1

The Typed Situation Calculus

YILAN GU AND MIKHAIL SOUTCHANSKI

ABSTRACT. We propose a theory for reasoning about actions based on order-sorted predicate logic where one can consider an elaborate taxonomy of objects. We are interested in the projection problem: whether a statement is true after executing a sequence of actions. To solve it we design a regression operator takes advantage of well-sorted unification between terms. We show that answering projection queries in our logical theories is sound and complete wrt answering similar queries in Reiter's basic action theories. This proves correctness of our approach. Moreover, we demonstrate that our regression operator based on order-sorted logic can provide significant computational advantages in comparison to Reiter's regression operator.

1 Preface

Hector Levesque's research about structured representation of knowledge and about computationally tractable reasoning over this knowledge makes long-standing and important contributions to AI. He contributed to the idea that assertional knowledge (ABox) and terminological knowledge (TBox) should be represented separately, invented the terms ABox and TBox, and together with Ron Brachman founded a research area that later became known as Description Logics [Brachman and Levesque 1982; Brachman, Levesque, and Fikes 1983]. He summarized some of his ideas in a well-known lecture presented upon receipt of the 1985 Computers and Thought Award [Levesque 1986]. A few years later, he also made significant contributions to the situation calculus and to Cognitive Robotics [Levesque 1994; Levesque, Reiter, Lespérance, Lin, and Scherl 1997; Bacchus, Halpern, and Levesque 1999; Scherl and Levesque 2003; Levesque and Lakemeyer 2008; Lakemeyer and Levesque 2011]. As Hector's graduate students we had an opportunity to learn about his research first hand. Not surprisingly, our own research has been influenced by his ideas. The results of one of our research exercises are presented below. Our main departure from more traditional themes in Hector's work is in exploring how many-sorted (or more precisely, order-sorted) representations can be useful to provide an internal structure for objects, while Hector himself explored different forms for representing knowledge. Nevertheless, our research benefited from discussions with him, and from his comments.

2 Introduction

Starting from 1970s, many-sorted reasoning and taxonomies gained in popularity in deductive databases and automated reasoning. In particular, [McSkimin 1976] subdivides a domain into semantic categories, uses them to build a semantic category graph and argues that this semantic world would provide computational advantages in a query answering system. These ideas have been implemented in MRPPS (the Maryland Refutation Proof Procedure System) described in in [McSkimin and Minker 1977]. In deductive databases, [Reiter 1977a; Reiter 1977b] also use boolean combination of monadic predicates to express taxonomies of types (each simple type is represented by a monadic predicate). Reiter develops a typed unification algorithm and argues that his approach is more suitable for real world databases than the approach in deductive question-answering research that deals with unrestricted first-order databases.

In his influential paper [Hayes 1971] titled "A Logic of Actions", P. Hayes proposed an outline of a logical theory for reasoning about actions based on many-sorted logic with equality. His paper inspired subsequent work on many-sorted logics in AI. In particular, A. Cohn [Cohn 1987; Cohn 1989] developed expressive many-sorted logic and reviewed all previous work in this area. Reasoning about action and change based on the situation calculus (SC) has been extensively developed in [Reiter 2001]. However, it considers a logical language with sorts for actions, situations and just one catch-all sort $Object$ for the rest that remains unelaborated. Surprisingly, even if the idea proposed by Hayes seems straightforward, there is still no formal study of logical and computational properties of a version of the SC with many related sorts for objects in the domain. Perhaps, this is because mathematical proofs of these properties are not straightforward although the intuition is. There are other action formalisms that permit sorted objects; such representations have also been used in planning formalisms, such as the standard PDDL language. However, to the best of our knowledge, none of the previous work investigated logical foundations and semantics of reasoning about actions over typed domains. We undertake this study and demonstrate that reasoning about actions with elaborated sorts has significant computational advantages in comparison to reasoning without them. In contrast to an approach to many-sorted reasoning [Schmidt 1938; Wang 1952; Herbrand 1971] where variables of different sorts range over unrelated universes, we consider a case when sorts are related to each other, so that one can construct an elaborated taxonomy. This is often convenient for representation of common-sense knowledge about a domain.

