Decidable Reasoning in a Modified Situation Calculus

Yilan Gu
Dept. of Computer Science
University of Toronto

Mikhail Soutchanski
Dept. of Computer Science
Ryerson University

January 12\textsuperscript{nd}, 2007
Shopping Online

Requests (E.g., buy/return books)

Web Servers (E.g., Amazon)

Arrangement

Shipping

Inventory

Clients (buyers)
Motivations

• Usually suppliers (Web servers) could not get complete information (OWA)
• Need composition of atomic services to achieve the clients’ requests
• Integrating Semantic Web with Web services
• Representing the dynamics
  – What needs to be represented?
    • Atomic services (i.e., actions), dynamic environment, effects of actions
  – Requirements:
    • Represent actions with arguments varying over large/infinite domains (E.g., people, weight, time)
    • Be able to represent knowledge such as “there exist some …”
• What do we care about?
  – Reasoning: Executability Problem, Projection Problem and Progression Problem
  – Expectations: efficient reasoning (here, decidability), soundness
The Situation Calculus (SC)

- A first-order logic language
- Three sorts:
  - Actions: \textit{buyBook}(x,y), \textit{returnBook}(x,y), \ldots
  - Situations: \textit{S}_0, \textit{do}(a,s), \textit{do}([a_1,\ldots,a_n],s)
  - Objects: things other than actions and situations
- Fluents: system features whose truth values may vary
  \textit{instore}(x,s), \textit{boughtBook}(x,y,s), \textit{bought}(x,y,s)\ldots
- Basic action theory (BAT) \( \mathcal{D} \)
  - Precondition axioms for actions \( \mathcal{D}_{ap} \):
    \[
    \text{Poss}(\text{buyBook}(x,y),s) \equiv \text{client}(x) \land \text{book}(y) \land \text{instore}(y,s)
    \]
  - Successor state axioms \( \mathcal{D}_{ss} \):
    \[
    \text{bought}(x,y,\text{do}(a,s)) \equiv a = \text{buyBook}(x,y) \lor a = \text{buyCD}(x,y) \\
    \text{bought}(x,y,s) \land \neg (a = \text{returnBook}(x,y) \lor a = \text{returnCD}(x,y))
    \]
  - Axioms for initial database \( \mathcal{D}_{S_0} \):
    - Knowledge known to be true in the situation \( S_0 \)
    - Non-changeable facts
    - Open World Assumption: the initial theory about \( S_0 \) is logically incomplete
Reasoning about Actions in SC

• Projection problem: given FO sentence $W$, decide whether $\mathcal{D} \models W$
• Executability problem: given a sequence of actions $A_1; \ldots; A_n$, decide whether $\mathcal{D} \models Poss(A_1,S_0) \land Poss(A_2,do(A_1,S_0)) \land \ldots \land Poss(A_n, do([A_1, \ldots, A_{n-1}], S_0))$
• Key reasoning mechanism -- regression operator $\mathcal{R}$.
• Successor state axioms support regression in a natural way:
  If $F(x_1, \ldots, x_n, do(a,s)) \equiv \Psi_F(x_1, \ldots, x_n, a, s)$, then
  $\mathcal{R} [F(t_1, \ldots, t_n, do(A,S))] = \mathcal{R} [\Psi_F(t_1, \ldots, t_n, A,S)]$.

\begin{align*}
W_0(S_0) & \xrightarrow{\mathcal{R}} \ldots \xrightarrow{\mathcal{R}} W'(do[a_1, \ldots, a_{n-1}], S_0) & \xleftarrow{\mathcal{R}} W(do[a_1, \ldots, a_n], S_0)
\end{align*}

• Important properties for regression:
  $\mathcal{D} \models W \equiv \mathcal{R}[W]$, $\mathcal{D} \models W$ iff $\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$.

**Advantage**: representing actions and effects very compactly.

**Disadvantage**: reasoning for actions in general is undecidable under the open world assumption (OWA).

**Solution**: Consider a fragment of first-order logic $C^2$. 
Description Logics vs. C²

- **Description logics**
  - Foundation of OWL
  - Different varieties
  - \( \text{ALCQIO}(\sqcap, \sqcup, \neg, |, id) \)

- **C²**: a fragment of FOL
  - At most two variables \( x, y \)
  - No function symbols
  - Add counting quantifiers \( \exists^{\ge n}, \exists^{\le n} \)

- \( \text{ALCQIO}(\sqcap, \sqcup, \neg, |, id) \) vs. C²
  - Concept names \( \iff \) unary predicates
    - \( \text{instore} \iff \text{instore}(x) \)
  - Role names \( \iff \) binary predicates
    - \( \text{boughtBook} \iff \text{boughtBook}(x, y) \)
  - E.g., \( \exists^{\ge n} R.C \iff \exists^{\ge n} y.R(x, y) \land C(y) \), \( \forall R.C \iff \forall y.R(x, y) \supset C(y) \)
    - \( \neg C \iff \neg C(x) \), \( C1 \sqcap C2 \iff C1(x) \land C2(x) \)

- **Advantages**
  - The features in Semantic Webs can be easily represented in C².
  - The reasoning in C² can also be easily translated into DLs.
  - May use current existing efficient DL reasoners for C² formulas.

