Actions representation and reasoning in ontology languages

Integrating Description Logics and Action Formalities: First Results.

Franz Baader, Carsten Lutz, Maja Milicic, Ulrike Sattler, Frank Wolter

Group 10 - Asan Agibetov (asan.agibetov@ge.imati.cnr.it), Imon Banerjee (imon@ge.imati.cnr.it), Fahad Khan (fahad.khan@ilc.cnr.it)

January 30, 2014
Overview of the problem

- Descriptive action formalism based on Situation Calculus (SitCalc) to support reasoning.
- Motive of this action formalism is to analysis that how the choice of DL influence the reasoning task.
  - **Executability problem**: determine whether a given sequence of ground actions is possible to be executed starting from the initial situation.
  - **Projection problem**: determine whether a given goal G is satisfiable after executing a sequence of ground actions starting from the initial situation.
Situation Calculus (SitCalc) [1]

- Situation calculus is designed for representing and reasoning about dynamic domains.

- **Basic elements:**
  
  **Action** that can be perform in the world
  
  $\text{Move}(x,y)$: robot is moving from position $x$ to $y$

  **Fluents** that describe the world
  
  $\text{is\_carrying(ball,S0)} = \text{false}$
  
  $\text{is\_carrying(ball, do(pick\_up(ball, S0)))} = \text{true}$

  **Situation** represent history of action occurrences
  
  $\text{do(move(2,3), S0)}$: denotes a new situation after performing action $\text{move(2,3)}$ in initial situation $S0$

Insufficiency in SitCalc

Reasoning for action in general is undecidable under “Open world assumption (OWA)”

Frame problem: How it can be decidable that after picking up an object, the robot stays in the same location?

It requires frame axioms like this,

\[
\text{Poss}(\text{pickup}(o), s) \cup \text{location}(s) = (x, y) \rightarrow \\
\text{location}(\text{do}(\text{pickup}(o), s)) = (x, y)
\]

problem: too many of such axioms, difficult to specify all
Well established solution: frame problem[2]

- Successor state axioms: Specify all the ways the value of a particular fluent can be changed

\[ \text{Poss}(a, s) \lor \gamma_+F(x, a, s) \rightarrow F(x, \text{do}(a, s)) \]
\[ \text{Poss}(a, s) \lor \gamma_-F(x, a, s) \rightarrow \neg F(x, \text{do}(a, s)) \]

\( \gamma_+F \) describes the conditions under which action \( a \) in situation \( s \) makes the fluent \( F \) become true in the successor situation \( \text{do}(a,s) \).

\( \gamma_-F \) describes the conditions under which action \( a \) in situation \( s \) makes the fluent \( F \) become false in the successor situation.

- For each action \( A \), a single action precondition axioms of the form: \( \prod_{A(s)} \supset \text{Poss}(A, s) \)

- Unique names axioms for the actions and for states

Proposed work

Design an initial framework for integrating DLs and action formalisms into a decidable hybrid based on *DL ALCQIO* [3] and a number of its sub languages.
i. Acyclic Tbox
A terminology (or TBox) is a set of definitions and specializations. Woman $\equiv$ Person $\sqcap$ Female
A terminology $T$ is Acyclic if it does not contain a concept which uses itself.

\[
\begin{align*}
\text{Father} & \equiv \text{Male} \sqcap \text{hasChild} \\
\text{hasChild} & \equiv \text{Father} \sqcup \text{Mother}
\end{align*}
\]

Not an Acyclic TBox
ii. ABox assertions
In an ABox one introduces individuals, by giving them names, and one asserts properties about them.
Assertion with concept C in the form: C(a), C(b) …
example: Woman(Shelly), Male(John), …
Assertion with role name s in the form: s(a,b), s(b,c), or ¬s(a,b)
example: Father (John, haschild)
Modeling framework (Definition 1)

