
$glo_{cas}(0, l) = \varnothing$
$glo(0, l) = \{\langle 0, l, 0 \rangle\}$

$glo_{obs}(\mathbf{t}_3, a) = \{\langle 0, a, 1 \rangle\}$
$glo_{pas}(\mathbf{t}_3, a) = \varnothing$
$glo_{cas}(\mathbf{t}_3, a) = \{\langle \mathbf{t}_3, a, 0 \rangle\}$
$glo(\mathbf{t}_3, a) = \{\langle \mathbf{t}_3, a, 0 \rangle, \langle 0, a, 1 \rangle\}$

$glo_{obs}(\mathbf{t}_2, a) = \{\langle 0, a, 1 \rangle\}$
$glo_{pas}(\mathbf{t}_2, a) = \varnothing$
$glo_{cas}(\mathbf{t}_2, a) = \varnothing$
$glo(\mathbf{t}_2, a) = \{\langle 0, a, 1 \rangle\}$

$glo_{obs}(\mathbf{t}_1, a) = \{\langle 0, a, 1 \rangle\}$
$glo_{pas}(\mathbf{t}_1, a) = \varnothing$
$glo_{cas}(\mathbf{t}_1, a) = \varnothing$
$glo(\mathbf{t}_1, a) = \{\langle 0, a, 1 \rangle\}$

$glo_{obs}(0, a) = \{\langle 0, a, 1 \rangle\}$
$glo_{pas}(0, a) = \varnothing$
$glo_{cas}(0, a) = \varnothing$
$glo(0, a) = \{\langle 0, a, 1 \rangle\}$

4 Comparative Assessment

According to the *Features and Fluents* systematic methodology, we shall now proceed to the assessment of the range of applicability of NAS. To keep in short with the notation, we write $\langle t, f, v\rangle \in \Sigma(\Upsilon)$ instead of "exists an interpretation $\langle M, H\rangle$ such that the development $\langle \mathcal{B}, M, H, \mathcal{A}, \mathcal{C}\rangle$ is in $Mod(\Upsilon)$ and $H(t, f) = v$".

Let $\mathcal{F}_\Upsilon$ be the set of all features occurring in Υ. By comparison with the simulative approach, the first result shows that the non-simulative approach is correctly applicable to any Υ in $\mathcal{K}$**sp-IAd**.

Proposition 4.1 *For any $\Upsilon \in \mathcal{K}\mathbf{sp\text{-}IAd}$ and for any $\langle t, f, v\rangle \in \mathcal{T} \times \mathcal{F}_\Upsilon \times \mathcal{V}$, the following relation holds: $\langle t, f, v\rangle \in \Sigma_{\mathcal{K}\mathbf{sp\text{-}IAd}}(\Upsilon) \Leftrightarrow \langle t, f, v\rangle \in Comp(\Upsilon)$.*

Proof. The relation $\langle t, f, v\rangle \in \Sigma_{\mathcal{K}\mathbf{sp\text{-}IAd}}(\Upsilon) \Leftrightarrow \langle t, f, v\rangle \in T_\Upsilon \uparrow \omega$ holds by [1, theorem 4]. We need to prove that $\langle t, f, v\rangle \in T_\Upsilon \uparrow \omega \Leftrightarrow \langle t, f, v\rangle \in Comp(\Upsilon)$. Since Υ is in $\mathcal{K}$**sp-IAd**, we assumed complete knowledge about the initial state of the world. This knowledge is represented by means of observations $\langle 0, f, v\rangle$ in the OBS part of Υ. Per definition is $T_\Upsilon \uparrow 0 = \{\langle t, f, v\rangle \in \mathcal{B}_\Upsilon : \langle t, f, v\rangle \in OBS \wedge t = 0\}$ and $glo(0, f) = glo_{obs}(0, f) \cup glo_{pas}(0, f) \cup glo_{cas}(0, f) = \{\langle t, f, v\rangle \in OBS : t = 0\} \cup \varnothing \cup \varnothing$. Then $\langle t, f, v\rangle \in T_\Upsilon \uparrow 0$ iff $\langle t, f, v\rangle \in glo(0, f)$. Suppose $\langle t, f, v\rangle \in T_\Upsilon \uparrow (\tau - 1)$ iff $\langle t, f, v\rangle \in glo(\tau - 1, f)$. We shall prove that $\langle t, f, v\rangle \in T_\Upsilon \uparrow \tau$ iff $\langle t, f, v\rangle \in glo(\tau, f)$. One of the followings may hold:

