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

  Move(x,y) : robot is moving from position x to y

  Fluents that describe the world

  is_carrying(ball,S0)=false
  is_carrying(ball, do(pick_up(ball, S0))) = true

  Situation represent history of action occurrences

  do(move(2,3), S0) : denotes a new situation after performing action move(2,3) in initial situation S0

[1] Reiter, R., "Knowledge in Action", MIT Press, 2001.

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,

$\mathbf{Poss}(\mathbf{pickup}(\mathbf{o}), \mathbf{s}) \cup \mathbf{location}(\mathbf{s}) = (\mathbf{x}, \mathbf{y}) \rightarrow$
$\mathbf{location}(\mathbf{do}(\mathbf{pickup}(\mathbf{o}), \mathbf{s})) = (\mathbf{x}, \mathbf{y})$
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

  $$Poss(a,s) \vee \gamma_{+F}(x,a,s) \rightarrow F(x, do(a,s))$$
  $$Poss(a,s) \vee \gamma_{-F}(x,a,s) \rightarrow \neg F(x, do(a,s))$$

  γ_{+F} describes the conditions under which action **a** in situation **s** makes the fluent **F** become true in the successor situation **do(a,s)**.
  γ_{-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: $\prod_{A(s)} \supset Poss(A, s)$
- Unique names axioms for the actions and for states

Reiter, R., "The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression", in AI and Mathematical Theory of Computation, Academic Press, 359–380, 1991.

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 it sub languages.

Modeling framework

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.

$$Father \equiv Male \sqcap hasChild$$
$$hasChild \equiv Father \sqcup Mother$$

Not an Acyclic TBox

Modeling framework (Continued)

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) . . .

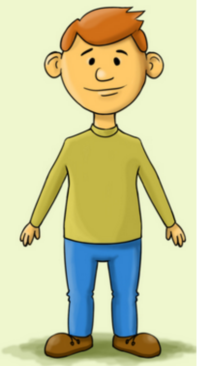
example: Woman(Shelly), Male(John), . . . Assertion with role name s in the form: s(a,b), s(b,c), or ¬s(a,b) example: Father (John, haschild)

Modeling framework (Definition 1)

Let $\mathbf{T}$ = Acyclic Tbox
Atomic action $\alpha = (\mathit{pre}, \ \mathit{occ}, \ \mathit{post})$

- a finite set *pre* of ABox assertions, the *pre-conditions*;
- a finite set occ of occlusions of the form $A(a)$ or $s(a, b)$
- *a finite set post of conditional post-conditions of the form* ϑ/ψ*, where* ϑ *is an ABox asertion and* ψ *is a primitive literal for* $\mathbf{T}$

Example of action definition



α_1 Opening a bank account in Italy

Ok, can you deposit 1000 euro? Do you have proof of address

pre_1

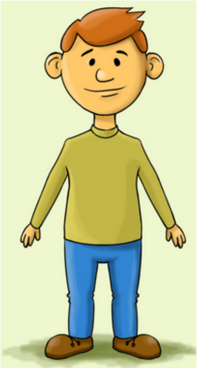
$\{Eligible\_bank(a), \exists holds.Proof\_address(a)\}$

$post_1$

$\{T(a)/holds(a, b),$

$\exists holds.letter(a)/B\_acc\_credit(b),$

$\neg\exists holds.letter(a)/B\_acc\_no\_credit(b)\}$



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

pre_2
$\{parents\_of(a,c), \exists hold.B_acc(a)\}$
$post_2$
$\{T(a)/receives\_c\_benefit\_for(a,c)\}$

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

$$
\begin{aligned}
A^{+}: &= \{b^{I} \mid \varphi/A(b) \in \text{post and } I \models \varphi\} \\
A^{-}: &= \{b^{I} \mid \varphi/-A(b) \in \text{post and } I \models \varphi\} \\
I_A: &= \left(\Delta^{I} \setminus \{b^{I} \mid A(b) \in \text{occ }\}\right) \cup \left(A^{+} \cup A^{-}\right) \\
s^{+}: &= \{\left(a^{I}, b^{I}\right) \mid \varphi/s(a,b) \in \text{ post and } I \models \varphi\} \\
s^{-}: &= \{\left(a^{I}, b^{I}\right) \mid \varphi/-s(a,b) \in \text{ post and } I \models \varphi\} \\
I_s: &= \left(\left(\Delta^{I} \times \Delta^{I}\right) \setminus \left(\left(a^{I}\right), b^{I}\right) \mid s(a,b) \in \text{occ }\right) \cup \left(s^{+} \cup s^{-}\right)
\end{aligned}
$$