Let $T = \text{Acyclic Tbox}$

Atomic action $\alpha = (\text{pre}, \text{occ}, \text{post})$

- a finite set $\text{pre}$ of ABox assertions, the \textit{pre-conditions};
- a finite set $\text{occ}$ of occlusion of the form $A(a)$ or $s(a, b)$
- a finite set $\text{post}$ of \textit{conditional post-conditions of the form} $\vartheta/\psi$, where $\vartheta$ is an ABox assertion and $\psi$ is a primitive literal for $T$
Example of action definition

$\alpha_1$ Opening a bank account in Italy
Ok, can you deposit 1000 euro? Do you have proof of address

$pre_1$
$\{ Eligible\_bank(a), \exists holds.\ Proof\_address(a) \}$

$post_1$
$\{ T(a)/holds(a, b),$
$\exists holds.letter(a)/B\_acc\_credit(b),$
$\neg \exists holds.letter(a)/B\_acc\_no\_credit(b) \}$
Apply for child benefit in Italy

Ok, do you have child? Do you have a bank account?

pre2
\{parents\_of (a, c), \exists hold. B\_acc (a)\}
post2
\{ T(a)/receives\_c\_benefit\_for (a, c)\}

TBox
Eligible\_bank \equiv \exists can\_deposit.1000,
Proof\_address \equiv
Passport \cup Carta\_identita,
B\_acc \equiv
B\_acc\_credit \cup B\_acc\_no\_credit
Semantics of actions

Where each primitive concept name: A, role name s:s(a,b),
Interpretation : I

\[ A^+ = \{ b^I \mid \varphi/A(b) \in \text{post and } I \models \varphi \} \]

\[ A^- = \{ b^I \mid \varphi/ - A(b) \in \text{post and } I \models \varphi \} \]

\[ I_A = (\Delta^I \setminus \{ b^I \mid A(b) \in \text{occ } \}) \cup (A^+ \cup A^-) \]

\[ s^+ = \{ (a^I, b^I) \mid \varphi/s(a,b) \in \text{post and } I \models \varphi \} \]

\[ s^- = \{ (a^I, b^I) \mid \varphi/ - s(a,b) \in \text{post and } I \models \varphi \} \]

\[ I_s = ((\Delta^I \times \Delta^I) \setminus ((a^I), b^I) \mid s(a,b) \in \text{occ } ) \cup (s^+ \cup s^-) \]
Modeling framework (Definition 2)

Action $\alpha$ may transform $I$ to $I'$ iff, for each primitive concept $A$ and role name $s$,

$$A^+ \cap A^- = s^+ \cap s^- = \emptyset, \quad A^{I'} \cap I_A = (\left((A^I \cup A^+) \setminus A^\neg\right) \cap I_A$$

$$s^{I'} \cap I_s = (\left((s^I \cup s^+) \setminus s^\neg\right) \cap I_s$$

The composite action $\alpha_1 \ldots \alpha_k$ may transform $I$ to $I'$ iff there exist models $I_0, \ldots, I_k$ of $I$ with $T = I_0$, $I' = I_k$ and $I_{i-1} \implies T_{\alpha_i} I_i$
Note for definition 1 and 2

Due to the acyclic TBox, action with empty occlusions there can not exist more than one $I'$ such that

$$ I \rightarrow_{\alpha} I' $$

Thus, actions are deterministic.

if $\vartheta_1/\psi$, $\vartheta_2/\neg\psi \in \text{post}$

such that both $\vartheta_1$ and $\vartheta_2$ are satisfied in $I$, then there is no successor model $I'$. So action is inconsistent with $I$. 
Reasoning Problems

Describe two main reasoning problems for actions.
Given an acyclic TBox $\mathcal{T}$, a composite action $\alpha = \alpha_1, \ldots, \alpha_k$ and an ABox $\mathcal{A}$ we want to know