- $SCD = \varnothing$. Then is $T_\Upsilon \uparrow \tau = T_\Upsilon \uparrow (\tau - 1) = \ldots = T_\Upsilon \uparrow 0$. As $glo_{cas}(\tau, f) = \varnothing$, then is $glo(\tau, f) = glo(\tau - 1, f)$.
- $SCD \neq \varnothing$. Let $\langle \mathbf{s}, \mathbf{t}, A\rangle$ be one of its elements. If $Antecedents(\tau, A) \subseteq T_\Upsilon \uparrow (\tau - 1)$ then $T_\Upsilon \uparrow \tau = T_\Upsilon \uparrow (\tau - 1) \cup Consequents(\tau, A)$; otherwise is $T_\Upsilon \uparrow \tau = T_\Upsilon \uparrow (\tau - 1)$. Concerning NAS, exists $\langle s, t, A\rangle \Rrightarrow \bigwedge_{j=1}^{m} S_j$ in LAW and exists a valuation $\theta = \{s/M(\mathbf{s}), t/M(\mathbf{t})\}$ such that $f \in Influence(\tau, \langle s\theta, t\theta, S_j(\theta)\rangle)$ and $glo_{cas}(\tau, f_j) = \{\langle \tau, f, v\rangle \in \mathcal{H} : \langle s, t, S\rangle \in lcc(\tau, f), s \sqsubset \tau \sqsubseteq t$ and $\langle \tau, f, v\rangle \in S\} = Consequents(\tau, A)$; otherwise is $glo(\tau, f_j) = glo(\tau - 1, f_j)$ because $glo_{cas}(\tau, f_j) = \varnothing$.

In each of the above situations is $glo(\tau, f) \overset{*}{\cup} luo(\tau, f) = glo(\tau, f)$, in fact for any $\langle t', f, v'\rangle \in luo(\tau, f)$ such that $\langle t', f, v'\rangle \notin glo(\tau, f)$, then also $\langle t', f, v'\rangle \notin T_\Upsilon \uparrow \tau$. ■

The following result shows NAS adequate for an image-level world with continuous time, continuous change and alternative results of actions.

Proposition 4.2 *For any $\Upsilon \in \mathcal{K}\mathbf{sp\text{-}RA}$ and for any $\langle \tau, f, v\rangle \in \mathcal{T} \times \mathcal{F}_\Upsilon \times \mathcal{V}$, the following relation holds: $\langle \tau, f, v\rangle \in \Sigma_{\mathcal{K}\mathbf{sp\text{-}RA}}(\Upsilon) \Leftrightarrow \langle \tau, f, v\rangle \in Comp(\Upsilon)$.*

