Reasoning about actions with $\mathcal{EL}^\perp$ ontologies and temporal answer sets

Laura Giordano[1], Alberto Martelli[2] and Daniele Theseider Dupré [1]

[1]*DISIT - Università del Piemonte Orientale, Alessandria, Italy*
[2]*Dip. di Informatica - Università di Torino, Italy*

Abstract
We propose an approach based on Answer Set Programming for reasoning about actions with domain descriptions including ontological knowledge, expressed in the lightweight description logic $\mathcal{EL}^\perp$. We consider a temporal action theory, which allows for non-deterministic actions and causal rules to deal with ramifications, and whose extensions are defined by temporal answer sets. We provide conditions under which action consistency can be guaranteed with respect to an ontology, by a polynomial encoding of an action theory extended with an $\mathcal{EL}^\perp$ knowledge base (in normal form) into a temporal action theory.

1. Introduction

The integration of description logics and action formalisms has gained a lot of interest in the past years [5, 4, 12, 1]. In this paper we explore the combination of a temporal action logic [23] and an $\mathcal{EL}^\perp$ knowledge bases, with the aim of allowing reasoning about action execution in the presence of the constraints given by an $\mathcal{EL}^\perp$ ontology.

As usual in many formalisms integrating description logics and action languages [5, 6, 12, 1], we regard inclusions in the KB as state constraints of the action theory, which we expect to be satisfied in the state resulting after action execution. In the literature of reasoning about actions it is well known that causal laws and their interplay with domain constraints are crucial for solving the ramification problem [33, 31, 34, 13, 18, 25]. In case knowledge about the domain is expressed in a description logic, the issue has been considered, e.g., in [4] where causal laws are used to ensure the consistency with the TBox of the resulting state, after action execution. For instance, given a TBox containing $\exists Teaches.Course \sqsubseteq Teacher$, and an ABox (i.e., a set of assertions on individuals) containing the assertion $Course(math)$, an action which adds the assertion $Teaches(john, math)$, without also adding $Teacher(john)$, will not give rise to a consistent next state with respect to the knowledge base. The addition of the causal law $\Box(Teacher(john) \leftarrow Teaches(john, math) \wedge Course(math))$ would enforce, for instance, the above TBox inclusion to be satisfied in the resulting state.

The approach proposed by Baader et al. [4] uses causal relationships to deal with the ramification problem in an action formalism based on description logics, and it exploits a semantics of actions and causal laws in the style of Winslett's [35] and McCain and Turner's [33] fixpoint

laura.giordano@uniupo.it (L. Giordano); mrt@di.uniuto.it (A. Martelli); dtd@uniupo.it (D. T.)

semantics. In this paper, we aim at extending this approach to reason about actions with an $\mathcal{EL}^\perp$ ontology with *temporal answer sets*. Reasoning about actions with temporal answer sets has been proposed in [21, 23] by defining a temporal logic programming language for reasoning about *complex actions* and *infinite computations*. This action language, besides the usual LTL operators, allows for general Dynamic Linear Time Temporal Logic (DLTL) formulas to be included in domain descriptions to constrain the space of possible extensions. In [23] a notion of Temporal Answer Set for domain descriptions is introduced, as a generalization of the usual notion of Answer Set, and a translation of domain descriptions into standard Answer Set Programming (ASP) is provided, by exploiting *bounded model checking techniques* for the verification of DLTL constraints, extending the approach developed by Helianko and Niemela [27] for bounded LTL model checking with Stable Models. An alternative ASP translation of this temporal action language has been investigated in [22, 24], by proposing an approach to bounded model checking which exploits the Büchi automaton construction while searching for a counterexample, with the aim of achieving completeness. Our temporal action logic has been shown to be strongly related to extensions of the $\mathcal{A}$ language [17, 9, 14, 7, 25]. The temporal formalism is also related to the recent temporal extension of Clingo, *telingo* dealing with finite computations [10].

The paper studies *extended temporal action theories*, combining the temporal action logic mentioned above with an $\mathcal{EL}^\perp$ knowledge base. It is shown that, for $\mathcal{EL}^\perp$ knowledge bases in normal form, the consistency of the action theory extensions with the ontology can be verified by adding to the action theory a set of causal laws and state constraints, by exploiting a fragment of the materialization calculus by Krötzsch [29]. Furthermore, sufficient conditions on the action theory can be defined to repair the states resulting from action execution and guarantee consistency with TBox. To this purpose, for each inclusion axiom in TBox, a suitable set of causal laws can be added to the action theory. Our approach provides a polynomial encoding of an extended action theory, with an $\mathcal{EL}^\perp$ knowledge base in normal form, into the language of the (DLTL) temporal action logic studied in [23]. The proof methods for this temporal action logic, based on bounded model checking, can then be exploited for reasoning about actions in the extended action theory.

A preliminary version of this work, which does not exploit a temporal action language, has been presented in CILC 2016 [20].

2. The description logic $\mathcal{EL}^\perp$

We consider a fragment of the logic $\mathcal{EL}^{++}$ [2] that, for simplicity of presentation, does not include role inclusions and concrete domains. The fragment, let us call it $\mathcal{EL}^\perp$, includes the concept $\perp$ as well as nominals.

We let N_C be a set of concept names, N_R a set of role names and N_I a set of individual names. A concept in $\mathcal{EL}^\perp$ is defined as follows:

$$C := A \mid \top \mid \perp \mid C \sqcap C \mid \exists r.C \mid \{a\}$$

where $A \in N_C$ and $r \in N_R$. Observe that complement, disjunction and universal restriction are not allowed in $\mathcal{EL}^\perp$.

A knowledge base K is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox containing a finite set of concept inclusions $C_1 \sqsubseteq C_2$ and $\mathcal{A}$ is an ABox containing assertions of the form $C(a)$ and $r(a,b)$, with C, C_1, C_2 concepts, $r \in N_R$ and $a, b \in N_I$.

We will assume that the TBox is in normal form [3]. Let BC_K be the smallest set of concepts containing $\top$, all the concept names occurring in K and all nominals $\{a\}$, for any individual name a occurring in K. An inclusion is in *normal form* if it has one of the following forms: $C_1 \sqsubseteq D$, $C_1 \sqcap C_2 \sqsubseteq D$, $C_1 \sqsubseteq \exists r.C_2$, $\exists r.C_2 \sqsubseteq D$, where $C_1, C_2 \in BC_K$, and $D \in BC_K \cup \{\bot\}$. In [3] it is shown that any TBox can be normalized in linear time, by introducing new concept and role names.

In the following we will denote with $N_{C,K}$, $N_{R,K}$ and $N_{I,K}$ the (finite) sets of concept names, role names and individual names occurring in K.

