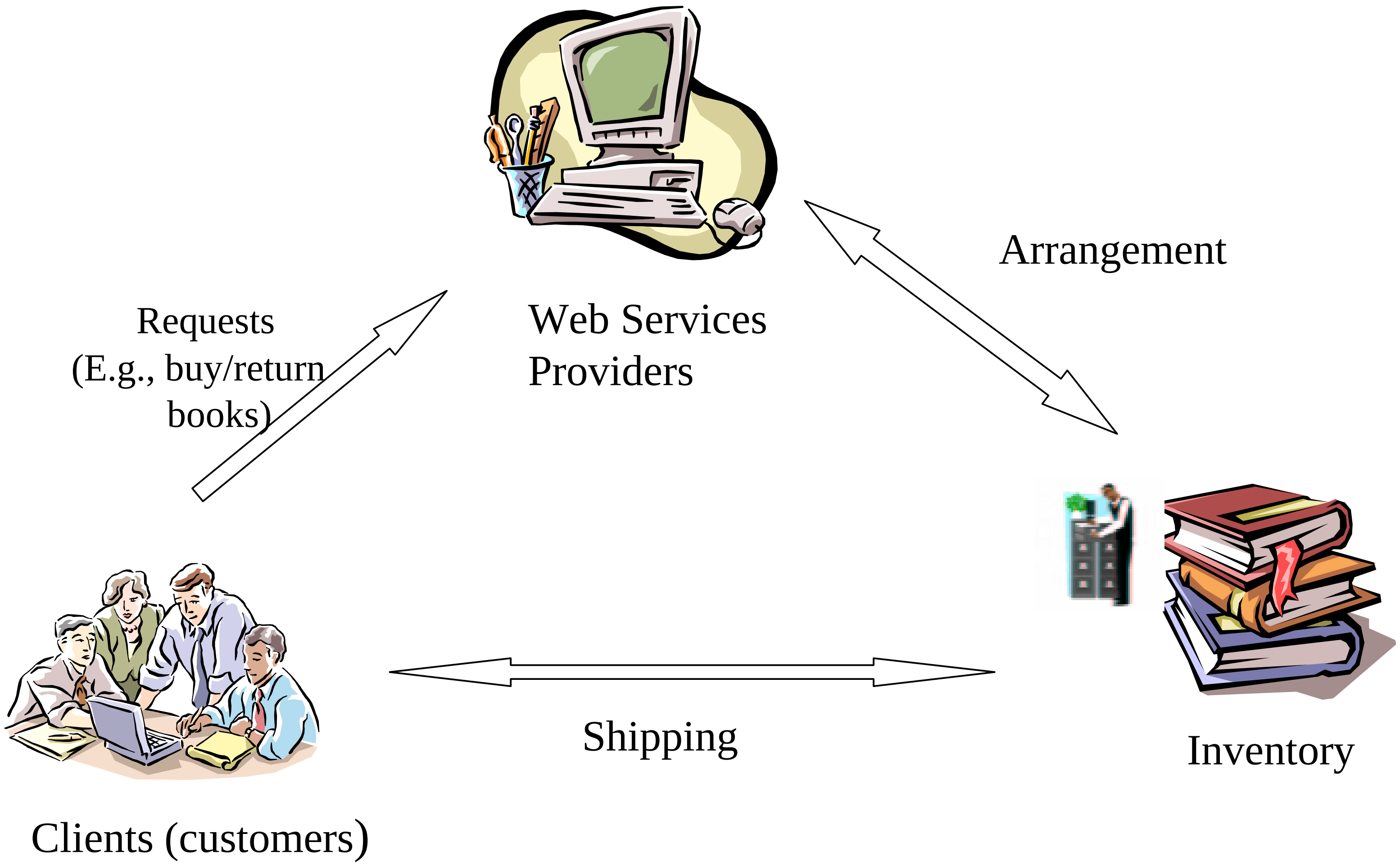
Decidable Reasoning in a Modified Situation Calculus

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



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, $do(a,s)$, $do([a_1,\ldots,a_n],s)$
  - 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 $\mathcal{D}_{ap}$:

    $Poss(buyBook(x,y),s) \equiv client(x) \wedge book(y) \wedge instore(y,s)$
  - Successor state axioms $\mathcal{D}_{ss}$:

    $bought(x,y,do(a,s)) \equiv a = buyBook(x,y) \vee a = buyCD(x,y) \vee$
    $bought(x,y,s) \wedge \neg (a = returnBook(x,y) \vee a = returnCD(x,y))$
  - Axioms for initial theory $\mathcal{D}_{S0}$:
    - 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
  $\mathcal{D} \models Poss(A_1,S_0) \wedge Poss(A_2,do(A_1,S_0)) \wedge \ldots \wedge Poss(A_n, do([A_1,\ldots,A_{n-1}], S_0))$
- Key reasoning mechanism – the regression operator $\mathcal{R}$ (Waldinger, 1977)
- Successor state axioms support regression in a natural way (Reiter, 2001):
  If $F(x_1,\ldots,x_n, do(a,s)) \equiv \Psi_F(x_1,\ldots,x_n,a,s)$, then
  $\mathcal{R}[F(t_1,\ldots,t_n, do(A,S))] = \mathcal{R}[\Psi_F(t_1,\ldots,t_n,A,S)]$.