- Executability: are all the preconditions of $\alpha$ satisfied in worlds considered possible?
- Projection: does a given assertion hold after applying $\alpha$?
Recall:
An ABox $\mathcal{A}$ is *consistent* with respect to a TBox $\mathcal{T}$ if there exists an interpretation $\mathcal{I}$ that is a model of both $\mathcal{A}$ and $\mathcal{T}$. 
Definition (Executability)
Given an acyclic TBox $\mathcal{T}$, a composite action $\alpha = \alpha_1, \ldots, \alpha_k$ where $\alpha_i = (\text{pre}_i, \text{occ}_i, \text{post}_i)$ and an ABox $\mathcal{A}$ we say that $\alpha$ is executable in $\mathcal{A}$ with respect to $\mathcal{T}$ if for any model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$:
- $\mathcal{I} \models \text{pre}_1$
Definition (Executability)
Given an acyclic TBox $\mathcal{T}$, a composite action $\alpha = \alpha_1, \ldots, \alpha_k$ where $\alpha_i = (\text{pre}_i, \text{occ}_i, \text{post}_i)$ and an ABox $\mathcal{A}$ we say that $\alpha$ is executable in $\mathcal{A}$ with respect to $\mathcal{T}$ if for any model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$:

- $\mathcal{I} \models \text{pre}_1$
- For all $i$ in $1 \leq i < k$, and all interpretations such that $\mathcal{I} \models_{\mathcal{T}} \mathcal{I}'$ we have $\mathcal{I}' \models \text{pre}_{i+1}$. (Recall that because we are dealing with acyclic ABoxes there is only one such interpretation $\mathcal{I}'$.)
Definition (Projection)
Given an acyclic TBox \( \mathcal{T} \), a composite action \( \alpha = \alpha_1, .., \alpha_k \) where \( \alpha_i = (\text{pre}_i, \text{occ}_i, \text{post}_i) \), an ABox \( \mathcal{A} \), we say that \( \phi \) is a consequence of applying \( \alpha \) in \( \mathcal{A} \) with respect to \( \mathcal{T} \) if for any model \( \mathcal{I} \) of \( \mathcal{A} \) and \( \mathcal{T} \) and any \( \mathcal{I}' \) such that \( \mathcal{I} \Rightarrow_{\alpha} \mathcal{I}' \) it is the case that \( \mathcal{I}' \models \phi \).
Consistency of Actions with TBoxes

- Executability is not sufficient to ensure that a composite action does not get stuck, i.e., that all the composite actions of an executable action will be carried out.

- It might be the case that we have a $\phi_1/\psi$ and $\phi_2/\neg\psi$ where $\phi_1$ and $\phi_2$ are both satisfied in the model $I$. In this case the action is said to be inconsistent with respect to $I$.

- Therefore to guarantee that an executable action is carried out without getting stuck we stipulate that each of the basic actions are consistent with any model $I$ of $A$ and $T$. 
Our aim is to find out complexity results for various (interesting) sublanguages of \textit{ALCQIO}.

Executability and Projection are mutually reducible in polynomial time. So we are free to focus on projection.

First we will look at some upper bound results.
Strategy of proof: show upper complexity bounds by reducing projection to a standard reasoning problem in DL.

Preliminary: Given a DL $\mathcal{L}$ we will denote by $\mathcal{LO}$ the extension of $\mathcal{L}$ with nominals.
Theorem
$L \in \{\text{ALC}, \text{ALCI}, \text{ALCO}, \text{ALCIO}, \text{ALCQ}, \text{ALCQO}, \text{ALCQT}\}$
Then the projection of composite actions in $L$ can be reduced in polynomial time to the problem of non-consistency in $L^O$ of an ABox relative to an acyclic TBox.
We define the complement of the projection problem wrt to an assertion $\phi$, an action $\alpha$, an ABox $A$, a TBox $T$ but this time we want to know whether there exist possible worlds $I$, $J$ such that $J$ follows from $I$ after applying the action $\alpha$ and $\neg\phi$ holds at $J$.

It turns out that we can reduce the complement of projection problem in $L$ to the consistency problem for Aboxes in $LO$.

Therefore solving the complement of the projection problem for $L$ cannot be more difficult than the consistency problem (since we can use an efficient algorithm for consistency to derive an efficient algorithm for the complement of projection).
This gives us an upper bound result. But for logics such as \textit{ALCO, ALCIO, ALCQO} where the complexity of ABox consistency for \( \mathcal{L} \) is the same as in \( \mathcal{LO} \) we also get matching lower bounds since it is very easy to reduce ABox nonconsistency to projection in \( \mathcal{L} \).