Definition 1 (Interpretations and models). *An interpretation in $\mathcal{EL}^\bot$ is any structure $(\Delta^I, \cdot^I)$ where: Δ^I is a domain; $\cdot^I$ is an interpretation function that maps each concept name A to set $A^I \subseteq \Delta^I$, each role name r to a binary relation $r^I \subseteq \Delta^I \times \Delta^I$, and each individual name a to an element $a^I \in \Delta^I$. Furthermore: $\top^I = \Delta^I$, $\bot^I = \emptyset$; $\{a\}^I = \{a^I\}$; $(C \sqcap D)^I = C^I \cap D^I$; $(\exists r.C)^I = \{x \in \Delta \mid \exists y \in C^I : (x,y) \in r^I\}$. An interpretation $(\Delta^I, \cdot^I)$ satisfies an inclusion $C \sqsubseteq D$ if $C^I \subseteq D^I$; it satisfies an assertion $C(a)$ if $a^I \in C^I$; it satisfies an assertion $r(a,b)$ if $(a^I, b^I) \in r^I$.*

Given a knowledge base $K = (\mathcal{T}, \mathcal{A})$, an interpretation $(\Delta^I, \cdot^I)$ is a model *of $\mathcal{T}$ if $(\Delta^I, \cdot^I)$ satisfies all inclusions in $\mathcal{T}$; $(\Delta^I, \cdot^I)$ is a* model *of K if $(\Delta^I, \cdot^I)$ satisfies all inclusions in $\mathcal{T}$ and all assertions in $\mathcal{A}$. $\mathcal{A}$ is* consistent with *$\mathcal{T}$ if there is a model of $\mathcal{T}$ satisfying all the assertions in $\mathcal{A}$.*

3. Temporal Action Theories

In this paper we refer to the notion of the temporal action theory in [19], a rule based fragment of which has been studied in [23, 24], which exploits the dynamic extension of LTL introduced by Henriksen and Thiagarajan, called Dynamic Linear Time Temporal Logic (DLTL) [28]. In DLTL the next state modality is indexed by actions and the until operator $\mathcal{U}^\pi$ is indexed by a program π which, as in PDL, can be any regular expression built from atomic actions using sequence (;), nondeterministic choice $(+)$ and finite iteration $(*)$. The derived modalities $\langle\pi\rangle$ and $[\pi]$ can be defined as: $\langle\pi\rangle\alpha \equiv \top\mathcal{U}^\pi\alpha$ and $[\pi]\alpha \equiv \neg\langle\pi\rangle\neg\alpha$. Similarly, $\bigcirc$ (next), $\Diamond$ and $\Box$ operators of LTL can be defined. We let Σ be a finite non-empty set of (atomic) actions and we refer to [19, 23] for the details concerning complex actions.

A *domain description* Π is a set of laws describing the effects of actions and their executability preconditions. Atomic propositions describing the state of the domain are called *fluents*. Actions may have direct effects, described by action laws, and indirect effects, described by causal laws capturing the causal dependencies among fluents.

Let $\mathcal{L}$ be a first order language which includes a finite number of constants and variables, but no function symbol. Let $\mathcal{P}$ be the set of predicate symbols, Var the set of variables and $Cons$ the set of constant symbols. We call *fluents* atomic literals of the form $p(t_1, \ldots, t_n)$, where, for each i, $t_i \in Var \cup Cons$. A *simple fluent literal* (or *s-literal*) l is an atomic literals $p(t_1, \ldots, t_n)$ or its negation $\neg p(t_1, \ldots, t_n)$. We denote by Lit_S the set of all simple fluent literals. Lit_T is the set of *temporal fluent literals*: if $l \in Lit_S$, then $[a]l, \bigcirc l \in Lit_T$, where a is an action name (an atomic proposition, possibly containing variables), and $[a]$ and $\bigcirc$ are the temporal operators introduced in the previous section. Let $Lit = Lit_S \cup Lit_T \cup \{\bot, \top\}$, where $\bot$ represents the inconsistency and $\top$ truth. Given a (simple or temporal) fluent literal l, $not\ l$ represents the default negation of l. A (simple or temporal) fluent literal possibly preceded by a default negation, will be called an *extended fluent literal*.

The laws are formulated as rules of a temporally extended logic programming language. Rules have the form

$$\Box(l_0 \leftarrow l_1, \ldots, l_m, not\ l_{m+1}, \ldots, not\ l_n) \tag{1}$$

where the l_i's are either simple fluent literals or temporal fluent literals, with the following constraints: (i) If l_0 is a simple literal, then the body cannot contain temporal literals; (ii) If $l_0 = [a]l$, then the temporal literals in the body must have the form $[a]l'$; (iii) If $l_0 = \bigcirc l$, then the temporal literals in the body must have the form $\bigcirc l'$. As usual in ASP, the rules with variables will be used as a shorthand for the set of their ground instances.

In the following we use of a notion of *state*: a set of ground fluent literals. A state is said to be *consistent* if it is not the case that both f and $\neg f$ belong to the state, or that $\bot$ belongs to the state. A state is said to be *complete* if, for each fluent f, either f or $\neg f$ belong to the state. The execution of an action in a state may possibly change the values of fluents in the state through its direct and indirect effects, thus giving rise to a new state.

While a law as (1) can be applied in all states, a law

$$l_0 \leftarrow l_1, \ldots, l_m, not\ l_{m+1}, \ldots, not\ l_n \tag{2}$$

which is not prefixed by $\Box$, only applies to the initial state.

A domain description can be defined as a pair $(\Pi, \mathcal{C})$, consisting of a set of laws Π and a set of temporal constraints $\mathcal{C}$. The following action laws describe the deterministic effect of the actions *shoot* and *load* for the Russian Turkey problem, as well as the nondeterministic effect of action *spin*, after which the gun may be loaded or not:

$\Box([shoot]\neg alive \leftarrow loaded)$ $\quad\quad$ $\Box[load]loaded$

$\Box([spin]loaded \leftarrow\ not\ [spin]\neg loaded)$ $\quad$ $\Box([spin]\neg loaded \leftarrow\ not\ [spin]loaded)$

The following precondition laws: $\Box([load]\bot \leftarrow loaded)$ specifies that, if the gun is loaded, *load* is not executable. The program $(\neg in\_sight?; wait)^*; in\_sight?; load; shoot$ describes the behavior of the hunter who waits for a turkey until it appears and, when it is in sight, loads the gun and shoots. Actions $in\_sight?$ and $\neg in\_sight?$ are test actions (we refer to [23]). If the constraint $\langle(\neg in\_sight?; wait)^*; in\_sight?; load; shoot\rangle\top$ is included in $\mathcal{C}$ then all the runs of the domain description which do not start with an execution of the given program will be filtered out. For instance, an extension in which in the initial state the turkey is not in sight and the hunter loads the gun and shoots is not allowed.

As we will see later, this temporal language is also well suited to describe causal dependencies among fluents as *static* and *dynamic causal laws* similar to the ones in the action languages $\mathcal{K}$ [14] and $\mathcal{C}^+$ [25].

The semantics of a domain description has been defined based on *temporal answer sets* [23], which extend the notion of *answer set* [15] to capture the linear structure of temporal models. Let us shortly recall the main notions. In the following, we consider the ground instantiations of the domain description Π, and we denote by Σ the set of all the ground instances of the action names in Π.