Proof. By proposition 4.1, the relation holds for the sub-class $\mathcal{K}$**sp-IAd**. However, the SCD part of the scenario description is a discrete sub-lattice of $(\mathcal{D}, \sqsubseteq)$ in fact, per definition, every action has a non-empty length. As such, there are at most as many timepoints in SCD as the natural numbers. Then the relation holds also for that subset of the intended model set for scenario descriptions in $\mathcal{K}$**sp-RAd** where strict inertia in continuous time and discrete deterministic change are allowed. In order to capture the full $\mathcal{K}$**sp-RA** class we need to show that NAS properly handles the continuous change and the alternative results of actions. Let $\langle M, H\rangle$ be an intended model in $\Sigma(\Upsilon)$ such that exists $\langle \mathbf{s}, \mathbf{t}, A\rangle$ in SCD, exists $\langle s, t, A\rangle \Rrightarrow \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} S_j$ in LAW and exists a valuation $\theta = \{s/M(\mathbf{s}), t/M(\mathbf{t})\}$ such that $s\theta \sqsubset \tau \sqsubseteq t\theta$, $f = f_j$, $f_j \in Infl(A, H(s\theta))$, and $H(\tau, f_j) = \varphi_{ij}(s\theta, \tau, t\theta) = v$, that is $\langle \tau, f_j, v\rangle \in H \rhd S_{ij}(\theta)$. The following holds:

$$\begin{array}{rcl} f_j \in Infl(A, H(s\theta)) & \Leftrightarrow & f_j \in Influence(\tau, \langle s\theta, t\theta, S_{ij}(\theta)\rangle) \\ & \Rightarrow & \langle \tau, f_j, v\rangle \in glo_{cas}(\tau, f_j) \\ & \Rightarrow & \langle \tau, f_j, v\rangle \in glo(\tau, f_j) \end{array}$$

Therefore, by definition of consistent union, $v \in History(\tau, f_j)$, $\langle f_j, v\rangle \in State(\tau)$ and $\langle t, f_j, v\rangle \in Comp(\Upsilon)$. ∎

Proposition 4.3 *For any* Υ *in* $\mathcal{K}$-**RACi**, $\mathcal{F}_\Upsilon = \varnothing$ *implies* $Comp(\Upsilon) = \mathcal{H}$.

Proof. Let agree that Υ is "empty per f" if and only if Υ contains no observations in OBS and occurrences of actions in SCD involving the feature f. The set of all features f such that Υ is empty per f is then $\mathcal{F} \setminus \mathcal{F}_\Upsilon$. The thesis is a straightforward corollary of the following statement: $\langle \tau, f, v\rangle \in \Sigma_{\mathcal{K}\text{-}\mathbf{RACi}}(\Upsilon) \Leftrightarrow \langle \tau, f, v\rangle \in Comp(\Upsilon)$ for any $\langle \tau, f, v\rangle \in \mathcal{T} \times \mathcal{F} \setminus \mathcal{F}_\Upsilon \times \mathcal{V}$. Let first prove that for any $\langle \tau, f, v\rangle \in \mathcal{T} \times \mathcal{F} \setminus \mathcal{F}_\Upsilon \times \mathcal{V}$ is $\langle \tau, f, v\rangle \in \Sigma_{\mathcal{K}\text{-}\mathbf{RACi}}(\Upsilon)$. Per definition of ego-world game, the game starts with the board in an initial configuration, where the initial state $H(0)$ is a certain non-deterministically given σ_0. Due to the non-determinism, a whatever value v could be assigned to f. During the game the world will persist in that value per f until an action is invoked by the ego that influences the feature. As Υ is empty per f by hypothesis, no such action will be invoked. At the end of the game, the value per f will be the non-deterministically assigned initial value v. As no specific choice is made on this initial value, no tuple $\langle 0, f, v\rangle$ appears in the OBS part of Υ that may restrict the number of models, so that for any $\langle \tau, f, v\rangle \in \mathcal{T} \times \mathcal{F} \setminus \mathcal{F}_\Upsilon \times \mathcal{V}$ is $\langle 0, f, v\rangle \in \Sigma_{\mathcal{K}\text{-}\mathbf{RACi}}(\Upsilon)$ by point 4 of the definition of complete development set. On the other hand, for any $\langle \tau, f, v\rangle \in \mathcal{T} \times \mathcal{F} \setminus \mathcal{F}_\Upsilon \times \mathcal{V}$ is also $\langle \tau, f, v\rangle \in Comp(\Upsilon)$. In fact, as $f \in \mathcal{F} \setminus \mathcal{F}_\Upsilon$, both $glo(\tau, f)$ and $luo(\tau, f)$ are trivially empty, so that $\langle \tau, f, v\rangle \in Comp(\Upsilon)$ via the definition of *History* and, in particular, via the operator of consistent union. ∎

The following is the main assessment result. It shows that for any scenario description Υ in $\mathcal{K}$-**RACi** algebraic and underlying semantics coincide, that is the model set obtained via NAS is equal to the intended model set.

