Decidable Reasoning in a Modified Situation Calculus

Yilan Gu
Dept. of Computer Science
University of Toronto
Toronto, Canada

Mikhail Soutchanski
Dept. of Computer Science
Ryerson University
Toronto, Canada

March 27, 2007
Shopping Online

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

Web Services Providers

Arrangement

Inventory

Shipping

Clients (customers)
Motivations

• Usually Web servers do not have complete information (OWA)
• Need composition of atomic services to implement clients’ requests
• Integrating Semantic Web (OWL) with Web services
• Representing the dynamics
  – What needs to be represented?
    • Atomic services (i.e., primitive actions) in dynamic environment:
      effects of actions, preconditions for 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, Progression Problem
  – Expectations: efficient reasoning (here, decidability), soundness
The Situation Calculus (SC)

- A first-order logic language
- Three sorts:
  - Actions: \( buyBook(x,y) \), \( returnBook(x,y) \), …
  - Situations: \( S_0 \), \( do(a,s) \), \( do([a_1,...,a_n],s) \)
  - Objects: things other than actions and situations
- Fluents: domain features whose truth values may vary
  \( instore(x,s) \), \( boughtBook(x,y,s) \), \( bought(x,y,s) \) …
- Basic action theory (BAT) \( \mathcal{D} \)
  - Precondition axioms for actions \( \mathcal{D}_{ap} \):
    \[ Poss(buyBook(x,y),s) \equiv client(x) \land book(y) \land instore(y,s) \]
  - Successor state axioms \( \mathcal{D}_{ss} \):
    \[ bought(x,y,do(a,s)) \equiv a = buyBook(x,y) \lor a = buyCD(x,y) \lor \]
    \[ bought(x,y,s) \land \neg (a = returnBook(x,y) \lor a = returnCD(x,y)) \]
  - Axioms for initial theory \( \mathcal{D}_{S_0} \):
    - Facts 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: for a regressable SC sentence $W$, decide whether $\mathcal{D} \models W$
- Executability problem: given a sequence of actions $A_1;\ldots;A_n$, decide whether $\mathcal{D} \models \text{Poss}(A_1,S_0) \land \text{Poss}(A_2,\text{do}(A_1,S_0)) \land \ldots \land \text{Poss}(A_n,\text{do}([A_1,\ldots,A_{n-1}],S_0))$
- Key reasoning mechanism – the regression operator $\mathcal{R}$ (Waldinger, 1977)
- Successor state axioms support regression in a natural way (Reiter, 2001):
  
  If $F(x_1,\ldots,x_n,\text{do}(a,s)) \equiv \Psi_F(x_1,\ldots,x_n,a,s)$, then
  
  $\mathcal{R}[F(t_1,\ldots,t_n,\text{do}(A,S))] = \mathcal{R}[\Psi_F(t_1,\ldots,t_n,A,S)]$.

- Important properties for regression:
  
  (1) $\mathcal{D} \models W \equiv \mathcal{R}[W]$, 
  (2) $\mathcal{D} \models W$ iff $\mathcal{D}_{S0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]$.

Advantage: compact representation of actions and their effects.

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

Solution: Consider $C^2$ - a fragment of the first-order logic with counting.
Description Logics vs. $C^2$

- **Description logics**
  - Foundation of OWL
  - A variety of logics
  - $\mathcal{ALCQIO} (\sqcap, \sqcup, \neg, |, id)$

- **$C^2$: a fragment of FOL**
  - At most two variables $x, y, =$
  - No function symbols
  - Add counting quantifiers $\exists^{\geq n}, \exists^{\leq n}$

- $\mathcal{ALCQIO} (\sqcap, \sqcup, \neg, |, id)$ vs. $C^2$
  - Concept names $\iff$ unary predicates
    - `instore` $\iff$ `instore(x)`
  - Role names $\iff$ binary predicates
    - `boughtBook` $\iff$ `boughtBook(x,y)`
  - E.g., $\exists^{\geq n} R. C \iff \exists^{\geq n} y. R(x,y) \land C(y)$
    - $\forall R. C \iff \forall y. R(x,y) \supset C(y)$
    - $\neg C \iff \neg C(x)$
    - $C \sqcap C \iff C(x) \land C(y)$

- **Advantages**
  - Many features in Semantic Web can be easily represented in $C^2$.
  - The reasoning in $C^2$ can also be translated into DLs.
  - May use current existing efficient DL reasoners for $C^2$ formulas.