3.1. Temporal answer sets

A temporal interpretation is defined as a pair (σ, S), where $\sigma \in \Sigma^\omega$ is a sequence of actions and S is a consistent set of ground literals of the form $[a_1; \ldots; a_k]l$, where $a_1 \ldots a_k$ is a prefix of σ and l is a ground simple fluent literal, meaning that l holds in the state obtained by executing $a_1 \ldots a_k$. S is *consistent* iff it is not the case that both $[a_1; \ldots; a_k]l \in S$ and $[a_1; \ldots; a_k]\neg l \in S$, for some l, or $[a_1; \ldots; a_k]\bot \in S$. A temporal interpretation (σ, S) is said to be *total* if either $[a_1; \ldots; a_k]p \in S$ or $[a_1; \ldots; a_k]\neg p \in S$, for each $a_1 \ldots a_k$ prefix of σ and for each fluent name p.

We define the *satisfiability of a simple, temporal or extended literal t in a partial temporal interpretation* (σ, S) *in the state* $a_1 \ldots a_k$, (written $(\sigma, S), a_1 \ldots a_k \models t$) as:

$(\sigma, S), a_1 \ldots a_k \models \top, \qquad (\sigma, S), a_1 \ldots a_k \not\models \bot$
$(\sigma, S), a_1 \ldots a_k \models l$ *iff* $[a_1; \ldots; a_k]l \in S$, for l s-literal
$(\sigma, S), a_1 \ldots a_k \models [a]l$ *iff* $[a_1; \ldots; a_k; a]l \in S$ or
$a_1 \ldots a_k, a$ is not a prefix of σ
$(\sigma, S), a_1 \ldots a_k \models \bigcirc l$ *iff* $[a_1; \ldots; a_k; b]l \in S$,
where $a_1 \ldots a_k b$ is a prefix of σ
$(\sigma, S), a_1 \ldots a_k \models$ *not* l *iff* $(\sigma, S), a_1 \ldots a_k \not\models l$

The satisfiability of rule bodies in a temporal interpretation is defined as usual. A rule $\Box(H \leftarrow Body)$ is satisfied in a temporal interpretation (σ, S) if, for all action sequences $a_1 \ldots a_k$ (including the empty action sequence ε), $(\sigma, S), a_1 \ldots a_k \models Body$ implies $(\sigma, S), a_1 \ldots a_k \models H$. A rule $H \leftarrow Body$ is satisfied in a partial temporal interpretation (σ, S) if, $(\sigma, S), \varepsilon \models Body$ implies $(\sigma, S), \varepsilon \models H$.

Let Π be a set of rules over an action alphabet Σ, not containing default negation, and let $\sigma \in \Sigma^\omega$.

Definition 2. *A temporal interpretation* (σ, S) *is a* temporal answer set of Π *if* S *is minimal (with respect to set inclusion) among the* S' *such that* (σ, S') *is a partial interpretation satisfying the rules in* Π.

To define answer sets of a program Π containing negation, given a temporal interpretation (σ, S) over $\sigma \in \Sigma^\omega$, the *reduct,* $\Pi^{(\sigma,S)}$*, of* Π *relative to* (σ, S) is defined, by extending Gelfond and Lifschitz' transform [16], roughly speaking, to compute a different reduct of Π for each prefix $a_1, \ldots, a_h$ of σ.

Definition 3. *The* reduct, $\Pi^{(\sigma,S)}_{a_1,\ldots,a_h}$, of Π relative to (σ, S) and to the prefix $a_1, \ldots, a_h$ of σ *, is the set of all the rules* $[a_1; \ldots; a_h](H \leftarrow l_1, \ldots, l_m)$ *such that* $H \leftarrow l_1, \ldots, l_m,$ *not* $l_{m+1}, \ldots,$ *not* l_n *is in* Π *and* $(\sigma, S), a_1, \ldots, a_h \not\models l_i$*, for all* $i = m + 1, \ldots, n$*.*

The reduct $\Pi^{(\sigma,S)}$ of Π relative to (σ, S) *is the union of all reducts* $\Pi^{(\sigma,S)}_{a_1,\ldots,a_h}$ *for all prefixes* $a_1, \ldots, a_h$ *of* σ*.*

In definition above, we say that rule $[a_1; \dots; a_h](H \leftarrow Body)$ is satisfied in a temporal interpretation (σ, S) if $(\sigma, S), a_1 \dots a_k \models Body$ implies $(\sigma, S), a_1 \dots a_k \models H$.

Definition 4 ([23]). *A temporal interpretation (σ, S) is an* answer set of Π *if (σ, S) is an answer set of the reduct $\Pi^{(\sigma,S)}$.*

Observe that a partial interpretation (σ, S) provides, for each prefix $a_1 \dots a_k$, a partial evaluation of fluents in the state corresponding to that prefix. The (partial) state $w^{(\sigma,S)}_{a_1 \dots a_k}$ obtained by the execution of the actions $a_1 \dots a_k$ in the sequence can be defined as: $w^{(\sigma,S)}_{a_1 \dots a_k} = \{l : [a_1; \dots; a_k]l \in S\}$.

Although the answer sets of a domain description Π are partial interpretations, in some cases, e.g., when the initial state is complete and all fluents are inertial, it is possible to guarantee that the temporal answer sets of Π are total. The case of total temporal answer sets is of special interest as a total temporal answer set (σ, S) can be regarded as a temporal model.

4. Combining Temporal Action Theories with $\mathcal{EL}^\perp$ KBs

In this section we define a notion of *extended temporal action theory*, consisting of a temporal action theory plus an $\mathcal{EL}^\perp$ knowledge base. Our approach, following most of the proposals for reasoning about actions in DLs [5, 6, 12, 4, 1] is to regard the TBox as a set of state constraints, i.e. conditions that must be satisfied by any state of the world (in all possible extensions of the action theory), and the ABox as a set of constraints on all possible initial states.

We want to regard DL assertions as fluents that may occur in our action laws as well as in the states of the action theory. Given a (normalized) $\mathcal{EL}^\perp$ knowledge base $K = (\mathcal{T}, \mathcal{A})$, we require that: (a) for each (possibly complex) concept C occurring in K there is a unary predicate $C \in \mathcal{P}$; (b) for each role name $r \in N_{R,K}$ there is a binary predicate $r \in \mathcal{P}$; (c) the set of constants $Cons$ includes all the individual names occurring in the K, i.e. $N_{I,K} \subseteq Cons$.

Observe that if a complex concept such as $\exists r.C$ occurs in K, there exists a predicate name $\exists r.C \in \mathcal{P}$ and, for each $a \in N_{I,KB}$, the fluent literals $(\exists r.C)(a)$ and $\neg(\exists r.C)(a)$ belong to the set Lit (we will still call such literals assertions). Although classical negation is not allowed in $\mathcal{EL}^\perp$, we use *explicit negation* [15] to allow negative literals of the form $\neg C(a)$ in the action language (to allow for deleting an assertion from a state).