Theorem 4.1 (soundness and completeness) *For any* $\Upsilon \in \mathcal{K}$-**RACi** *and for any* $\langle \tau, f, v\rangle \in \mathcal{H}$*, the following relation holds:*

$$\langle \tau, f, v\rangle \in \Sigma_{\mathcal{K}\text{-}\mathbf{RACi}}(\Upsilon) \Leftrightarrow \langle \tau, f, v\rangle \in Comp(\Upsilon)$$

Proof. By proposition 4.3, the relation holds for any $\langle t, f, v\rangle$ in $\mathcal{T} \times \mathcal{F} \setminus \mathcal{F}_\Upsilon \times \mathcal{V}$. Let now prove the relation for features in $\mathcal{F}_\Upsilon$, where $\mathcal{F}_\Upsilon \neq \varnothing$. By proposition 4.2, via the *glo* function, the relation holds for any Υ in $\mathcal{K}$**sp-RA**, that is for pure prediction problems. We need to prove the relation for chronicles in $\mathcal{K}$-**RACi** $\setminus \mathcal{K}$**sp-RA**, that is for pure post-diction and pre-post-diction problems. The former case is straightforward. By construction, in fact, the definition of *luo* is *symmetrical* to that of *glo*, so that proposition 4.2 proves the relation for pure post-diction problems too, although with similar converse technique. Concerning pre-post-diction problems, the following case holds according to the definition of image-level world and Complete Development Set (we strictly refer to [1]): exists $\langle \mathbf{s}, \mathbf{t}, A\rangle$ in SCD and an intended model $\langle M, H\rangle$ such that $M(\mathbf{s}) \sqsubset \tau \sqsubseteq M(\mathbf{t})$, $f \in Infl(A, H(M(\mathbf{s})))$, $H(\tau, f) = v$, and exists $\langle M(\mathbf{s}), g, w\rangle$ precondition of A such that $H(0, g) = H(M(\mathbf{s}), g) = w$ and, since $\Upsilon \cup \{\langle \tau, f, v\rangle\}$ is a pre-post-diction problem, Υ is empty per g for any t in $[0, M(\mathbf{s}))$. Therefore, the problem of determining whether $\langle \tau, f, v\rangle$ belongs to $Comp(\Upsilon)$ reduces to the problem of determining whether $\langle M(\mathbf{s}), g, w\rangle$ belongs to $Comp(\Upsilon)$, which is a pure post-diction problem. ∎

It is now assessed that the range of applicability of NAS is the full $\mathcal{K}$-**RACi** class of reasoning problems, and $History(\tau, f)$ is then the set of all values v of the feature f at timepoint τ such that $\langle \tau, f, v \rangle$ is true in at least one intended model. The following corollary holds:

Definition 4.1 *(Sandewall, [10])*
Let Υ and Υ' be scenario descriptions.