Modeling framework (Definition 2)

Action α may transform $\mathbf{I}$ to $\mathbf{I'}$ iff, for each primitive concept A and role name s,

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

The composite action $\alpha_1 \ldots \alpha_k$ may transform $\mathbf{I}$ to $\mathbf{I'}$ 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 $\mathbf{I'}$ such that

$$I \implies {}^{T}_{\alpha} I'$$

Thus, actions are deterministic.

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

such that both ϑ_1 and ϑ_2 are satisfied in $\mathbf{I}$, then there is no successor model $\mathbf{I'}$. So action is inconsistent with $\mathbf{I}$.

Reasoning Problems

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

- Executability: are all the preconditions of α satisfied in worlds considered possible?
- Projection: does a given assertion hold after applying α?

Consistency

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 $\mathcal{T}$.

Reasoning Problems - Executability

Definition (Executability)

Given an acyclic TBox $\mathcal{T}$, a composite action $\alpha = \alpha_1, .., \alpha_k$ where $\alpha_i = (pre_i, occ_i, post_i)$ and an ABox $\mathcal{A}$ we say that α is *executable* in $\mathcal{A}$ with respect to $\mathcal{T}$ if for any model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$:

- $\mathcal{I} \models pre_1$

Definition (Executability)

Given an acyclic TBox $\mathcal{T}$, a composite action $\alpha = \alpha_1, .., \alpha_k$ where $\alpha_i = (pre_i, occ_i, post_i)$ and an ABox $\mathcal{A}$ we say that α is *executable* in $\mathcal{A}$ with respect to $\mathcal{T}$ if for any model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$:

- $\mathcal{I} \models pre_1$
- For all i in $1 \leq i < k$, and all interpretations such that $\mathcal{I} \Rightarrow^{\mathcal{T}}_{\alpha} \mathcal{I}'$ we have $\mathcal{I}' \models pre_{i+1}$. (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 $\alpha_i = (pre_i, occ_i, post_i)$, an ABox $\mathcal{A}$, we say that ϕ is *a consequence of applying* α *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 $\mathcal{I} \Rightarrow_{\alpha}^{\mathcal{T}} \mathcal{I}'$ 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 ϕ_1/ψ and $\phi_2/\neg\psi$ where ϕ_1 and ϕ_2 are both satisfied in the model $\mathcal{I}$. In this case the action is said to be *inconsistent* with respect to $\mathcal{I}$.
- 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 $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$.

Complexity for Execution and Projection

- Our aim is to find out complexity results for various (interesting) sublanguages of $\mathcal{ALCQIO}$.
- 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.

Complexity for Execution and Projection

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

$\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCI}, \mathcal{ALCO}, \mathcal{ALCIO}, \mathcal{ALCQ}, \mathcal{ALCQO}, \mathcal{ALCQI}\}$
Then the projection of composite actions in $\mathcal{L}$ can be reduced in polynomial time to the problem of non-consistency in $\mathcal{LO}$ of an ABox relative to an acyclic TBox.

Complexity for Execution and Projection

- We define the complement of the projection problem wrt to an assertion ϕ, an action α an ABox $\mathcal{A}$, a TBox $\mathcal{T}$ but this time we want to know whether there exist possible worlds $\mathcal{I}$, $\mathcal{J}$ such that $\mathcal{J}$ follows from $\mathcal{I}$ after applying the action α and $\neg\phi$ holds at $\mathcal{J}$.
- It turns out that we can reduce the complement of projection problem in $\mathcal{L}$ to the consistency problem for Aboxes in $\mathcal{LO}$.
- Therefore solving the complement of the projection problem for $\mathcal{L}$ 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).