A simple literal in Lit is said to be a *simple assertion* if it has the form $B(a)$ or $r(a, b)$ or $\neg B(a)$ or $\neg r(a, b)$, where $B \in BC_K$ is a base concept in K, $r \in N_{R,K}$ and $a, b \in N_{I,K}$. Observe that $\{a\}(c)$ and $\neg\{a\}(c)$ are simple assertions, while $(\exists r.C)(a)$ and $\neg(\exists r.C)(a)$ are non-simple assertions.

In order to deal with existential restrictions, in addition to the individual names $N_{I,KB}$ occurring in the KB we introduce a finite set Aux of auxiliary individual names, as proposed in [29] to encode $\mathcal{EL}^\perp$ inference in Datalog, where Aux contains a new individual name $aux^{A \sqsubseteq \exists r.B}$, for each inclusion $A \sqsubseteq \exists r.B$ occurring in the KB. We further require that $Aux \subseteq Cons$.

Definition 5 (Extended action theory). *An* extended action theory *is a tuple $(K, \Pi, \mathcal{C})$, where: $K = (\mathcal{T}, \mathcal{A})$ is an $\mathcal{EL}^\perp$ knowledge base; Π is a set of laws: action, causal, executability and initial state laws (see below); $\mathcal{C}$ is a set of temporal constraints.*

Action laws describe the immediate effects of actions. They have the form:

$$\Box([a]l_0 \leftarrow t_1, \ldots, t_m, \mathit{not}\ t_{m+1}, \ldots, \mathit{not}\ t_n) \tag{3}$$

where l_0 is a simple fluent literal and the t_i's are either simple fluent literals or temporal fluent literals of the form $[a]l$. Its meaning is that executing action a in a state in which the conditions $t_1, \ldots, t_m$ hold and conditions $t_{m+1}, \ldots, t_n$ do not hold causes the effect l_0 to hold. As an example, $[Assign(c, x)]\,Teaches(x, c) \leftarrow Course(c)$ means that executing the action of assigning a course c to x has the effect that x teaches c. Non-deterministic effects of actions can be defined using default negation in the body of action laws, as for the action *spin* in Section 3.

Causal laws describe indirect effects of actions. They have the form: In Π we allow two kinds of causal laws. *Static causal laws* have the form:

$$\Box(l_0 \leftarrow l_1, \ldots, l_m, \mathit{not}\ l_{m+1}, \ldots, \mathit{not}\ l_n) \tag{4}$$

where the l_i's are simple fluent literals. Their meaning is: if $l_1, \ldots, l_m$ hold in a state and $l_{m+1}, \ldots, l_n$ do not hold in that state, than l_0 is caused to hold in that state. *Dynamic causal laws* have the form:

$$\Box(\bigcirc l_0 \leftarrow t_1, \ldots, t_m, \mathit{not}\ t_{m+1}, \ldots, \mathit{not}\ t_n) \tag{5}$$

where l_0 is a simple fluent literal and the t_i's are either simple fluent literals or temporal fluent literals of the form $\bigcirc l_i$. For instance, $\bigcirc\, Teacher(x) \leftarrow \bigcirc\, Teaches(x, y) \wedge Course(y)$. Observe that, differently from [5, 4], we do not restrict direct and indirect effects of actions to be simple assertions. *State constraints* that apply to the initial state or to all states can be obtained when $\bot$ occurs in the head of initial state laws (6) or static causal laws (4).

Precondition laws describe the executability conditions of actions. They have the form: $\Box([a]\bot \leftarrow l_1, \ldots, l_m, \mathit{not}\ l_{m+1}, \ldots, \mathit{not}\ l_n)$ where $a \in \Sigma$ and the l_i's are simple fluent literals. The meaning is that the execution of an action a is not possible in a state in which $l_1, \ldots, l_m$ hold and $l_{m+1}, \ldots, l_n$ do not hold

Initial state laws are needed to introduce conditions that have to hold in the initial state. They have the form:

$$l_0 \leftarrow l_1, \ldots, l_m, \mathit{not}\ l_{m+1}, \ldots, \mathit{not}\ l_n \tag{6}$$

where the l_i's are simple fluent literals. Observe that initial state laws, unlike static causal laws, only apply to the initial state as they are not prefixed by the $\Box$ modality. As a special case, the initial state can be defined as a set of simple fluent literals. For instance, the initial state $\{alive, \neg in\_sight, \neg frightened\}$ is defined by the initial state laws: $alive$, $\neg in\_sight$, $\neg frightened$.

Following Lifschitz [30] we call *frame fluents* the fluents to which the law of inertia applies. Persistence of frame fluents from a state to the next one can be captured by introducing in Π a set of causal laws, said *persistence laws* for all frame fluents f.

$$\Box(\bigcirc f \leftarrow f,\ \mathit{not} \bigcirc \neg f) \qquad \Box(\bigcirc \neg f \leftarrow \neg f,\ \mathit{not} \bigcirc f)$$

meaning that, if f holds in a state, then f will hold in the next state, unless its negation $\neg f$ is caused to hold (and similarly for $\neg f$). Persistence of a fluent is blocked by the execution of an action which causes the value of the fluent to change, or by a nondeterministic action which may cause it to change. The persistence laws above play the role of *inertia rules* in $\mathcal{C}$ [26], $\mathcal{C}+$ [25] and $\mathcal{K}$ [14].

If f is *non-frame* with respect to an action a, f is not expected to persist and may change its value when an action α is executed, either non-deterministically:

$$\Box(\bigcirc f \leftarrow not \bigcirc \neg f) \qquad \Box(\bigcirc \neg f \leftarrow not \bigcirc f)$$

or by taking some default value (see [23] for some examples).

In the following we assume that persistence laws and non-frame laws can be applied to simple assertions but not to non-simple ones (such as $(\exists r.B)(x)$), whose value in a state (as we will see) is determined from the value of simple assertions. For simple assertions $A(c)$ in a domain description, one has to choose whether the concept A is frame or non-frame (so that either persistence laws or non-frame laws can be introduced). In particular, we assume that all the nominals always correspond to frame fluents: if $\{a\}(b)$ (respectively $\neg\{a\}(b)$) belongs to a state, it will persist to the next state unless it is cancelled by the direct or indirect effects of an action.

ABox assertions may incompletely specify the initial state. As we want to reason about states corresponding to $\mathcal{EL}^{\perp}$ interpretations, we assume that the laws: $f \leftarrow not \neg f$ and $\neg f \leftarrow not f$ for *completing the initial state* are introduced in Π for all simple literals f (including assertions with nominals). As shown in [23], the assumption of complete initial states, together with suitable conditions on the laws in Π, gives rise to semantic interpretations (extensions) of the domain description in which all states are complete. In particular, to guarantee that each reachable state in each extension of a domain description is complete for simple fluents, we assume that, either the fluent is frame, and persistence laws are introduced for it, or it is non-frame, and non-frame laws are introduced. Other literals, such as existential assertions, are not subject to this requirement but, as we will see below, the value of existential assertions in a state will be determined from the value of simple assertions. Under the condition above, starting from an initial state which is complete for simple fluents, all the reachable states are also complete for simple fluents. We call *well-defined* the domain descriptions satisfying this condition (and sometimes we will simply say that Π is well-defined).