Generally speaking, we are usually interested in a comprehensive taxonomic structure for sorts, where sorts may inherit from each other and may have non-empty intersections. Note that even if both many-sorted logic and order-sorted logic can be translated to unsorted logic, using sorted ones can bring about significant computational advantages, for example in deduction, comparing to unsorted logic. This was a primary driving force for [Walther 1987; Cohn 1987]. Hence, we consider formulating the SC in an order-sorted logic to describe taxonomic information about objects. Based on the newly formulated language, we consider solving the projection problem via regression. We show that regression in the order-sorted SC can benefit from well-sorted unification. One can gain computational efficiency by terminating regression steps earlier when objects of incommensurable sorts are involved. However, before we can address issues of computational advantages, we have to investigae formally the relations between the proposed typed version and the situation calculus developed in [Reiter 2001], since the latter became de-facto standard. It turns out that proving soundness and completness of our new version with respect to Reiter's formulation is non-trivial matter.

3 Background

In general, order-sorted logic (OSL) [Oberschelp 1962; Walther 1987; Oberschelp 1990; Schmidt-Schauβ 1989; Bierle, Hedtstück, Pletat, Schmitt, and Siekmann 1992; Weidenbach 1996] restricts the domain of variables to subsets of the universe (i.e., *sorts*). Notation $x : Q$ means that variable x is of sort Q and $\mathbf{V}_Q$ is the set of variables of sort Q. For any

n, sort cross-product $Q_1 \times \cdots \times Q_n$ is abbreviated as $\vec{Q}_{1..n}$; term vector $t_1, \ldots, t_n$ is abbreviated as $\vec{t}_{1..n}$; variable vector $x_1, \ldots, x_n$ is abbreviated as $\vec{x}_{1..n}$; and, variable declaration sequence $x_1 \!:\! Q_1, \ldots, x_n \!:\! Q_n$ is abbreviated as $\vec{x}_{1..n} \!:\! \vec{Q}_{1..n}$.

A theory in OSL includes a set of declarations (called *sort theory*) to describe the hierarchical relationships among sorts and the restrictions on ranges of the arguments of predicates and functions. In particular, a sort theory $\mathbb{T}$ is a set of *term declarations* of the form $t \!:\! Q$ representing that term t is of sort Q, *subsort declarations* of the form $Q_1 \leq Q_2$ representing that sort Q_1 is a (direct) subsort of sort Q_2 (i.e., every object of sort Q_1 is also of sort Q_2), and *predicate declarations* of the form $P \!:\! \vec{Q}_{1..n}$ representing that the i-th argument of the n-ary predicate P is of sort Q_i for $i = 1..n$. In particular, when a logic has more than one sort symbol and there are no subsort declarations (any two sorts are disjoint), it is called *many-sorted* logic. A *function declaration* is a special term declaration where term t is a function with distinct variables as arguments: for each n-ary function f, the abbreviation of its function declaration is of the form $f : \vec{Q}_{1..n} \rightarrow Q$, where Q_i is the sort of the i-th argument of f and Q is the sort of the value of f. $c : Q$ is a special function declaration, representing that constant c is of sort Q. Arguments of equality "=" can be of any sort. Below, we consider a *finite simple* sort theory only, in which there are finitely many sorts and declarations, the term declarations are all function declarations, and for each function there is one and only one declaration (i.e., no polymorphism is allowed).