- Υ entails Υ', *written* $\Upsilon \Vdash \Upsilon'$, *iff* $Mod(\Upsilon) \subseteq Mod(\Upsilon')$*;*
- Υ *is* equivalent *to* Υ', *written* $\Upsilon \cong \Upsilon'$, *iff* $Mod(\Upsilon) = Mod(\Upsilon')$*;*
- Υ *is* consistent *with* Υ', *written* $\Upsilon \,\|\!\diamond \Upsilon'$, *iff* $Mod(\Upsilon) \cap Mod(\Upsilon') \neq \varnothing$*.*

Corollary 4.1

- $\Upsilon \Vdash \Upsilon' \Leftrightarrow Comp(\Upsilon) \subseteq Comp(\Upsilon')$.
- $\Upsilon \cong \Upsilon' \Leftrightarrow Comp(\Upsilon) = Comp(\Upsilon')$.
- $\Upsilon \,\|\!\diamond \Upsilon' \Leftrightarrow Comp(\Upsilon) \cap Comp(\Upsilon') \neq \varnothing$.

According to the *Features and Fluents* systematic approach it is now possible to compare the range of applicability of the Non-simulative Algebraic Semantics with that of the language $\mathcal{A}$ and the Abductive Logic Programming, as well as all the methods for which an assessment result is available.

The following results are available from the literature. In [10] the PMON entailment (Pointwise Minimization of Occlusion with No-change Premises) has been designed and proven sound and complete with respect to the full $\mathcal{K}$-**IA** family. PMON is the best entailment method defined in [10]; it is equivalent to the Chronological Assignment and Minimization of Occlusion and Change (CAMOC) and subsumes the Prototypical Global Minimization (PGM, $\mathcal{K}$**r-IsAz**), the Original Chronological Minimization (OCM, $\mathcal{K}$**sp-IsAd**), the Prototypical Chronological Minimization (PCM, $\mathcal{K}$**p-IAex**), the Prototypical Chronological Minimization with Filtering (PCMF, $\mathcal{K}$-**IAex**), the Global Minimization of Occlusion with No-change Premises (GMOC, $\mathcal{K}$**r-IsA**), and the Chronological Minimization of Occlusion and Change (CMOC, $\mathcal{K}$-**IAe**). In [12] the language $\mathcal{A}$ [6] has been proven sound and complete with respect to the $\mathcal{K}$-**IbsAd** family, and in [2] the Abductive Logic Programming has been proven sound and complete with respect to the same $\mathcal{K}$-**IbsAd** family.

With no need to getting straight to the details of each reasoning method (which has been already fulfilled when proving each assessment result), the comparison reveals the non-simulative algebraic semantics both epistemologically and ontologically stronger than all the cited methods. In particular:

- the Abductive Logic Programming (ALP) is epistemologically equivalent to NAS, as both range on $\mathcal{K}$, but ontologically weaker, as ALP ranges on problems where actions operate in a single time-step (**Is**) and all features are two-valued (**Ib**), while NAS has no such restrictions;
- due to [2, theorem 5], the comparison of NAS against the Action Description Language $\mathcal{A}$ leads to the same conclusions;
- the PMON entailment is epistemologically equivalent to NAS, as both range on $\mathcal{K}$, but ontologically weaker, as PMON ranges on problems where actions have discrete trajectories that take place in integer time, while NAS has no such restrictions.

The systematic approach is useful not only for classifying and comparing the different logics, but also for immediately establishing which problems can be solved by each classified logic. For any given reasoning problem, it is possible to say whether a given reasoning method will solve it or not by saying whether the problem itself belongs to the class for which the method is provably sound. A list of popular reasoning problems and their relative classification is given in [10].

5 Computability

Let consider the problem of computing the completion set of Υ.

Proposition 5.1 *For any* Υ *in* $\mathcal{K}$-$\mathbf{IA}$, $Comp(\Upsilon)$ *is a recursively enumerable set.*

PROOF. By definition, for any $\tau \in \mathbb{N}$, $State(\tau)$ is the set of all tuples $\langle f, v \rangle \in \mathcal{F} \times \mathcal{V}$ such that $v \in History(\tau, f)$. The set $History(\tau, f)$ is finite; in fact, if $a(i)$ is the number of action alternatives of the action i and m is the number of action occurrences in SCD, the number of relevant and successfully executed action alternatives for f at τ is at most $\sum_{i=1}^{m} a(i)$. As timepoints are recursively enumerable, as they are natural numbers, we conclude that for any Υ in $\mathcal{K}$-$\mathbf{IA}$, $Comp(\Upsilon)$ is a recursively enumerable set. ∎