The third component $\mathcal{C}$ of a domain description $(K, \Pi, \mathcal{C})$ is the set of *temporal constraints* in DLTL, which allow general temporal conditions to be imposed on the executions of the domain description. Their effect is that of restricting the space of the possible executions (or *extensions*). For a domain description $D = (\Pi, \mathcal{C})$, as introduced in [23], extensions are defined as follows.

Definition 6. *An* extension of a well-defined domain domain description $D = (\Pi, \mathcal{C})$ over Σ *is a (total) answer set* (σ, S) *of* Π *which satisfies the constraints in* $\mathcal{C}$.

In the next section we extend the notion of extension to a domain description $(K, \Pi, \mathcal{C})$ with K an ontology.

5. Ontology axioms as state constraints

Given an action theory $(K, \Pi, \mathcal{C})$, where $K = (\mathcal{T}, \mathcal{A})$, we define an extension of $(K, \Pi, \mathcal{C})$ as an extension (σ, S) of the action theory $(\Pi, \mathcal{C})$ satisfying all axioms of the ontology K. Informally, each state w in the extension is required to correspond to an $\mathcal{EL}^{\perp}$ interpretation and to satisfy all inclusion axioms in TBox $\mathcal{T}$. Additionally, the initial state must satisfy all assertions in the ABox $\mathcal{A}$.

To define the extensions of an action theory $(K, \Pi, \mathcal{C})$, we restrict to well-defined domain descriptions $(K, \Pi, \mathcal{C})$, so that all states in an extension are complete for simple fluents (and for simple assertions). We prove that such states represent $\mathcal{EL}^{\perp}$ interpretations on the language of K, provided an additional set of laws $\Pi_{\mathcal{L}(K)}$ is included in the action theory. Next, we add to Π another set $\Pi_{\mathcal{T}}$ of constraints, to guarantee that each state satisfies the inclusion axioms in $\mathcal{T}$. Finally, we add to Π the set of laws $\Pi_{\mathcal{A}}$, to guarantee that all assertions in $\mathcal{A}$ are satisfied in the initial state.

Overall, this provides a transformation of the action theory $(K, \Pi, \mathcal{C})$ into a new action theory $(\Pi \cup \Pi_K, \mathcal{C})$, by eliminating the ontology while introducing a set of static causal laws and constraints $\Pi_K = \Pi_{\mathcal{L}(K)} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{A}}$, intended to exclude those extensions which do not satisfy the axioms in K.

The set of domain constraints and causal laws in $\Pi_{\mathcal{L}(K)}$ is intended to guarantee that any state w of an extension respects the semantics of DL concepts occurring in K. The definition of $\Pi_{\mathcal{L}(K)}$ is based on a fragment of the materialization calculus for $\mathcal{EL}^{\perp}$, which provides a Datalog encoding of an $\mathcal{EL}^{\perp}$ ontology. Here, the idea is that of regarding an assertion $C(a)$ in a state w as the evidence that $a^I \in C^I$ in the corresponding interpretation. Let $\Pi_{\mathcal{L}(K)}$ be the following set of causal laws:

(1) $\Box(\perp \leftarrow \perp(x))$ (2) $\Box(\top(x) \leftarrow)$ (3) $\Box(\{a\}(a) \leftarrow)$

(4) $\Box(exists\_r\_B(x) \leftarrow r(x,y) \wedge B(y))$

(5) $\Box((\exists r.B)(x) \leftarrow exists\_r\_B(x))$

(6) $\Box(\neg(\exists r.B)(x) \leftarrow not\ exists\_r\_B(x))$

(7) $\Box(\perp \leftarrow \{a\}(x) \wedge B(x) \wedge not\ B(a))$, for $x \neq a$
(8) $\Box(\perp \leftarrow \{a\}(x) \wedge B(a) \wedge not\ B(x))$, for $x \neq a$
(9) $\Box(\perp \leftarrow \{a\}(x) \wedge r(z,x) \wedge not\ r(z,a))$, for $x \neq a$

for all $x, y \in \Delta$, $a \in N_{I,K}$, $B \in BC_K$ (the base concepts occurring in K) and $r \in N_{R,K}$ (the roles occurring in K). Observe that the first constraint has the effect that a state S, in which the concept $\perp$ has an instance, is made inconsistent. Law (4) makes $exists\_r\_B(x)$ hold in any state in which there is a domain element y such that $r(x,y)$ and $B(y)$ hold (where $exists\_r\_B$ is an additional auxiliary predicate for $B \in BC_K$ and $r \in N_{R,K}$). Laws (5) and (6) guarantee that, for all $x \in \Delta$, either $(\exists r.B)(x)$ or $\neg(\exists r.B)(x)$ is contained in the state. State constraints (7-9) are needed for the treatment of nominals and are related to materialization calculus rules (27-29) [29].

Let (σ, S) be an extension of the action theory $(\Pi \cup \Pi_{\mathcal{L}(K)}, \mathcal{C})$. It can be proven that any state w of (σ, S) represents an $\mathcal{EL}^{\perp}$ interpretation. Given a state w, let w^+ be the set of $\mathcal{EL}^{\perp}$ assertions $C(a)$ $(r(a,b))$, such that $C(a) \in w$ (resp., $r(a,b) \in w$). Let w^- be the set of $\mathcal{EL}^{\perp}$ assertions $C(a)$ $(r(a,b))$, such that $\neg C(a) \in w$ (resp., $\neg r(a,b) \in w$).

Proposition 1. *Let (σ, S) be an extension of the action theory $(\Pi \cup \Pi_{\mathcal{L}(K)}, \mathcal{C})$ and let w be a state of (σ, S). Then there is an interpretation $(\Delta^I, \cdot^I)$ that satisfies all the assertions in w^+ and falsifies all assertions in w^- (and we say that $(\Delta^I, \cdot^I)$* agrees with state w*).*

As mentioned, we are interested in the states satisfying the TBox $\mathcal{T}$ of K. Provided Π is well-defined, for each extension (σ, S) of the action theory $(\Pi \cup \Pi_{\mathcal{L}(K)}, \mathcal{C})$, any state w is consistent and complete for all simple literals (and hence, by (5) and (6), for all assertions). We say that w *satisfies the TBox* $\mathcal{T}$ if for all interpretations $(\Delta^I, \cdot^I)$ such that $(\Delta^I, \cdot^I)$ agrees with state w, $(\Delta^I, \cdot^I)$ is a model of $\mathcal{T}$.