For any sort theory $\mathbb{T}$, subsort relation $\leq_{\mathbb{T}}$ is a partial ordering defined by the reflexive and transitive closure of the subsort declarations. Then, following the standard terminology of lattice theory, if each pair of sort symbols in $\mathbb{T}$ has greatest lower bound (g.l.b.), we say that *the sort hierarchy of* $\mathbb{T}$ *is a meet semi-lattice* [Walther 1987]. Moreover, a *well-sorted term* (wrt $\mathbb{T}$) is either a sorted variable, or a constant declared in $\mathbb{T}$, or a functional term $f(\vec{t}_{1..n})$, in which each t_i is well-sorted and the sort of t_i is a subsort of Q_i, given that $f : \vec{Q}_{1..n} \rightarrow Q$ is in $\mathbb{T}$. A *well-sorted atom* (wrt $\mathbb{T}$) is an atom $P(\vec{t}_{1..n})$ (can be $t_1 = t_2$), where each t_i is a well-sorted term of sort Q'_i, and $Q'_i \leq_{\mathbb{T}} Q_i$, given that $P : \vec{Q}_{1..n}$ is in $\mathbb{T}$. A *well-sorted formula* (wrt $\mathbb{T}$) is a formula in which all terms (including variables) and atoms are well-sorted. Any term or formula that is not well-sorted is called *ill-sorted*. A formal definition of *well-sorted formulas* is given in [Bierle, Hedtstück, Pletat, Schmitt, and Siekmann 1992]. A *well-sorted substitution* (wrt $\mathbb{T}$) is a substitution ρ s.t. for any variable $x : Q$, ρx (the result of applying ρ to x) is a well-sorted term and its sort is a (non-empty) subsort of Q. Given any set $E = \{\langle t_{1,1}, t_{1,2}\rangle, \ldots, \langle t_{n,1}, t_{n,2}\rangle\}$, where each $t_{i,j}$ ($i = 1..n, j = 1..2$) is a well-sorted term, a *well-sorted most general unifier* (well-sorted mgu) of E is a well-sorted substitution that is an mgu of E. It is important that in comparison to an mgu in unsorted logic (i.e., predicate logic without sorts), an mgu in OSL can include new weakened variables of sorts which are subsorts of the sorts of unified terms. For example, assume that $E = \{\langle x, y\rangle\}$, $x \in \mathbf{V}_{Q_1}$, $y \in \mathbf{V}_{Q_2}$ and the g.l.b. of $\{Q_1, Q_2\}$ is a non-empty sort Q_3. Then, $\mu = [x/z, y/z]$ (x is substituted by z, y is substituted by z) for some new variable $z \in \mathbf{V}_{Q_3}$ is a well-sorted mgu of E. A well-sorted mgu neither always exists nor it is unique. However, it is proved that the well-sorted mgu of unifiable sorted terms is unique up to variable renaming when the sort hierarchy of $\mathbb{T}$ is a meet semi-lattice [Walther 1987].