Complexity Results

- This gives us an upper bound result. But for logics such as $\mathcal{ALCO}, \mathcal{ALCIO}, \mathcal{ALCQO}$ 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 non consistency to projection in $\mathcal{L}$.
- (Since $\neg\top(a)$ is a consequence of applying the empty action $(\emptyset, \emptyset, \emptyset)$ iff there exists no model of $\mathcal{A}$ and $\mathcal{T}$).
- So
  - $\mathcal{ALC}, \mathcal{ALCO}, \mathcal{ALCIO}, \mathcal{ALCQO}$ are PSPACE-complete.
  - $\mathcal{ALCIO}$ is EXPTIME-complete.
  - $\mathcal{ALCQIO}$ is co-NEXPTIME-complete.

Complexity Results

- For the logics $\mathcal{ALCI}$ and $\mathcal{ALCQI}$ 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 $\mathcal{ALCIO}$($\mathcal{ALCQIO}$) with a single nominal and an empty TBox to the projection problem This gives us that
  - $\mathcal{ALCI}$ is EXPTIME-complete.
  - $\mathcal{ALCQI}$ is co-NEXPTIME-complete.

Semantics of services

Service

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

- *pre* ABox assertions, all must be true in order to execute service,
- *occ* assertions that should not change by $\mathcal{S}$, only allow primitive concepts,
- *post* finite set of *conditional post-conditions* φ/ψ, 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 $\preceq_{\mathcal{I},\mathcal{S},\mathcal{T}}$ on interpretations, characterizes their *proximity* to a given $\mathcal{I}$.
We use $M_1 \nabla M_2$ to denote symmetric difference between sets M_1 and M_2.

Preferred interpretations

$\mathcal{I}' \preceq_{\mathcal{I},\mathcal{S},\mathcal{T}} \mathcal{I}''$ **iff**

$$A^{\mathcal{I}} \nabla A^{\mathcal{I}'} \setminus \{a^{\mathcal{I}} | A(a) \in occ\}) \subseteq A^{\mathcal{I}} \nabla A^{\mathcal{I}''}$$
$$s^{\mathcal{I}} \nabla s^{\mathcal{I}'} \setminus \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | s(a,b) \in occ\}) \subseteq s^{\mathcal{I}} \nabla s^{\mathcal{I}''}$$

Service application

Satisfaction of post-conditions

Pair $\mathcal{I}, \mathcal{I}'$ satisfies set of post-conditions *post*($\mathcal{I}, \mathcal{I}' \models post$) **iff**

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

We say that $\mathcal{S}$ *may transform* $\mathcal{I}$ to $\mathcal{I}'$($I \Rightarrow_{\mathcal{S}}^{\mathcal{T}} I'$) **iff**

1. $\mathcal{I}, \mathcal{I}' \models post$, *and*
2. $\nexists \mathcal{J}, \mathcal{I}, \mathcal{J} \models post, \mathcal{J} \neq \mathcal{I}'$, *and* $\mathcal{J} \preceq_{\mathcal{I},\mathcal{S},\mathcal{T}} \mathcal{I}'$.

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

$$\forall \mathcal{I} \in \mathcal{M}(\mathcal{T}), \exists_{\leq 1} \mathcal{I}', I \Rightarrow_{\mathcal{S}}^{\mathcal{T}} I'$$

Application of services without occlusions

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

$$
\begin{aligned}
A^{\mathcal{I}'} &:= (A^{\mathcal{I}} \cup \{b^{\mathcal{I}} | \varphi / A(b) \in post \text{ and } \mathcal{I} \models \varphi\}) \setminus \\
&\qquad \{b^{\mathcal{I}} | \varphi / \neg A(b) \in post \text{ and } \mathcal{I} \models \varphi\}, \\
s^{\mathcal{I}'} &:= (s^{\mathcal{I}} \cup \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | \varphi / s(a, b) \in post \text{ and } \mathcal{I} \models \varphi\}) \setminus \\
&\qquad \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | \varphi / \neg s(a, b) \in post \text{ and } \mathcal{I} \models \varphi\},
\end{aligned}
$$

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 $\mathcal{ALCQIO}$

transitive role $r \in \mathcal{N}_{tR} \subset \mathcal{N}_{\mathcal{R}}$ is interpreted as transitive relation $r^{\mathcal{I}}$ in all models $\mathcal{I}$

Addition of *transitive roles* $\mathcal{N}_{tR}$ no longer guarantees *determinism* for services without occlusions, i.e.

$$I \Rightarrow_{\mathcal{S}}^{\mathcal{T}} I' \text{ and } I \Rightarrow_{\mathcal{S}}^{\mathcal{T}} I'' \text{ may not necessarily imply } I' = I''$$

Due to the fact that $\Rightarrow_{\mathcal{S}}^{\mathcal{T}}$ does not take into account $r \in \mathcal{N}_{tR}$

Transitive roles (contd.)