The requirement that each w should satisfy the TBox $\mathcal{T}$ can be incorporated in the action theory through a set of constraints, that we call $\Pi_{\mathcal{T}}$, by exploiting the fact that the TBox $\mathcal{T}$ is in normal form. $\Pi_{\mathcal{T}}$ contains the following state constraints:

$$\Box(\bot \leftarrow A(x) \wedge not\ D(x)),\ \text{for each } A \sqsubseteq D \text{ in } \mathcal{T};$$
$$\Box(\bot \leftarrow A(x) \wedge B(x) \wedge not\ D(x)),\ \text{for each } A \sqcap B \sqsubseteq D \text{ in } \mathcal{T};$$
$$\Box(\bot \leftarrow A(x) \wedge not\ (\exists r.B)(x)),\ \text{for each } A \sqsubseteq \exists r.B \text{ in } \mathcal{T};$$
$$\Box(\bot \leftarrow (\exists r.B)(x) \wedge not\ D(x)),\ \text{for each } \exists r.B \sqsubseteq D \text{ in } \mathcal{T};$$

where $A, B \in BC_K$, $D \in BC_K \cup \{\bot\}$ and $x \in N_{I,K} \cup Aux$. For $D = \bot$, the condition $not\ D(x)$ is omitted. The following proposition can be proved for a well-defined Π.

Proposition 2. *Let (σ, S) be an extension of the action theory $(\Pi \cup \Pi_{\mathcal{L}(K)}, \mathcal{C})$ and let w be a state of (σ, S); w satisfies $\mathcal{T}$ iff w satisfies all constraints in $\Pi_{\mathcal{T}}$.*

We can then add the constraints in $\Pi_{\mathcal{T}}$ to an action theory $(\Pi \cup \Pi_{\mathcal{L}(K)}, \mathcal{C})$ to single out the extensions (σ, S) whose states all satisfy the TBox $\mathcal{T}$. In a similar way, we can restrict to answer sets (σ, S) whose initial state $w_{\varepsilon}^{(\sigma,S)}$ satisfies all assertions in $\mathcal{A}$, by defining a set of initial state constraints as:

$$\Pi_{\mathcal{A}} = \{\bot \leftarrow not\ A(c)) \mid A(c) \in \mathcal{A}\} \cup \{\bot \leftarrow not\ r(c,d)) \mid r(c,d) \in \mathcal{A}\}$$

We define the *extensions of the extended action theory* $(K, \Pi, \mathcal{C})$ as the extensions of the action theory $(\Pi \cup \Pi_K, \mathcal{C})$, where $\Pi_K = \Pi_{\mathcal{L}(K)} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{A}}$.

Given this notion of extension of an action theory $(K, \Pi, \mathcal{C})$, we can verify, in the temporal action logic, whether a sequence of actions is executable from the initial state (executability problem) or whether any execution of an action sequence makes some property (e.g., assertion) hold in all reachable states (temporal projection problem).

Let us consider the example from the introduction.

Example 1. *Let $K = (\mathcal{T}, \mathcal{A})$ be a knowledge base such that $\mathcal{T}$= $\{\exists Teaches.Course \sqsubseteq Teacher\}$ and $\mathcal{A}$= $\{$ $Person(john)$, $Course(cs1)\}$. We assume that all simple assertions, i.e., $Person(x)$, $Teacher(x)$, $Course(x)$, $Teaches(x,y)$, are frame for all $x, y \in N_{I,K} \cup Aux$.*

Let us consider a state w_0 where John does not teach any course and is not a teacher. If an action $Assign(cs1, john)$ were executed in w_0, given a Π containing the action law $[Assign(cs1, john)]\,Teaches(john, cs1)$, the resulting state would contain $\exists Teaches.Course)(john)$, but would not contain $Teacher(john)$, thus violating the state constraint

$$\Box(\bot \leftarrow (\exists Teaches.Course)(x) \wedge not\ Teacher(x)),$$

in $\Pi_{\mathcal{T}}$. In this case, there is no extension of the action theory in which action $Assign(cs1, john)$ can be executed in the initial state.

As observed in [4], when this happens, the action specification can be regarded as being underspecified, as it is not able to capture the dependencies among fluents which occur in the TBox. To guarantee that TBox is satisfied in the new state, causal laws are needed which allow the state to be repaired. *In the specific case, adding causal law $\Box(Teacher(x) \leftarrow Teaches(x, y) \wedge Course(y))$ to Π would suffice to cause $Teacher(x)$ in the resulting state, as an indirect effect of action $Assign(cs1, john)$.*

6. Causal laws for repairing inconsistencies: sufficient conditions

Can we identify which and how many conditions are needed to guarantee that an action theory is able to repair the state, after executing an action, so to satisfy all inclusions of a (normalized) $\mathcal{EL}^{\perp}$ TBox, when possible? Let us continue Example 1.

Example 2. *Consider the case when action $retire(john)$ is executed in a state w_1 where $Person(john)$, $Course(cs1)$, $Teaches(john, cs1)$ and $Teacher(john)$ hold. Suppose that the action law: $[retire(john)]\neg Teacher(john)$ is in Π. Then, $\neg Teacher(john)$ will belong to the new state (let us call it w_2), but w_2 will still contain the literals: $Course(cs1)$, $Teaches(john, cs1)$, which persist from the previous state (as $Course$ and $Teaches$ are frame fluents). Hence, w_2 would violate the TBox $\mathcal{T}$. To avoid this, Π should contain some causal law to repair the inconsistency, for instance, $\Box(\neg Teaches(x, y) \leftarrow \neg Teacher(x) \wedge Course(y))$. By this causal law, when John retires he stops teaching all the courses he was teaching before. In particular, he stops teaching $cs1$. On the contrary, $\Box(\neg Course(y) \leftarrow \neg Teacher(x) \wedge Teaches(x, y))$ would be unintended.*

As we can see, the causal laws needed to restore consistency when an action is executed can in essence be obtained from the inclusions in the TBox and from their contrapositives, even though not all contrapositives are always wanted.

In general, while defining a domain description, one has to choose which causal relationships are intended and which are not. The choice depends on the domain and should not be done automatically, but some sufficient conditions to restore the consistency of the resulting state (if any) with a TBox (in normal form) can be defined.

The case of a (normalized) inclusion $A \sqsubseteq B$, with $A, B \in N_C$, is relatively simple; the execution of an action α with effect $A(c)$ (but not $B(c)$), in a state in which none of $A(c)$ and $B(c)$ holds, would lead to a state which violates the constraints in the KB. Similarly for an action β with effect $\neg B(c)$. Deleting $B(c)$ should cause $A(c)$ to be deleted as well, if we want the inclusion $A \sqsubseteq B$ to be satisfied. Hence, to guarantee that the TBox is satisfied in the new state, for each inclusion $A \sqsubseteq B$, two causal laws are needed: $\Box(B(x) \leftarrow A(x))$ and $\Box(\neg A(x) \leftarrow \neg B(x))$.

For an axiom $A \sqcap B \sqsubseteq \perp$, consider the concrete case $pending \sqcap approved \sqsubseteq \perp$, representing mutually exclusive states of a claim in a process of dealing with an insurance claim, we expect the following causal laws to be included:

$\Box(\neg pending(x) \leftarrow approved(x)) \qquad \Box(\neg approved(x) \leftarrow pending(x))$
even though the second one is only useful if a claim can become pending again after having become (temporarily) approved. For the general case, we have the following.