(\text{Since } \neg \top(a) \text{ is a consequence of applying the empty action } (\emptyset, \emptyset, \emptyset) \text{ iff there exists no model of } A \text{ and } T).$

So

- \textit{ALC, ALCQ, ALCIO, ALCQO} are PSPACE-complete.
- \textit{ALCIO} is EXPTIME-complete.
- \textit{ALCQIO} is co-NEXPTIME-complete.
For the logics $\text{ALCI}$ and $\text{ALCQI}$ where adding nominals gives a corresponding increase in the complexity of the ABox consistency problem we can still get lower bound results by reducing the satisfiability problem for $\text{ALCIO}(\text{ALCQIO})$ with a single nominal and an empty TBox to the projection problem. This gives us that

- $\text{ALCI}$ is EXPTIME-complete.
- $\text{ALCQI}$ is co-NEXPTIME-complete.
Semantics of services

Service
Let $\mathcal{T}$ acyclic TBox, atomic service $S = (pre, occ, post)$ for $\mathcal{T}$

- $pre$ ABox assertions, all must be true in order to execute service,

- $occ$ assertions that should not change by $S$, only allow primitive concepts,

- $post$ finite set of conditional post-conditions $\varphi/\psi$, only allow primitive concepts

How the application of an atomic service changes the world?
Assumption - interpretation domain is never changed by the application of a service

Idea
Interpretation of atomic concepts and roles should change as little as possible while still making post-condition true
Possible Models Approach

Precedence relation $\leq_{I,S,T}$ on interpretations, characterizes their
proximity to a given $I$.
We use $M_1 \nabla M_2$ to denote symmetric difference between sets $M_1$ and $M_2$.

Preferred interpretations
$I' \leq_{I,S,T} I''$ iff

$$A^I \nabla A^{I'} \setminus \{a^I|A(a) \in \text{occ}\} \subseteq A^I \nabla A^{I''}$$

$$s^I \nabla s^{I'} \setminus \{(a^I, b^I)|s(a, b) \in \text{occ}\} \subseteq s^I \nabla s^{I''}$$
Service application

Satisfaction of post-conditions

Pair \( I, I' \) satisfies set of post-conditions \( post(I, I' \models post) \) iff

\[
\forall (\varphi / \psi) \in post, I' \models \psi, \text{ whenever } I \models \varphi
\]

We say that \( S \) may transform \( I \) to \( I'(I \Rightarrow_{S} I') \) iff

1. \( I, I' \models post \), and
2. \( \not\exists J, I, J \models post \), \( J \neq I' \), and \( J \preceq_{I, S, T} I' \).

Since TBoxes are acyclic and post-conditions allow primitive concepts only, services without occlusions are deterministic, i.e.

\[
\forall I \in \mathcal{M}(T), \exists \leq_1 I', I \Rightarrow_{S} I'
\]
Application of services without occlusions

Let $\mathcal{T}$ - acyclic TBox, $S = (\text{pre}, \emptyset, \text{post})$ a service for $\mathcal{T}$, and for $\mathcal{I}, \mathcal{I}' \in \mathcal{M}(\mathcal{T})$, $I \xrightarrow{T} S I'$. $A$ - primitive concept, $s$ - role name, then

$$A^{\mathcal{I}'} := (A^{\mathcal{I}} \cup \{b^{\mathcal{I}} | \varphi/A(b) \in \text{post} \text{ and } I \models \varphi\}) \setminus \{b^{\mathcal{I}} | \varphi/\neg A(b) \in \text{post} \text{ and } I \models \varphi\},$$

$$s^{\mathcal{I}'} := (s^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | \varphi/s(a, b) \in \text{post} \text{ and } I \models \varphi\}) \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | \varphi/\neg s(a, b) \in \text{post} \text{ and } I \models \varphi\},$$
Problematic Extensions