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

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

We may have $I \Rightarrow^{\mathcal{T}}_{\mathcal{S}} I'$, $I \Rightarrow^{\mathcal{T}}_{\mathcal{S}} I''$ and $I' \neq I''$, where

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

and

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

applying $\mathcal{S}$ in $\{has\_part(engine, valve)\} \not\models has\_part(car, engine)$ (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. $\Rightarrow_{\mathcal{S}}^{\mathcal{T}}$ only takes into account primitive concepts

Consider the following example:

$$\begin{aligned} \mathcal{A} &:= \{Dog(a)\} \\ \mathcal{T} &:= \{Dog \equiv \exists parent.Dog\} \\ post &:= \{Cat(b)\} \end{aligned}$$

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

Counter model construction

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

$$\begin{aligned}
\Delta^{\mathcal{I}} &:= \{b\} \cup \{d_0, d_1, d_2, \ldots\} \\
Dog^{\mathcal{I}} &:= \{d_0, d_1, d_2, \ldots\} \\
Cat^{\mathcal{I}} &:= \varnothing \\
parent^{\mathcal{I}} &:= \{(d_i, d_{i+1} | i \in \mathbb{N}\} \\
a^{\mathcal{I}} &:= d_0 \\
b^{\mathcal{I}} &:= b
\end{aligned}$$

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

Semantic issue

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

$$\mathcal{I} \models \mathcal{A}, I \Rightarrow_{\mathcal{S}}^{\mathcal{T}} I', \text{ and } \mathcal{I}' \not\models Dog(a)$$

Possible solutions

- Include defined concepts in the minimization of changes, i.e. treat them in $\Rightarrow_{\mathcal{S}}^{\mathcal{T}}$
  - 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

Complex Concepts in Post-Conditions

Post conditions are of the form φ/ψ, if we allow *arbitrary* (complex) assertions φ and ψ 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* $x \in \Delta^{\mathcal{I}}$ may be chosen to satisfy $(a^{\mathcal{I}}, x) \in r^{\mathcal{I}}$ and $x \in A^{\mathcal{I}}$ after execution.

e.g.

$$\mathcal{S} := (\varnothing, \varnothing, \{mary : \exists has\_child.\neg Female\})$$
$$\mathcal{A} := \{Female(mary)\}$$

(applying $\mathcal{S}$ in $\mathcal{A}$) $\not\models Female(mary)$

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 $\Rightarrow^{\mathcal{T}}_{\mathcal{S}}$ 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

Polynomial reduction from *executability* to *projection* and vice versa

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 $\mathcal{T}$. S is executable in $\mathcal{A}$ **iff**

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

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

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
(application of $S_1, \ldots, S_k$ in $\mathcal{A}) \models \varphi$?
Consider, $S'_1, \ldots, S'_k, S'$, where $S'_i = (\varnothing, occ_i, post_i), \forall i \in [i, k]$,
and $S' = (\{\varphi, \varnothing, \varnothing\})$. Then

$$\text{application } S_1, \ldots, S_k \text{ in } \mathcal{A} \models \varphi \,\textbf{iff}\, S'_1, \ldots, S'_k, S' \text{ is executable}$$

Relationship to SitCalc

Services without occlusions - instance of SitCalc

- Expand $\mathcal{T}$ and replace in $\mathcal{A}$ and in $S_1, \ldots, S_k$
- Translate $\Rightarrow^{\mathcal{T}}_{\mathcal{S}}$ 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

References

- Franz Baader, Carsten Lutz, Ulrike Sattler and Frank Wolter.
  Integrating description logics and actionformalisms: First Results.
  In Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005), 572-577, 2005.

- Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider.
  The description logic handbook: theory, implementation, and applications.
  Cambridge University Press New York, NY, USA, 2003.

- Raymond Reiter.
  The frame problem in situation the calculus: a simple solution (sometimes) and a completeness result for goal regression.
  Artificial intelligence and mathematical theory of computation, 359-380, 1991.