Let us define a set $\Pi_{C(\mathcal{T})}$ of causal laws associated with $\mathcal{T}$ as follows:

- For $A \sqsubseteq B$ in $\mathcal{T}$: $\Box(B(x) \leftarrow A(x))$ and $\Box(\neg A(x) \leftarrow \neg B(x))$;
- For $A \sqcap B \sqsubseteq D$: $\Box(D(x) \leftarrow A(x) \wedge B(x))$ and at least one among $\Box(\neg A(x) \leftarrow \neg D(x) \wedge B(x))$ and $\Box(\neg B(x) \leftarrow \neg D(x) \wedge A(x))$;
- For $A \sqsubseteq \exists r.B$: $\Box(r(x, aux^{A \sqsubseteq \exists r.B}) \leftarrow A(x))$, $\Box(B(aux^{A \sqsubseteq \exists r.B}) \leftarrow A(x))$ and $\Box(\neg A(x) \leftarrow \neg(\exists r.B)(x))$;
- For $\exists r.B \sqsubseteq A$ in $\mathcal{T}$: $\Box(A(x) \leftarrow (\exists r.B)(x)$, $\Box(\neg(\exists r.B)(x) \leftarrow \neg A(x))$ and at least one of: $\Box(\neg r(x, y) \leftarrow \neg A(x) \wedge B(x))$ and $\Box(\neg B(x) \leftarrow \neg A(x) \wedge r(x, y))$.

Proposition 3. *Given a well-defined action theory* $(\Pi, \mathcal{C})$ *and a TBox* $\mathcal{T}$*, any state* w *of an extension* (σ, S) *of* $(\Pi \cup \Pi_{\mathcal{L}(K)} \cup \Pi_{C(\mathcal{T})} \cup \Pi_{\mathcal{A}}, \mathcal{C})$ *satisfies* $\mathcal{T}$.

Observe that, for the case for $A \sqsubseteq \exists r.B$, from $r(x, aux^{A \sqsubseteq \exists r.B})$ and $B(aux^{A \sqsubseteq \exists r.B})$ $(\exists r.B)(x)$ is caused by laws (4-5) in $\Pi_{\mathcal{L}(K)}$. While the causal laws in $\Pi_{C(\mathcal{T})}$ are sufficient to guarantee the consistency of a resulting state with TBox $\mathcal{T}$, one cannot exclude that some action effects are inconsistent with TBox and cannot be repaired (e.g., an action with direct effects $A(c)$ and $\neg B(c)$, conflicting with an axiom $A \sqsubseteq B$ in $\mathcal{T}$). In such a case, the action would not be executable.

Notice that the encoding above of $\mathcal{EL}^{\perp}$ TBox into a set of temporal laws only requires a polynomial number of laws to be added to the action theory (in the size of K). Based on this mapping, the proof methods for our temporal action logic, which are based on the ASP encodings of bounded model checking [22, 23, 24], can be exploited for reasoning about actions in an action theory extended with an $\mathcal{EL}^{\perp}$ knowledge base.

7. Conclusions and Related Work

In this paper we have proposed an approach for reasoning about actions by combining a temporal action logic in [23], whose semantics is based on a notion of temporal answer sets, and an $\mathcal{EL}^{\perp}$ ontology. It is shown that, for $\mathcal{EL}^{\perp}$ knowledge bases in normal form, the consistency of the action theory extensions with the ontology can be verified by adding to the action theory a set of causal laws and state constraints, by exploiting a fragment of the materialization calculus by Krötzsch [29]. Starting from the idea by Baader et al. [4] that causal rules can be used to ensure the consistency of states with the TBox, we have defined sufficient conditions on the action theory to repair the states resulting from action execution and guarantee consistency with TBox. For each inclusion axiom, a suitable set of causal laws has to be added. Our approach provides a polynomial encoding of an extended action theory, including an $\mathcal{EL}^{\perp}$ knowledge base in normal form, into the language of the (DLTL based) temporal action logic studied in [23]. The proof methods for this temporal action logic, based on ASP encodings of bounded model checking [22, 23, 24], can then be exploited for reasoning about actions in an extended action theory. It would also be interesting, for action domains with finite executions, to investigate whether the action theories in [23] can be encoded in in *telingo* [10], and whether the optimized implementation of *telingo* can be exploited for reasoning about action in our extended action theories.

Many of the proposals in the literature for combining DLs with action theories focus on expressive DLs. In their seminal work [5], Baader et al. study the integration of action formalisms with expressive DLs, from $\mathcal{ALC}$ to $\mathcal{ALCOIQ}$, under Winslett's *possible models approach* (PMA) [35], based on the assumption that TBox is acyclic and on the distinction between defined and primitive concepts (i.e., concept names that are not defined in the TBox), where only primitive concepts are allowed in action effects. They determine the complexity of the executability and projection problems and show that they get decidable fragments of the situation calculus. Our semantics departs from PMA as causal laws are considered. As [32] and [4] we do not require the restriction to acyclic TBoxes and primitive concepts in postconditions.

The requirement of acyclic TBoxes is lifted in the work by Liu et al. [32], where an approach to the ramification problem is proposed which does not use causal relationships, but exploits *occlusion* to provide a specification of the predicates that can change through the execution of actions. The idea is to leave to the designer of an action domain the control of the ramifications.

Similar considerations are at the basis of the approach by Baader et al. [4] that, instead, exploits causal relationships for modeling ramifications in an action language for $\mathcal{ALCO}$, and defines its semantics in the style of McCain and Turner fixpoint semantics [33] (the action theory does not deal with non-deterministic effects of actions). It is shown that temporal projection is decidable and ExpTime-complete. In this paper, following [4], we exploit causal laws for modeling ramifications in the context of a temporal action language for $\mathcal{EL}^{\bot}$. It allows for non-deterministic effects of actions and for the distinction between frame and non-frame fluents [30] (which is strongly related to occlusion used in [32]) based on the temporal logic. We have also provided sufficient conditions for an action specification to be consistent with a normalized $\mathcal{EL}^{\bot}$ KB.

Ahmetai et al. [1] study the evolution of Graph Structured Data as a result of updates expressed in an action language. They provide decidability and complexity results for expressive DLs such as $\mathcal{ALCHOIQ}br$ (under finite satisfiability) as well as for variants of DL-*lite*. Complex actions including action sequence and conditional actions are considered. Complex actions are considered as well in [12], where an action formalism is introduced for a family DLs, from $\mathcal{ALCO}$ to $\mathcal{ALCHOIQ}$, exploiting PDL program constructors to define complex actions. As in [5], the TBox is assumed to be acyclic.

In [8] Description Logic and Action Bases are introduced, where an initial Abox evolves over time due to actions which have conditional effects. In [11] the approach is extended to allow for different notions of *repairing* of the resulting state, such as a maximal subset consistent with the Tbox, or the intersection of all such subsets. In this paper, we rely on causal laws for repairing states; selecting the appropriate causal laws means acquiring more knowledge, and allows for a finer control on the resulting state.