$\mathcal{ALCQIO} (\sqcap, \sqcup, \neg, |, id) \iff C^2$, the translation algorithm is linear in the size of the given formula, both logics are decidable even under OWA.
The Decidable Situation Calculus $\mathcal{L}^{DL}_{SC}$

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

- **Sorts:**
  - Terms of sort *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 variables allowed in $\mathcal{L}^{DL}_{SC}$ theories

- **Fluents with either two or three arguments:**
  - (Dynamic) concepts $\text{instore}(x,s)$, ....
  - (Dynamic) roles $\text{boughtBook}(x,y,s)$, $\text{bought}(x,y,s)$, ...

- **Facts with either one or two arguments:**
  - (Static) concepts $\text{person}(x)$, $\text{client}(x)$, $\text{book}(y)$, $\text{cd}(y)$, ...
  - (Static) roles $\text{hasCreditCard}(x,y)$, ...

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

- **Precondition axioms:** The RHS is a $C^2$ formula if $s$ is suppressed
- **Successor 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** (terminology):
  - 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 a $C^2$ formula when the situation argument $s$ is suppressed
    - E.g., $valCust(x,s) \equiv person(x) \land (\exists y \geq 3) bought(x,y,s) \land book(y)$
    - $client(x) \equiv person(x) \land (\exists y) hasCreditCard(x,y)$
  - Reasoning: use lazy unfolding for dynamic axioms
- **RBox axioms** (role inclusions):
  - $R1 \supset R2$ for roles $R1$, $R2$  
    - E.g., $boughtBook(x,y,s) \supset bought(x,y,s)$, $boughtCD(x,y,s) \supset bought(x,y,s)$
  - 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 regressable in $L_{DC}^{SC}$
  -- All situation terms in $W$ have a syntactic form $do([A_1, \ldots, A_{n-1}], S_0)$
  -- 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., $\mathcal{R}[W1 \land W2] = \mathcal{R}[W1] \land \mathcal{R}[W2]$, …
  – Add $\mathcal{R}[\exists^\geq n \nu.W] = \exists^\geq n \nu.\mathcal{R}[W]$
  – 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 for actions (to get rid of action functions)
A Regression Example in $\mathcal{L}_{sc}^{DL}$

- Example: online shopping

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

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

- Suppose $W$ is a regressable formula of $L_{\text{SC}}^{\text{DL}}$ 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 \textit{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 queries about ground fluent $F(X,S)$ is co-NEXPTIME
- Executability problems and projection problems are \textit{decidable} in $L_{\text{SC}}^{\text{DL}}$
  - Whether a composite service is executable
  - Whether desirable/undesirable properties will be true/false after the execution
Classical Progression

- Regression is not practical when have to reason about properties after executing a very long sequence of actions
- Progression: compute the new theory given the current theory
- [Lin & Reiter 1997] 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 (we suppress the situation argument)
    
    $[ \forall x (x = B_1 \supset \neg ontable(x)) \lor \forall y (y \neq B_2 \supset ontable(y))] \land$
    
    $\forall z (z \neq B_3 \land z \neq B_4 \supset hold(z))$

- A local-effect BAT: every SSA axiom is local-effect, i.e.,
  
  $F(x, do(A, s)) \equiv x=B_1 \land p_1(s) \lor \ldots \lor x=B_m \land p_m(s) \lor$
  
  $F(x, s) \land \neg (x=C_1 \land q_1(s) \lor \ldots \lor x=C_n \land q_n(s))$

  where $s$ is the only variable (both free and quantified) in any $p_i$ and $q_j$. 
A Progression Algorithm & Properties

- We provide an algorithm for computing 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
  - For each fluent, add truth values for those objects where it will definitely become true (or false)
  - Update the remaining consistent information by removing knowledge about conflicting 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 an action language suitable for decidable reasoning about Web services
  – Our language facilitates compact representation and is quite expressive
  – Consider the knowledge base progression/update problem in $L^\text{DL}_{\text{SC}}$

• 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 theory in $\mathcal{E}$

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