\( \text{ALCQIO}(\sqcap, \sqcup, \neg, |, id) \iff C² \), the translation algorithm is linear in the size of the given formula, both are decidable even under OWA.
The Decidable Situation Calculus $\mathcal{L}^{DL}_{\text{SC}}$

Purpose: to ensure the formula resulting from regression is a $C^2$ formula.

- Sorts:
  - Terms of objects are either variable $x$, variable $y$, or constants
  - Action functions have at most two arguments
  - Variable symbol $a$ of sort $action$ and symbol $s$ of sort $situation$ are the only additional variable symbols

- Fluents with either two or three arguments:
  - (Dynamic) concepts $instore(x,s)$, 
  - (Dynamic) roles $boughtBook(x,y,s)$, $bought(x,y,s)$, 

- Facts with either one or two arguments:
  - (Static) concepts $person(x)$, $client(x)$, $book(x)$, $cd(x)$, 
  - (Static) roles $hasCreditCard(x,y)$, 

- Logic: add counting quantifiers $\exists^{\geq n}$, $\exists^{\leq n}$
Basic Action Theory of $\mathcal{L}_{SC}^{DL}$

- **Precondition axioms**: The RHS is a $C^2$ formula if $s$ is suppressed
- **Success state axioms**:
  - Allow counting quantifiers
  - Variables $a$ and $s$ are free in the RHS of the axioms
  - Moreover, $x,y,a$ and $s$ are the only variables (both free and quantified)
- **Axioms for initial databases**: Each axiom is a $C^2$ formula if $S_0$ is suppressed
- **Acyclic TBox axioms**:
  - Dynamic ones: $C(x,s) \equiv \Phi_c(x,s)$ ($C$ defined dynamic concept)
  - Static ones: $C(x) \equiv \Phi_c(x)$ (provided in the $D_{S0}$)
  - The RHS is $C^2$ when the situation argument $s$ is suppressed
    E.g., $\text{valCust}(x,s) \equiv \text{person}(x) \land (\exists y^3) \text{bought}(x,y,s)$
    $\text{client}(x) \equiv \text{person}(x) \land (\exists y) \text{hasCreditCard}(x,y)$
  - Reasoning: use lazy unfolding for Dynamic axioms
- **RBox axioms**:
  - $R1 \supset R2$ for roles $R1$, $R2$
    E.g., $\text{boughtBook}(x,y,s) \supset \text{bought}(x,y,s)$, $\text{boughtCD}(x,y,s) \supset \text{bought}(x,y,s)$
  - Correctly compiled in $D_{SS}$, i.e., $D \models (\forall x,y,s).R1(x,y)[s] \supset R2(x,y)[s]$
Reasoning: Regression + Lazy Unfolding

- Expectations
  - Resulting formula should be $C^2$ if $S_0$ is suppressed
  - Be able to handle dynamic TBox axioms

- Reiter’s regression operator is not suitable:
  - It introduces new variables to deal with quantifiers

- Formula $W$ that is regresssable in $\mathcal{L}_{SC}^{DC}$
  - The situation terms in $W$ are ground
  - Variables in $W$ can only include $x, y$

- Modified regression operator $\mathcal{R}$
  - When $W$ is not atomic, the operator is still defined recursively
    - E.g., $\mathcal{R}[W_1 \land W_2] = \mathcal{R}[W_1] \land \mathcal{R}[W_2]$, …
  - Add $\mathcal{R}[\exists^n v. W] = \exists^n v. \mathcal{R}[W]$
  - Reuse variables $x$ and $y$ when $W$ is atomic
  - Lazy unfolding: use TBox axioms when $W$ is a defined dynamic concept
  - Apply Unique name axioms axioms for actions
A Regression Example in $\mathcal{L}_{SC}^{DL}$

- Example: online shopping

\[ A1 = buyCD(Tom, BackStreetBoys) \]
\[ A2 = buyBook(Tom, HarryPotter) \]
\[ A3 = buyBook(Tom, TheFirm) \]

\[ \mathcal{R}[\exists x. valCust(x, do([A1,A2,A3],S_0))] \]
\[ = \mathcal{R}[\exists x. person(x) \land (\exists y \geq 3) bought(x, y, do([A1,A2,A3], S_0))] \]

(lazy unfolding)
\[ = (\exists x). person(x) \land (\exists y \geq 3) \mathcal{R}[bought(x, y, do([A1,A2,A3], S_0))] \]
\[ = \ldots \text{ (recursively do regression using the successor state axioms)} \]
\[ = (\exists x). person(x) \land (\exists y \geq 3) [(x=Tom \land y = TheFirm) \lor \]
\[ \phantom{(x=Tom \land y = HarryPotter) \lor}
\[ \phantom{(x=Tom \land y = HarryPotter) \lor}
\[ bought(x,y,S_0)] \]
Important Properties