Our semantics for actions, as many of the proposals in the literature, requires that a state provides a complete description of the world and is intended to represent an interpretation of the $\mathcal{EL}^{\bot}$ knowledge base. An alternative approach has been adopted in [8], where a state can provide an incomplete specification of the world. In our approach, an incomplete state could be represented as an epistemic state, which distinguish between what is known to be true (or to be false) and what is unknown. An epistemic extension of our action logic, based on temporal answer sets, has been developed in [24], and it can potentially be exploited for reasoning about actions with incomplete states also in presence of ontological knowledge. We leave the study of this case for future work.

References

[1] Ahmetaj, S., Calvanese, D., Ortiz, M., Simkus, M.: Managing change in graph-structured data using description logics. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 966–973 (2014)

[2] Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Kaelbling, L., Saffiotti, A. (eds.) Proc. IJCAI 2005. pp. 364–369. Edinburgh, Scotland, UK (August 2005)

[3] Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: LTCS-Report LTCS-05-01. Inst. for Theoretical Computer Science, TU Dresden (2005)

[4] Baader, F., Lippmann, M., Liu, H.: Using causal relationships to deal with the ramification problem in action formalisms based on description logics. In: LPAR-17. pp. 82–96 (2010)

[5] Baader, F., Lutz, C., Milicic, M., Sattler, U., Wolter, F.: Integrating description logics and action formalisms: First results. In: Proc. AAAI 2005. pp. 572–577 (2005)

[6] Baader, F., Liu, H., ul Mehdi, A.: Verifying properties of infinite sequences of description logic actions. In: ECAI. pp. 53–58 (2010)

[7] Babb, J., Lee, J.: Cplus2asp: Computing action language $\mathcal{C}$+ in Answer Set Programming. In: Proc. Logic Programming and Nonmonotonic Reasoning, LPNMR 2013. pp. 122–134 (2013)

[8] Bagheri Hariri, B., Calvanese, D., Montali, M., De Giacomo, G., De Masellis, R., Felli, P.: Description logic knowledge and action bases. J. Artif. Intell. Res. 46, 651–686 (2013)

[9] Baral, C., Gelfond, M.: Reasoning agents in dynamic domains. In: Logic-Based Artificial Intelligence, pp. 257–279 (2000)

[10] Cabalar, P., Kaminski, R., Morkisch, P., Schaub, T.: telingo = ASP + time. In: Logic Programming and Nonmonotonic Reasoning - 15th International Conference, LPNMR 2019, Philadelphia, PA, USA, June 3-7, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11481, pp. 256–269. Springer (2019)

[11] Calvanese, D., Kharlamov, E., Montali, M., Santoso, A., Zheleznyakov, D.: Verification of inconsistency-aware knowledge and action bases. In: Proc. IJCAI 2013 (2013)

[12] Chang, L., Shi, Z., Gu, T., Zhao, L.: A family of dynamic description logics for representing and reasoning about actions. J. Autom. Reasoning 49(1), 1–52 (2012)

[13] Denecker, M., Theseider Dupré, D., Van Belleghem, K.: An inductive definitions approach to ramifications. Electronic Transactions on Artificial Intelligence 2, 25–97 (1998)

[14] Eiter, T., Faber, W., Leone, N., Pfeifer, G., Polleres, A.: A logic programming approach to knowledge-state planning: Semantics and complexity. ACM Transactions on Computational Logic 5(2), 206–263 (2004)

[15] Gelfond, M.: Handbook of Knowledge Representation, chapter 7, Answer Sets. Elsevier (2007)

[16] Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Logic Programming, Proc. of the 5th Int. Conf. and Symposium. pp. 1070–1080 (1988)

[17] Gelfond, M., Lifschitz, V.: Action languages. Electron. Trans. Artif. Intell. 2, 193–210 (1998)

[18] Giordano, L., Martelli, A., Schwind, C.: Ramification and causality in a modal action logic. J. Log. Comput. 10(5), 625–662 (2000)

[19] Giordano, L., Martelli, A., Schwind, C.: Reasoning about actions in dynamic linear time temporal logic. The Logic Journal of the IGPL 9(2), 289–303 (2001)

[20] Giordano, L., Martelli, A., Spiotta, M., Theseider Dupré, D.: ASP for reasoning about actions with an $EL^{\perp}$ knowledge base. In: Fiorentini, C., Momigliano, A. (eds.) Proceedings of the 31st Italian Conference on Computational Logic, Milano, Italy, June 20-22, 2016. CEUR Workshop Proceedings, vol. 1645, pp. 214–229. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1645/paper_15.pdf

[21] Giordano, L., Martelli, A., Theseider Dupré, D.: Reasoning about actions with temporal answer sets. In: Faber, W., Leone, N. (eds.) Proceedings of the 25th Italian Conference on Computational Logic, Rende, Italy, July 7-9, 2010. CEUR Workshop Proceedings, vol. 598. CEUR-WS.org (2010)

[22] Giordano, L., Martelli, A., Theseider Dupré, D.: Achieving completeness in bounded model checking of action theories in ASP. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012. AAAI Press (2012), http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4532

[23] Giordano, L., Martelli, A., Theseider Dupré, D.: Reasoning about actions with temporal answer sets. Theory and Practice of Logic Programming 13, 201–225 (2013)

[24] Giordano, L., Martelli, A., Theseider Dupré, D.: Achieving completeness in the verification of action theories by bounded model checking in ASP. J. Log. Comput. 25(6), 1307–1330 (2015), https://doi.org/10.1093/logcom/ext067

[25] Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., , Turner, H.: Nonmonotonic causal theories. Artificial Intelligence 153(1-2), 49–104 (2004)

[26] Giunchiglia, E., Lifschitz, V.: An action language based on causal explanation: Preliminary report. In: Proc. AAAI/IAAI 1998. pp. 623–630 (1998)

[27] Heljanko, K., Niemelä, I.: Bounded LTL model checking with stable models. Theory and Practice of Logic Programming 3(4-5), 519–550 (2003)

[28] Henriksen, J., Thiagarajan, P.: Dynamic linear time temporal logic. Annals of Pure and Applied logic 96(1-3), 187–207 (1999)

[29] Krötzsch, M.: Efficient inferencing for OWL EL. In: Proc. JELIA 2010. pp. 234–246 (2010)

[30] Lifschitz, V.: Frames in the space of situations. Artificial Intelligence 46, 365–376 (1990)

[31] Lin, F.: Embracing causality in specifying the indirect effects of actions. In: IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes. pp. 1985–1993 (1995)

[32] Liu, H., Lutz, C., Milicic, M., Wolter, F.: Reasoning about actions using description logics with general tboxes. In: Proc. JELIA 2006, Liverpool, UK. pp. 266–279 (2006)

[33] McCain, N., Turner, H.: A causal theory of ramifications and qualifications. In: Proc. IJCAI 95. pp. 1978–1984 (1995)

[34] Thielscher, M.: Ramification and causality. Artif. Intell. 89(1-2), 317–364 (1997)

[35] Winslett, M.: Reasoning about action using a possible models approach. In: Proc. AAAI, St. Paul, MN, August 21-26, 1988. pp. 89–93 (1988)