Syntactic restrictions adopted in this approach:

1. Transitive roles are disallowed (although available in OWL-DL)
2. Only acyclic TBoxes are allowed
3. No complex concepts in post-conditions, (i.e. $\varphi/C(a)$ or $\varphi/\neg C(a)$ only)

Relaxing first restriction leads to *semantic* problems, removing second and third leads to *semantic* and *computational* problems.
Transitive roles

interpretation of transitive roles in \( ALCQI\O \)
transitive role \( r \in \mathcal{N}_tR \subset \mathcal{N}_R \) is interpreted as transitive relation \( r^I \) in all models \( I \)

Addition of transitive roles \( \mathcal{N}_tR \) no longer guarantees determinism for services without occlusions, i.e.

\[
I \Rightarrow^T_S I' \text{ and } I \Rightarrow^T_S I'' \text{ may not necessarily imply } I' = I''
\]

Due to the fact that \( \Rightarrow^T_S \) does not take into account \( r \in \mathcal{N}_tR \)
Transitive roles (contd.)

Consider $S = (\emptyset, \emptyset, \{\text{has part}(\text{car, engine})\}), \text{has part} \in \mathcal{N}_{tR}$, and a model $\mathcal{I}$

$$\Delta^\mathcal{I} := \{\text{car, engine, valve}\}$$
$$\text{has part}^\mathcal{I} := \{(\text{engine, valve})\}$$
$$z^\mathcal{I} := z \text{ for } z \in \Delta^\mathcal{I}.$$ 

We may have $I \xrightarrow{T_S} I', I \xrightarrow{T_S} I''$ and $I' \neq I''$, where

$$\text{has part}^\mathcal{I'} := \{(\text{car, engine}), (\text{engine, valve}), (\text{car, valve})\},$$ 

and

$$\text{has part}^\mathcal{I''} := \{(\text{car, engine})\},$$

applying $S$ in $\{\text{has part}(\text{engine, valve})\} \not\models \text{has part}(\text{car, engine})$ (counterintuitive)
Cyclic TBoxes and GCIs (general concept inclusion)

Problems

1. For *acyclic* TBoxes, the interpretation of primitive concepts uniquely determines the extension of defined ones, which is not the case for cyclic ones.

2. $\Rightarrow_{S}$ only takes into account primitive concepts

Consider the following example:

$A := \{\text{Dog}(a)\}$

$T := \{\text{Dog} \equiv \exists \text{parent}.\text{Dog}\}$

$post := \{\text{Cat}(b)\}$

(application of $S = (\emptyset, \emptyset, post)$ in $A$ w.r.t. $T$) $\not\models$ $\text{Dog}(a)$ (as one would intuitively expect)
Counter model construction

Define interpretation $\mathcal{I}$ as follows:

$$\Delta^\mathcal{I} := \{b\} \cup \{d_0, d_1, d_2, \ldots\}$$

$$Dog^\mathcal{I} := \{d_0, d_1, d_2, \ldots\}$$

$$Cat^\mathcal{I} := \emptyset$$

$$parent^\mathcal{I} := \{(d_i, d_{i+1}) | i \in \mathbb{N}\}$$

$$a^\mathcal{I} := d_0$$

$$b^\mathcal{I} := b$$

Define $\mathcal{I}'$ as $\mathcal{I}$ except for $Cat^{\mathcal{I}'} := \{b\}$ and $Dog^{\mathcal{I}} := \emptyset$.

Semantic issue

$Dog$ - defined concept, not considered in $\Rightarrow^\mathcal{T}_S$, hence

$$\mathcal{I} \models A, I \Rightarrow^\mathcal{T}_S I', \text{ and } \mathcal{I}' \not\models Dog(a)$$
Possible solutions

- Include defined concepts in the minimization of changes, i.e. treat them in $\Rightarrow^T_S$
  - infeasible, even minimization of Boolean concepts induces technical problems
- Use semantics that regains the “definitorial power” of acyclic TBoxes (Fixpoint semantics)
  - in the case of least or greatest fixpoint semantics proposed by Nebels, indeed primitive concepts uniquely determine defined ones


*Post conditions* are of the form $\varphi/\psi$, if we allow *arbitrary* (complex) assertions $\varphi$ and $\psi$ we run into *Semantic problems*.