As real numbers are not recursively enumerable, the similar proposition does not hold for Υ in $\mathcal{K}$-$\mathbf{RACi}$. The set $Comp(\Upsilon)$ is so extensive that no algorithm could possibly list all its elements, so that *intended models for* $\mathcal{K}$-$\mathbf{RACi}$ *are not effectively computable models*. But would we really ask for more information that we could ever hope to use? The problem of computing the *whole* intended model set for scenario descriptions in $\mathcal{K}$-$\mathbf{RACi}$ does not seem indeed a realistic problem, and neither does for $\mathcal{K}$-$\mathbf{IA}$. In resource bounded situations, in fact, Υ is a finite scenario description, so that the OBS, SCD and LAW components of Υ are finite sets too, as well as the set $\mathcal{T}_\Upsilon$ of all timepoints occurring in Υ and the set $\mathcal{F}_\Upsilon$ of all features occurring in Υ. As each action occurrence in SCD has a finite duration, it follows that in any model of the scenario description changes due to an action are limited to a finite interval in time. Since no other events are allowed in $\mathcal{K}$-$\mathbf{RACi}$ besides those specified in SCD, the set of all intervals where persistence arise is a finite set too. $History(\tau, f)$ then consists of a finite set of feature values and $State_\Upsilon(\tau)$, i.e. $State(\tau)$ for those features occurring in Υ, is a finite set too. The problem of computing $Comp(\Upsilon)$ then reduces to computing a finite number of finite states:

$$Ker(\Upsilon) = \{\langle \tau, f, v \rangle \in \{0\} \cup \mathcal{T}_\Upsilon \times \mathcal{F}_\Upsilon \times \mathcal{V} : \langle f, v \rangle \in State_\Upsilon(\tau)\}$$

We will refer to $Ker(\Upsilon)$ as the *kernel* of the completion set of Υ, namely the most representative of all decidable subsets of $Comp(\Upsilon)$. When needed, and apart from the kernel, the value of any feature at any given timepoint can always be computed via the $History$ function, where *computable partial fluents* [1] are used for characterizing features during time periods where their values change. Computing the set of all timepoints for which a given feature f has a given value v will certainly require standard numerical-analysis techniques over those periods where actions influencing f are performed:

$$Holds\_at(f, v) = \{\tau \in \mathcal{T} : v \in History(\tau, f)\}$$

Adopting the non-simulative algebraic semantics as proof procedure, a meta-theoretic extension of the Horn Clause Logic has been defined using the non-ground representation of the object-level variables. The meta-interpreter consists in the

Horn clause representation of the proof procedure. It is built on top of the EPSILON system (ESPRIT project P-530 [4]), and is executable as a conventional logic program by the SLD-resolution rule. The resulting calculus is domain independent and, due to theorem 4.1, class dependent. A companion web-site to this work can be accessed via the author's Internet address at `http://www.dcs.qmw.ac.uk/~sb`.

6 Conclusion

This paper continues a new line of work on causal reasoning in Logic Programming. The new line begins in [2], where the *Features and Fluents* research methodology is extended to Logic Programming and the reasoning method of van Emden and Kowalski's simulative algebraic semantics for the Horn Clause Logic is proven adequate for solving non-monotonic temporal-reasoning problems within the $\mathcal{K}$**sp-IAd** class of assumptions. In this paper, the converse non-simulative method is designed and its range of applicability is proven to be the full $\mathcal{K}$-**RACi** class, a superclass of $\mathcal{K}$**sp-IAd**. The relatively straightforward comparative assessment also reveals the non-simulative method epistemologically equivalent and ontologically stronger than the Abductive Logic Programming, the Action Description Language $\mathcal{A}$ and the $PMON$ entailment.