The semantics of OSL is defined similar to unsorted logic. Note that the definition of interpretations for well-sorted terms and formulas is similar to unsorted logic, but the semantics is not defined for ill-sorted terms and formulas. For any well-sorted formula ϕ, a $\mathbb{T}$-interpretation $\mathbb{I} = \langle \mathcal{M}, I\rangle$ is a tuple with a structure $\mathcal{M}$ and an assignment I from the set of free variables to the universe of $\mathcal{M}$, s.t. it satisfies the following conditions: (1) For each sort Q, $Q^{\mathbb{I}}$ is a subset of the whole universe $\mathbf{U}$. In particular, $\top^{\mathbb{I}} = \mathbf{U}$, $\bot^{\mathbb{I}} = \emptyset$, and $Q_1^{\mathbb{I}} \subseteq Q_2^{\mathbb{I}}$ for any $Q_1 \leq_{\mathbb{T}} Q_2$. (2) For any predicate declaration $P : \vec{Q}_{1..n}$, $P^{\mathbb{I}} \subseteq Q_1^{\mathbb{I}} \times \cdots \times Q_n^{\mathbb{I}}$ is a relation in $\mathcal{M}$. (3) For any function declaration $f : \vec{Q}_{1..n} \to Q$, $f^{\mathbb{I}} : Q_1^{\mathbb{I}} \times \cdots \times Q_n^{\mathbb{I}} \to Q^{\mathbb{I}}$ is a function in $\mathcal{M}$. (4) $x^{\mathbb{I}} = I(x)$ is in $Q^{\mathbb{I}}$ for any variable $x \in \mathbf{V}_Q$, $c^{\mathbb{I}} \in Q^{\mathbb{I}}$ for any constant declaration $c : Q$, and $(f(\vec{t}_{1..n}))^{\mathbb{I}} \stackrel{def}{=} f^{\mathbb{I}}(t_1^{\mathbb{I}}, \ldots, t_n^{\mathbb{I}})$ for any well-sorted term $f(\vec{t}_{1..n})$. $\mathbb{I}$ is not defined for ill-sorted terms and formulas. (5) If $\mathbb{T}$ includes a declaration for equality symbol "=", then $=^{\mathbb{I}}$ must be defined as a set $\{(d, d) \mid d \in \mathbf{U}\}$. For any sort theory $\mathbb{T}$ and a well-sorted formula ϕ, a structure $\mathcal{M}$ is a $\mathbb{T}$-*model* of ϕ, written as $\mathcal{M} \models^{\mathrm{os}}_{\mathbb{T}} \phi$ iff for every $\mathbb{T}$-interpretation $\mathbb{I} = \langle \mathcal{M}, I \rangle$, $\mathbb{I}$ satisfies ϕ. In particular, when ϕ is a sentence, this does not depend on any variable assignment and $\mathbb{I} = \mathcal{M}$. Moreover, the notion a $\mathbb{T}$-interpretation $\mathbb{I} = \langle \mathcal{M}, I \rangle$ satisfies ϕ, written as $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi$, is defined by structural induction on formulas as follows. (a) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} P(\vec{t}_{1..n})$ iff $(t_1^{\mathbb{I}}, \ldots, t_n^{\mathbb{I}}) \in P^{\mathbb{I}}$. (b) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \neg\phi$ iff $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi$ does not hold. (c) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_1 \wedge \phi_2$ iff $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_1$ and $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_2$. (d) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_1 \vee \phi_2$ iff $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_1$ or $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_2$. (e) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi_1 \supset \phi_2$ iff $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \neg\phi_1 \vee \phi_2$. (f) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \forall x : Q.\phi$ iff for every $d \in Q^{\mathbb{I}}$, $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi[x/d]$, where $\phi[x/o]$ represent the formula obtained by substituting x with o. (g) $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \exists x : Q.\phi$ iff there is some $d \in Q^{\mathbb{I}}$ s.t. $\mathbb{I} \models^{\mathrm{os}}_{\mathbb{T}} \phi[x/d]$. Given a sort theory $\mathbb{T}$ as the background, a theory Φ including well-sorted sentences only satisfies a well-sorted sentence ϕ, written as $\Phi \models^{\mathrm{os}}_{\mathbb{T}} \phi$, iff every model of Φ is a model of ϕ.

Due to the space limitations, we do not introduce the SC. Details can be found in [Reiter 2001] and we refer to this language as Reiter's SC $\mathcal{L}_{sc}$ below. Also note that we use $\models^{\mathrm{os}}_{\mathbb{T}}$ to represent the logical entailment wrt a sort theory $\mathbb{T}$ in order-sorted logic, $\models^{\mathrm{ms}}$ to represent the logical entailment in Reiter's SC (a many-sorted logic), and $\models^{\mathrm{fo}}$ to represent the logical entailment in unsorted predicate logic.