- Suppose $W$ is a regressable formula of $\mathcal{L}_{SC}^{DL}$ with BAT $\mathcal{D}$
  - The regression $\mathcal{R}[W]$ terminates in a finite number of steps
  - $\mathcal{R}[W]$ is a $C^2$ formula if $S_0$ is suppressed
  - $\mathcal{D} \models W \equiv \mathcal{R}[W]$
  - $\mathcal{D} \models W$ iff $\mathcal{D}_{S_0} \models \mathcal{R}[W]$
- The problem whether is $\mathcal{D} \models W$ is **decidable**
  - $\mathcal{D}_{S_0} \models \mathcal{R}[W]$ is a decidable reasoning in $C^2$
- When the SSA for $F$ is context-free, the computational complexity of answering the queries of ground term $F(X,S)$ is co-NEXPTIME
- Executability problems and projection problems are **decidable** in $\mathcal{L}_{SC}^{DL}$
  - Whether a composite service is executable
  - Whether desired/undesired properties will be true/false after the execution
Classical Progression

• Regression is not practical when executing a very large sequence of actions
• Progression: to compute the new theory given the current theory
• [Reiter 2001] A set of sentences $D_a$ is the classical progression of the initial KB $D_0$ (wrt BAT $D$) after performing a ground action $a$ in the situation $S_0$ iff
  – $D_a$ is uniform in $do(a, S_0)$;
  – $D \models D_a$;
  – for every model $M_a$ of $(D \setminus D_0) \cup D_a$, there is a model $M$ of $D$ such that $M_a$ and $M$ have the same domain and interpret situation independent predicates, function symbols, Poss and all fluents about the future of $do(a, S_0)$ identically.
• The classical progression of a finite first-order knowledge base (KB) is not always FOL definable
A modified progression in $\mathcal{L}_{sc}^{DL}$

- The (classical) progression of a KB in $\mathcal{L}_{sc}^{DL}$ is not always FOL definable, hence is not definable in $\mathcal{L}_{sc}^{DL}$
- The definability of a finite KB in $\mathcal{L}_{sc}^{DL}$ remains open
- Consider a (weaker than classical) modified progression in $\mathcal{L}_{sc}^{DL}$ for a CNF-based KB for a local-effect BAT
- A CNF-based KB
  - More general than proper KBs defined in [Liu & Levesque 2005]
  - Includes two parts:
    1. Situation independent facts
    2. Conjunctions of disjunctions of equality-based formulas
  - An example
    \[
    \left( \forall x (x = B_1 \supset \neg \text{ontable}(x)) \lor \forall y (y \neq B_2 \supset \text{ontable}(y)) \right) \land \\
    \forall z (z \neq B_3 \land z \neq B_4 \supset \text{hold}(z))
    \]
- A local-effect BAT: every SSA axiom is local-effect, i.e.,
  \[
  F(x, \text{do}(A, s)) \equiv x = B_1 \land p_1(s) \lor \ldots \lor x = B_m \land p_m(s) \lor \\
  F(x, s) \land \neg (x = C_1 \land q_1(s) \lor \ldots \lor x = C_n \land q_n(s))
  \]
  where $s$ is the only variable (both free and quantified) in any $p_i$ and $q_j$. 

A Progression Algorithm & Properties

• provided an algorithm to obtaining a modified progression of a CNF-based KB after executing a ground action wrt a local-effect BAT

• The intuition of the algorithm
  – Keep all situation independent information
  – Add truth values for each fluent for those objects where it will definitely become true (or false)
  – Update the remaining consistent information by removing conflicting knowledge for objects from the current KB

• Properties
  – If the given BAT is consistent, so is the modified progression
  – The modified progression is (classically) sound, i.e., any model of the classical progression of the current KB wrt the given BAT is a model of the modified progression

• Open problem
  – Under what cases, the modified progression will be (classically) complete, i.e., any model of the modified progression of the current KB wrt the given BAT is a model of the classical progression
Discussions and Future Work

- **Conclusions**
  - Formalize a decidable language suitable for Web services
  - Have compact powerful expression power
  - Consider the knowledge base progression/update problem in $L_{DL}^{SC}$

- **Other related research**
  - Web services
    - [McIlraith & Son 2002] assumes that all sufficient information is available
    - [Berardi et al. 2003] uses propositional dynamic logic to model services
      - e-services $\rightarrow$ constants, fluents $\rightarrow$ F(s) (propositional fragment of SC)
    - [Artale & Franconi 2001] extends DLs with temporal logics to capture the change of the world over time instead of caused by actions
    - [Baader et al. 2005] defines a service using a triple of sets of DL formulas
  - Progression
    - [Liu & Levesque 2005] considers a weaker progression of proper KBs
    - [Vassos & Levesque 2007] considers progression for functional fluents
    - [Claßen & Lakemeyer 2007] proposes a progression of an ADL database

- **Possible future work**
  - Implementations
  - Consider open problems such as
    - FOL definability of a finite KB in the modified SC
    - classical completeness of the modified progression