The new line of work goes beyond the current and past work on causal reasoning in Logic Programming in several ways. First, it tackles the issue of explicitly and formally defining the simplifying assumptions that are made when representing problems. The issue is new to the several Event Calculus approaches and to Logic Programming itself, where these assumptions are often left implicit. Second, it tackles the issue of *classifying* and *comparing* the reasoning methods according to their range of applicability, formally assessed by means of a soundness and completeness result of the semantics with respect to the underlying semantics that formally defines the simplifying assumptions. The original problem of establishing whether a given reasoning method solves a given reasoning problem is then reduced to saying whether the problem itself belongs to the class for which the method is provably sound. Finally, the overall research task neatly differs from the classical one, as it encourages the design of logic programming languages for solving whole classes of reasoning problems, and strongly discourages the drawing of guidelines or recipes for which logic programs are often *claimed* to solve non-trivial problems.

Work continues along this line. As noted at page 10, the comparison is only possible between methods for which an assessment result is available. At the time of writing, no such general assessment result is available for the more recent Action Description Languages [7], for the Event Calculus [8] and its variants, like the Circumscriptive Event Calculus [11], and for the Logic Programming language GOLOG [9]. The general assessment would be useful not only in order to compare these methods against NAS, but also against other formalisms that do not strictly belong to the Logic Programming approach, as shown in this paper. At the time of writing, the range of applicability of the Event Calculi is under assessment.

Acknowledgements: I am grateful to Erik Sandewall for the helpful clarifications. A subtle issue of the underlying semantics for $\mathcal{K}$-**IA**, inherited by $\mathcal{K}$-**RACi**, now justifies proposition 4.3. I am also grateful to Giorgio Levi for the helpful discussions and comments on an earlier draft of this paper.

References

[1] S. Brandano. $\mathcal{K}$-**RACi**. In *Non-monotonic Reasoning, Action and Change, Proceedings of the International (IJCAI) Workshop*, pages 9–16, Stockholm, Sweden, August 1999.

[2] S. Brandano. Features and Fluents for Logic Programming: Simulative Algebraic Semantics. Manuscript, 2000.

[3] S. Brandano. On the meta-theoretic approach to non-monotonic reasoning, its extension to the continuum case and relation with classical Newtonian Mechanics. Manuscript, 2000.

[4] P. Coscia, P. Franceschi, G. Levi, G. Sardu, and L. Torre. Meta-level definition and compilation of inference engines in the EPSILON Logic Programming environment. In R. Kowalski and K. Bowen, editors, *International Conference on Logic Programming, Proceedings*, pages 359–373. MIT Press, 1988.

[5] M. Denecker and D. De Schreye. Representing incomplete knowledge in Abductive Logic Programming. *Journal of Logic and Computation*, 5(5):553–578, 1995.

[6] M. Gelfond and V. Lifschitz. Representing Actions and Change by Logic Programs. *Journal of Logic Programming*, 17, 1993.

[7] M. Gelfond and V. Lifschitz. Action Languages. *Electronic Transactions on Artificial Intelligence (ETAI)*, 3(16), 1998. http://www.ep.liu.se/ea/cis/1998/016/.

[8] R. A. Kowalski and M. Sergot. A Logic-based Calculus of Events. *New Generation Computing*, 4(1):67–95, 1986.

[9] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. GOLOG: a Logic Programming language for Dynamic Domains. *Journal of Logic Programming*, 31(1-3):59–83, 1997.

[10] E. Sandewall. *Features and Fluents: The Representation of Knowledge about Dynamical Systems*, volume 1. Oxford University Press, 1994.

[11] M. Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press, 1997.

[12] M. Thielscher. An Analysis of Systematic Approaches to Reasoning about Actions and Change. In *Artificial Intelligence : Methodology, Systems, Applications, Proceedings of the International Conference (AIMSA)*, 1994.