4 An Order-Sorted Situation Calculus

Here, we consider a modified SC based on order-sorted logic, called *order-sorted SC* and denoted as $\mathcal{L}^{OS}$ below. $\mathcal{L}^{OS}$ includes a set of sorts **Sort** = **Sort**$_{obj} \cup \{\top, \bot, Act, Sit\}$, where $\top$ represents the whole universe, $\bot$ is the empty sort, Act is the sort for all actions, Sit is the sort for all situations, and **Sort**$_{obj}$ is a set of sub-sorts of $Object$ including sort $Object$ itself. We assume that for every sort (except $\bot$) there is at least one ground term (constant) of this sort to avoid the problem with "empty sorts" [Goguen and Meseguer 1987]. Moreover, the number of individual variable symbols of each sort in **Sort** is infinitely countable. For the sake of simplicity, we do not consider functional fluents here.

Here, we will consider dynamical systems that can be described using *order-sorted basic action theories* (order-sorted BATs). An order-sorted BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$ includes the following two parts of theories.

• $\mathbb{T}$ is a sort theory based on a finite set of sorts $\mathbf{Q}_{\mathcal{D}}$ s.t. $\mathbf{Q}_{\mathcal{D}} \subseteq$ **Sort** and $\{\bot, \top, Object, Act, Sit\} \subseteq \mathbf{Q}_{\mathcal{D}}$. Moreover, the sort theory includes the following declarations for finitely many predicates and functions:

1. Subsort declarations of the form $Q_1 \leq Q_2$ for $Q_1, Q_2 \in \mathbf{Q}_{\mathcal{D}} - \{\top, Act, Sit\}$, and subsort declarations: $Object \leq \top$, $Act \leq \top$, $Sit \leq \top$, $\bot \leq Act$, $\bot \leq Sit$. Here, we only consider those sort theories whose sort hierarchies are meet semi-lattices.
2. One and only one predicate declaration of the form $F : \vec{Q}_{1..n}$ for each n-ary relational fluent F in the system, where $Q_i \leq_{\mathbb{T}} Object$ and $Q_i \neq \bot$ for $i = 1..(n-1)$, and Q_n is Sit.

3. One and only one predicate declaration for the special predicate $Poss$, that is, $Poss: Act \times Sit$.

4. One and only one predicate declaration of the form $P : \vec{Q}_{1..n}$ for each n-ary situation independent predicate P in the system, where $Q_i \leq_{\mathbb{T}} Object$ and $Q_i \neq \bot$ for $i = 1..n$.

5. A special declaration for equality symbol $= :\ \top \times \top$.

6. One and only one function declaration of the form $A : \vec{Q}_{1..n} \rightarrow Act$ for each n-ary action function A in the system, where $Q_i \leq_{\mathbb{T}} Object$ and $Q_i \neq \bot$ for $i = 1..n$. Note that, when $n = 0$, the declaration is $A : Act$ for constant action function A.

7. One and only one function declaration of the form $f : \vec{Q}_{1..n} \rightarrow Q_{n+1}$ for each n-ary ($n \geq 0$) situation independent function f (other than action functions), where each $Q_i \leq_{\mathbb{T}} Object$ and $Q_i \neq \bot$ for each $i = 1..(n+1)$. When $n = 0$, the function declaration is of the form $c : Q$ for constant c of sort Q.

8. One and only one function declaration $do : Act \times Sit \rightarrow Sit$, and $S_0 : Sit$ for the initial situation S_0.

• $\mathbf{D}$ is a set of axioms represented using well-sorted sentences wrt $\mathbb{T}$, which includes the following subsets of axioms.