$$W_0(S_0) \xleftarrow{\mathcal{R}} \cdots \xleftarrow{\mathcal{R}} W'(do[a_1,\ldots,a_{n-1}], S_0) \xleftarrow{\mathcal{R}} W(do[a_1,\ldots,a_n], S_0)$$

- 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 ⇔ unary predicates *instore* ------- *instore*(*x*)
  - Role names ⇔ binary predicates *boughtBook* ------ *boughtBook*(*x*,*y*)
  - E.g., $\exists^{\geq n} R.C \Leftrightarrow \exists^{\geq n} y.R(x,y) \wedge C(y)$, $\forall R.C \Leftrightarrow \forall y.R(x,y) \supset C(y)$

    $\neg C \Leftrightarrow \neg C(x)$, $C1 \sqcap C2 \Leftrightarrow C1(x) \wedge C2(x)$
- 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*) ⇔ 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 $\mathcal{L}_{SC}^{DL}$

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 $\mathcal{L}_{SC}^{DL}$ theories
- 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*(y), *cd*(y),…
  - (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
- 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 – *<u>defined</u>* dynamic concept)
  - Static ones: $C(x) \equiv \Phi_C(x)$ (provided in the $\mathcal{D}_{S0}$)
  - The RHS is a C^2 formula when the situation argument s is suppressed

    E.g., $valCust(x,s) \equiv person(x) \wedge (\exists^{\geq 3} y)\ bought(x,y,s) \wedge book(y)$

    $client(x) \equiv person(x) \wedge (\exists\ y)\ hasCreditCard(x,y)$
  - 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 $\mathcal{D}_{SS}$, 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 introduces new variables to deal with quantifiers
- Formula *W* that is regressable in $\mathcal{L}_{SC}^{DC}$
  - -- 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., $\mathcal{R}[W1 \wedge W2] = \mathcal{R}[W1] \wedge \mathcal{R}[W2], \ldots$
  - ***Add*** $\mathcal{R}[\exists^{\geq n} v.W] = \exists^{\geq n} v.\mathcal{R}[W]$
  - ***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 for actions (to get rid of action functions)

A Regression Example in $\mathcal{L}_{SC}^{DL}$

- Example: online shopping

$A1 = buyCD(\text{Tom, BackStreetBoys})$

$A2 = buyBook(\text{Tom, HarryPotter})$

$A3 = buyBook(\text{Tom, TheFirm})$

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

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

(lazy unfolding)

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

= … (recursively do regression using the successor state axioms)

$= (\exists x).\ person(x) \wedge (\exists^{\geq 3} y)\ [(x\text{=Tom} \wedge y = \text{TheFirm}) \vee$

$(x\text{=Tom} \wedge y = \text{HarryPotter}) \vee$

$(x\text{=Tom} \wedge y = \text{BackStreetBoys}) \vee$

$bought(x,y,S_0)\ ]$

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* $\mathcal{D}_{S0} \models \mathcal{R}[W]$
- The problem whether is $\mathcal{D} \models W$ is ***decidable***
  - $\mathcal{D}_{S0} \models \mathcal{R}[W]$ 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 ***decidable*** in $\mathcal{L}_{SC}^{DL}$
  - 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 $\mathcal{D}_a$ is the classical progression of the initial KB $\mathcal{D}_0$ (wrt BAT $\mathcal{D}$) after performing a ground action a in the situation S_0 iff
  - $\mathcal{D}_a$ is uniform in $do(a, S_0)$;
  - $\mathcal{D} \models \mathcal{D}_a$;
  - for every model M_a of $(\mathcal{D} \setminus \mathcal{D}_0) \cup \mathcal{D}_a$, there is a model M of $\mathcal{D}$ 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)

    $[\forall x(x = B_1 \supset \neg\, ontable(x)) \vee \forall y(y \neq B_2 \supset ontable(y))] \wedge$

    $\forall z\, (z \neq B_3 \wedge z \neq B_4 \supset hold(z))$
- A local-effect BAT: every SSA axiom is local-effect, i.e.,

$$F(x, do(A, s)) \equiv x{=}B_1 \wedge p_1(s) \vee \ldots \vee x{=}B_m \wedge p_m(s) \vee$$

$$F(x,s) \wedge \neg\, (\, x{=}C_1 \wedge q_1(s) \vee \ldots \vee x{=}C_n \wedge q_n(s)\,)$$

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 $\mathcal{L}_{SC}^{DL}$
- 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 → constants, fluents → 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
    - [Claβen & Lakemeyer 2007] proposes a progression of an ADL theory in $\mathcal{ES}$
- 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