**Example**

Let $a : \exists r. A$ be a post-condition, not satisfied before the execution of the service, then *any* $x \in \Delta^I$ may be chosen to satisfy $(a^I, x) \in r^I$ and $x \in A^I$ after execution.

e.g.

$$S := (\emptyset, \emptyset, \{ \text{mary} : \exists \text{has_child}. \neg \text{Female} \})$$

$$A := \{ \text{Female} (\text{mary}) \}$$

(applying $S$ in $A$) $\not\models \text{Female} (\text{mary})$
Computational problems with GCIs

GCI is an expression $C \sqsubseteq D$, with $C$ and $D$ (possibly complex) concepts. It generalizes cyclic TBoxes, i.e. $A \equiv C$ may be rewritten as two GCIS $A \sqsubseteq C$ and $C \sqsubseteq A$

Minimization of all concepts

▶ GCIs do not allow obvious partitioning of complex concepts into primitive and defined.
▶ Thus $\Rightarrow^T_S$ has to minimize all concepts (infeasible as mentioned before)

Executability and projection for generalized services become undecidable
Proven by redaction of the domino problem to non-consequence and non-executability
Conclusion and Future work

Main technical results

▶ Standard problems in reasoning about actions (projection, execution) become decidable
▶ Complexity of inferences is determined by the complexity of standard DL reasoning in $\mathcal{L}$ extended by nominals

Possible extensions of formalism

▶ Consider cyclic TBoxes and fixpoint semantics
▶ Decide projection problem through progression instead of regression
▶ Check for which of the extensions of Reiter’s action formalism these results still hold
▶ Allow for more complex composition of actions
▶ Support automatic composition of services, how planning fits in this formalism
Lemma. Executability and projection can be reduced to each other in polynomial time

Proof

$S_1, \ldots, S_k$ with $S_i = (pre_i, occ_i, post_i)$ composite service for $T$. $S$ is executable in $A$ iff

$(i) \forall M \in \mathcal{M}(A, T), pre_1$ satisfied in $M$

$(ii) \forall i \in [1, k), (\text{application of } S_1, \ldots, S_i \text{ in } A) \models pre_{i+1}$

Condition $(ii)$ is a projection problem, $(i)$ is a projection problem for $S = (\emptyset, \emptyset, \emptyset)$
Polynomial reduction from *executability* to *projection* and vice versa (contd)

**Proof (contd)**

Conversely, assume we want to know whether \((\text{application of } S_1, \ldots, S_k \text{ in } A) \models \varphi?)

Consider, \(S'_1, \ldots, S'_k, S'\), where \(S'_i = (\emptyset, \text{occ}_i, \text{post}_i), \forall i \in [i, k]\), and \(S' = (\{\varphi, \emptyset, \emptyset\})\). Then

\[
\text{application } S_1, \ldots, S_k \text{ in } A \models \varphi \text{ iff } S'_1, \ldots, S'_k, S' \text{ is executable}
\]
Services without occlusions - instance of SitCalc

- Expand $\mathcal{T}$ and replace in $A$ and in $S_1, \ldots, S_k$
- Translate $\Rightarrow_{\mathcal{T}}$ into first-order logic (action pre-conditions and successor state axioms)
- Primitive concepts and roles regarded as fluents
- ABox first-order translation is the initial state
- projection and executability are instances of Reiter’s definitions

However, this translation leads to a standard first-order theory, which is not in the scope of what GOLOG can handle.
Franz Baader, Carsten Lutz, Ulrike Sattler and Frank Wolter.  
*Integrating description logics and action formalisms: First Results.*  

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider.  
*The description logic handbook: theory, implementation, and applications.*  

Raymond Reiter.  
*The frame problem in situation the calculus: a simple solution (sometimes) and a completeness result for goal regression.*  