1. Foundational axioms Σ for situations, which are the same as those in [Reiter 2001].

2. A set $\mathcal{D}_{una}$ of unique name axioms for actions: for any two distinct action function symbols A and B with declarations $A : \vec{Q}_{1..n} \rightarrow Act$ and $B : \vec{Q}'_{1..m} \rightarrow Act$, we have
$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, \vec{y}_{1..m} : \vec{Q}'_{1..m}).\ A(\vec{x}_{1..n}) \neq B(\vec{y}_{1..m}).$$
Moreover, for each action function symbol A, we have
$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, \vec{y}_{1..n} : \vec{Q}_{1..n}).\ A(\vec{x}_{1..n}) = A(\vec{y}_{1..n}) \supset \bigwedge_{i=1}^{n} x_i = y_i.$$

3. The initial theory $\mathcal{D}_{S_0}$, which includes well-sorted (first-order) sentences that are uniform in S_0.

4. A set $\mathcal{D}_{ap}$ of precondition axioms for actions represented using well-sorted formulas: for each action symbol A, whose sort declaration is $A : \vec{Q}_{1..n} \rightarrow Act$, its precondition axiom is
$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, s : Sit).Poss(A(\vec{x}_{1..n}), s) \equiv \Pi_A(\vec{x}_{1..n}, s), \quad (1)$$
where $\Pi_A(\vec{x}_{1..n}, s)$ is a well-sorted formula uniform in s, whose free variables are at most among $\vec{x}_{1..n}$ and s.

5. A set $\mathcal{D}_{ss}$ of successor state axioms (SSAs) for fluents represented using well-sorted formulas: for each fluent F with declaration $F : \vec{Q}_{1..n} \times Sit$, its SSA is of the form
$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, a : Act, s : Sit).F(\vec{x}_{1..n}, do(a, s)) \equiv \phi_F(\vec{x}_{1..n}, a, s), \quad (2)$$
where $\phi_F(\vec{x}_{1..n}, a, s)$ is a well-sorted formula uniform in s, whose free variables are at most among $\vec{x}_{1..n}$ and a, s.

Here is a simple example of an order-sorted BAT.

EXAMPLE 1 (Transport Logistics) We present an order-sorted BAT $\mathcal{D}$ of a simplified example of logistics. $\mathbb{T}$ includes the following subsort declarations:

$MovObj \le Object$, $\perp \le City$, $\perp \le Box$, $\perp \le Truck$,

$Truck \le MovObj$, $City \le Object$, $Box \le MovObj$,

where $MovObj$ is the sort of movable objects, and other sorts are self-explanatory. The predicate declarations are $InCity : MovObj \times City \times Sit$, $On : Box \times Truck \times Sit$ for the fluents $InCity(o, l, s)$ and $On(o, t, s)$. An example of function declaration can be $twinCity : City \rightarrow City$. The function declarations for actions $load(b, t)$, $unload(b, t)$ and $drive(t, c_1, c_2)$ are obvious. For instance, $drive : Truck \times City \times City \rightarrow Act$. Besides $S_0 : Sit$, the constant declarations may include:

$B_1 : Box$, $B_2 : Box$, $T_1 : Truck$,

$T_2 : Truck$, $Toronto : City$, $Boston : City$.

Axioms in $\mathcal{D}_{S_0}$ can be:

$\exists x : Box. InCity(x, Boston, S_0)$,

$(\forall x : Box, t : Truck). \neg On(x, t, S_0)$,

$InCity(T_1, Boston, S_0) \vee InCity(T_2, Boston, S_0)$.

As an example, the precondition axiom for $load$ is:

$$(\forall x : Box, t : Truck, s : Sit). Poss(load(x, t), s) \equiv \\ \neg On(x, t, s) \wedge \exists y : City. InCity(x, y, s) \wedge InCity(t, y, s),$$

and the preconditions for $unload$ and $drive$ are obvious. As an example, the SSA of fluent $InCity$ is:

$$(\forall d : MovObj, c : City, a : Act, s : Sit). InCity(d, c, do(a, s)) \equiv \\ (\exists t : Truck, c_1 : City). a = drive(t, c_1, c) \wedge \\ (d = t \vee \exists b : Box. b = d \wedge On(b, t, s))) \vee \\ InCity(d, c, s) \wedge \neg(\exists t : Truck, c_1 : City. a = drive(t, c, c_1) \wedge \\ (d = t \vee \exists b : Box. b = d \wedge On(b, t, s))),$$

and the SSA of fluent On is obvious.

