The Two-Variable Situation Calculus

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

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

August 28, 2006
Outline

• Motivations
• Preliminaries
• Specification of the modified situation calculus for services
• Decidable reasoning about actions in this logic
• Discussions and future work
Shopping Online

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

Web Servers (E.g., Amazon)

Citations (buyers)

Arrangement

Shipping

Inventory
Motivations

• Usually suppliers (Web servers) could not get complete information (OWA)
• Need composition of atomic services to achieve the clients’ requests
• Integrating Semantic webs with Web services
• Representing the dynamics
  – What needs to be represented?
    atomic services (i.e., actions), dynamic environment (such as what books are available currently), the effect of service action
  – Expectations:
    • Represent actions for large/infinite domains (such as people, weight, time)
    • Be able to represent knowledge such as “there exist some ...”
For composite services and the environment,

– What do we care about? (reasoning)
  
  • *Executability Problem*:
    Whether the composite services are possible to be executed successfully?

  • *Projection Problem*:
    Whether certain properties/goals can be satisfied after the execution?

– Expectations:
  efficient reasoning (here, decidability)
The Situation Calculus

• A first-order logic language
  – Represent actions and effects in a natural way
  – Very compact

• Three sorts:
  – Actions: \textit{buyBook}(x,y), \textit{returnBook}(x,y), …
  – Situations: \textit{S}_0, \textit{do}(a,s), \textit{do}([a_1,...,a_n],s)
  – Objects: things other than actions and situations.
    E.g., places, names, numbers, etc.

• Fluents: system features whose truth values may vary.
  E.g., \textit{instore}(x,s), \textit{boughtBook}(x,y,s), \textit{bought}(x,y,s)…
Basic action theory $\mathcal{D}$

- A set of first-order axioms to model actions and effects
- 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
Reasoning about Actions

• E.g., \((\exists x)(\forall y)(\forall y') boughtBook(x, y, S) \land boughtBook(x, y', S) \supset y = y'\)

• 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
    \[\mathcal{R}[F(t_1, \ldots, t_n, do(a', S))] = \mathcal{R}[\Psi_F'(t_1, \ldots, t_n, a', S)].\]

• Important properties for regression
  – \(\mathcal{D} \models W \equiv \mathcal{R}[W]\)
  – \(\mathcal{D} \models W \iff \mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models \mathcal{R}[W]\)
Disadvantages of the Situation Calculus

Advantage: representing actions and effects very compactly.

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

1. Can we get rid of the disadvantage?
2. Can we specify the Semantic Web features in a natural way?

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

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

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

- $\mathcal{ALCQIO}(\sqcap, \sqcup, \lnot, |, id)$ vs. $\mathcal{C}^2$
  - Concept names $\Leftrightarrow$ unary predicates
    - $\text{instore} \Leftrightarrow \text{instore}(x)$
  - Role names $\Leftrightarrow$ binary predicates
    - $\text{boughtBook} \Leftrightarrow \text{boughtBook}(x,y)$
  - E.g.,
    - $\exists^\geq n R.C \Leftrightarrow \exists^\geq n y. R(x,y) \land C(y), \quad \forall R.C \Leftrightarrow \forall y. R(x,y) \supset C(y)$
    - $\lnot C \Leftrightarrow \lnot C(x), \quad C1 \sqcap C2 \Leftrightarrow C1(x) \land C2(x)$
Decidability of DLs and C²

- [Borgida1996] \( ALCQIO(\sqcap, \sqcup, \neg, |, id) \) plus cross-product \( \Leftrightarrow C² \).
- We showed that: \( C1 \times C2 = (R \sqcap \neg R)_{C2} \sqcap ((R \sqcap \neg R)_{C1})^{-} \).
- [Grädel et al., Pacholski et al. 1997] \( C² \) is decidable even under OWA.

\( ALCQIO(\sqcap, \sqcup, \neg, |, id) \Leftrightarrow C² \), the translation algorithm is linear in the size of the given formula.

\( ALCQIO(\sqcap, \sqcup, \neg, |, id) \) is decidable even under OWA.

- Other 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.
The Decidable Situation Calculus $\mathcal{L}_{DL}^{SC}$

- **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 $\text{instore}(x, s)$, ….
  - (Dynamic) roles $\text{boughtBook}(x, y, s)$, $\text{boughtCD}(x, y, s)$, $\text{bought}(x, y, s)$, …

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

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

• Precondition axioms:
  – The RHS is a $C^2$ formula if the situation argument $s$ 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: (with OWA)
  – Each axiom is a $C^2$ formula if $S_0$ is suppressed

Purpose: to ensure the formula resulting from regression is a $C^2$ formula (regardless $S_0$).
Extensions of the Basic Action Theory

• Allowing specify certain features similar to DLs

• 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 \( D_{S_0} \))
  – The RHS is \( C^2 \) when the situation argument \( s \) is suppressed
    E.g., \( \text{valCust}(x,s) \equiv \text{person}(x) \land (\exists^{\geq 3} y) \text{bought}(x,y,s) \)
    \( \text{client}(x) \equiv \text{person}(x) \land (\exists y) \text{hasCreditCard}(x,y) \)
  – Reasoning: use lazy unfolding for Dynamic axioms

• RBox axioms:
  – For taxonomic reasoning purposes
  – \( R1 \supset R2 \) 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., \( \mathcal{D} \models (\forall x,y,s).R1(x,y)[s] \supset R2(x,y)[s] \)
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 introduce new variables to deal with quantifiers

• Formula $W$ that is regressable in $\mathcal{L}_{SC}$
  – 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 (examples on the next slide)
  – When $W$ is a defined dynamic concept, use TBox axioms (lazy unfolding)
A Regression Example in $\mathcal{L}^{\text{DL}}_{\text{SC}}$

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

$\mathcal{R}[(\exists x).\text{valCust}(x, do([A1,A2,A3], S_0))]$

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

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

$= \ldots$ \text{(recursively do regression using the successor state axioms)}

$= (\exists x).\text{person}(x) \land (\exists \geq 3 y) [(x=\text{Tom} \land y = \text{TheFirm}) \lor$

$(x=\text{Tom} \land y = \text{HarryPotter}) \lor$

$(x=\text{Tom} \land y = \text{HarryPotter}) \lor$

$\text{bought}(x,y,S_0) ]$
Important Properties

Suppose $W$ is a regressable formula of $L_{SC}^{DL}$ with the basic action theory $D$

- 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
- $D \models W \equiv \mathcal{R}[W]$
- $D \models W$ iff $D_{S0} \cup D_{una} \models \mathcal{R}[W]$

• The problem whether is $D \models W$ is *decidable*
  
  - $D_{S0} \cup D_{una} \models \mathcal{R}[W]$ is a decidable reasoning in $C^2$ when $S_0$ is suppressed everywhere

• The executability problems and projection problems are *decidable* in $L_{SC}^{DL}$
  
  - Whether a composite service is executable
  - Whether desired/undesired properties will be true/false after the execution
Discussions and Future Work

• Conclusions
  – Formalize a decidable language suitable for Web services
  – Have compact powerful expression power

• Other related researches
  – [McIlraith and 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 the situation calculus)
  – [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

• Possible future work
  – Implementations
  – Consider the knowledge base progression/update problem in $L_{DL}^{SC}$
  – Etc.