5 Order-Sorted Regression and Reasoning

We now consider the central reasoning mechanism in the order-sorted SC. The definition of a regressable formula of $\mathcal{L}^{OS}$ is the same as the definition of a regressable formula of $\mathcal{L}_{sc}$ except that instead of being stated for a formula in $\mathcal{L}_{sc}$, it is formulated for a well-sorted formula in $\mathcal{L}^{OS}$.

A formula W of $\mathcal{L}^{OS}$ is *regressable* (wrt an order-sorted BAT $\mathcal{D}$) iff (1) W is a well-sorted first-order formula wrt $\mathbb{T}$; (2) every term of sort Sit in W starts from S_0 and has the syntactic form $do([\alpha_1, \cdots, \alpha_n], S_0)$, where each α_i is of sort Act; (3) for every atom of the form $Poss(\alpha, \sigma)$ in W, α has the syntactic form $A(\vec{t}_{1..n})$ for some n-ary action function symbol A; and (4) W does not quantify over situations, and does not mention the relation symbols "$\sqsubset$" or "$=$" between terms of sort Sit. A *query* is a regressable sentence.

EXAMPLE 2 Consider the BAT $\mathcal{D}$ from Example 1. Let W be

$$\exists d : Box. d = Boston \wedge On(d, T_1, do(load(B_1, T_1), S_0)).$$

W is a (well-sorted) regressable sentence (wrt $\mathcal{D}$); while

$$On(Boston, T_1, do(load(B_1, T_1), S_0))$$

is ill-sorted and therefore is not regressable.

The regression operator $\mathcal{R}^{os}$ in $\mathcal{L}^{OS}$ is defined recursively similar to the regression operator in [Reiter 2001]. Moreover, we take advantages of the sort theory during regression: when there is no well-sorted mgu for equalities between terms that occur in a conjunctive sub-formula of a query, this sub-formula is logically equivalent to false and it should not be regressed any further. We will see that this key idea helps eliminate useless sub-trees of a regression tree.

DEFINITION 3 Consider a regressable formula W in $\mathcal{L}^{OS}$ with respect to a background order-sorted BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$. The *regression* of W, $\mathcal{R}^{os}[W]$, is recursively defined as follows. In what follows, $\vec{t}$ and $\vec{\tau}$ are tuples of terms, α and α' are terms of sort Act, σ and σ' are terms of sort Sit, and W is a regressable formula of $\mathcal{L}^{OS}$.

1. If W is a non-atomic formula and is of the form $\neg W_1$, $W_1 \vee W_2$, $(\exists v : Q).W_1$ or $(\forall v : Q).W_1$, for some regressable formulas W_1, W_2 in $\mathcal{L}^{OS}$, then

$$\mathcal{R}^{os}[\circ W_1] = \circ\mathcal{R}^{os}[W_1] \text{ for constructor } \circ \in \{\neg, (\exists x : Q), (\forall x : Q)\}$$
$$\mathcal{R}^{os}[W_1 \vee W_2] = \mathcal{R}^{os}[W_1] \vee \mathcal{R}^{os}[W_2].$$

2. Else, if W is a non-atomic formula, W is not of the form $\neg W_1$, $W_1 \vee W_2$, $(\exists v : Q)W_1$ or $(\forall v : Q)W_1$, but of the form $W_1 \wedge W_2 \wedge \cdots \wedge W_n$ ($n \geq 2$), where each W_i ($i = 1..n$) is not of the form $W_{i,1} \wedge W_{i,2}$ for some sub-formulas $W_{i,1}, W_{i,2}$ in W_i. After using commutative law for $\wedge$, without loss of generality, there are two sub-cases:

2(a) Suppose that for some j, $j = 1..n$, each W_i ($i = 1..j$) is of the form $t_{i,1} = t_{i,2}$ for some (well-sorted) terms $t_{i,1}, t_{i,2}$, and none of W_k, $k = (j+1)..n$, is an equality between terms. In particular, when $j = n$, $\bigwedge_{k=j+1}^{n} W_k \overset{def}{=} true$. Then,

$$\mathcal{R}^{os}[W] = \begin{cases} W_1 \wedge W_2 \wedge \cdots \wedge W_j \wedge \mathcal{R}^{os}[W_0'] & \text{if there is a well-sorted mgu } \mu \text{ for } \{\langle t_{i,1}, t_{i,2}\rangle \mid i = 1..j\}; \\ false & \text{otherwise.} \end{cases}$$

Here, W_0' is a new formula obtained by applying mgu μ to $\bigwedge_{k=j+1}^{n} W_k$ and it is existentially-quantified at front for every newly introduced sort weakened variable in μ. Moreover, note that based on the assumption that we consider meet semi-lattice sort hierarchies only, such mgu is unique if it exists. Notice that $W_1 \wedge \cdots \wedge W_j$ needs to be kept, because it carries unification information between terms $\{\langle t_{i,1}, t_{i,2}\rangle | i = 1..j\}$ that cannot be omitted, and the unifiability of these terms does not mean $W_1 \wedge \cdots \wedge W_j \equiv true$.

2(b) Otherwise, $\mathcal{R}^{os}[W] = \mathcal{R}^{os}[W_1] \wedge \cdots \wedge \mathcal{R}^{os}[W_n]$.

3. Otherwise, W is atomic. There are four sub-cases.

3(a) Suppose that W is of the form $Poss(A(\vec{t}), \sigma)$ for an action term $A(\vec{t})$ and a situation term σ, and the action precondition axiom for A is of the form (1). Without loss of generality, assume that all variables in Axiom (1) have had been renamed (with variables of the same sorts) to be distinct from the free variables (if any) of W. Then,

$$\mathcal{R}^{os}[W] = \mathcal{R}^{os}[\Pi_A(\vec{t}, \sigma)].$$

3(b) Suppose that W is of the form $F(\vec{t}, do(\alpha, \sigma))$ for some relational fluent F. Let F's SSA be of the form (2). Without loss of generality, assume that all variables in Axiom (2) have had been renamed (with variables of the same sorts) to be distinct from the free variables (if any) of W. Then,

$$\mathcal{R}^{os}[W] = \mathcal{R}^{os}[\phi_F(\vec{t}, \alpha, \sigma)].$$

3(c) Suppose that atom W is of the form $t_1 = t_2$. for some well-sorted terms t_1, t_2. Then,

$$\mathcal{R}^{os}[W] = \begin{cases} W & \text{if there is a well-sorted mgu } \mu \text{ for } \langle t_1, t_2\rangle; \\ false & \text{otherwise.} \end{cases}$$

3(d) Otherwise, if atom W has S_0 as its only situation term, then

$$\mathcal{R}^{os}[W] = W.$$

Notice that although the definition seems to depend on syntactic form of a formula, we prove below that for any regressable formulas W_1 and W_2 in $\mathcal{L}^{OS}$ that are logically equivalent, their regressed results are still equivalent wrt $\mathcal{D}$ (see Corollary 7). Here are some examples.

EXAMPLE 4 Consider the order-sorted BAT $\mathcal{D}$ from Example 1 and the query W from Example 2. Then, it is easy to see that $\mathcal{R}^{os}[W] = false$, since there is no well-sorted mgu for $(d, Boston)$, where $d\!:\!Box$. Now, let W_1 be

$$\neg\forall d\!:\!Box.\, d \neq Boston \vee \neg On(d, T_1, do(load(B_1, T_1), S_0)).$$

W_1 is a sentence that is equivalent to W. It is easy to check that $\mathcal{R}^{os}[W_1]$ is a formula equivalent to $false$ (wrt $\mathcal{D}$).

Here is another example to illustrate the necessity of keeping $W_1 \wedge \cdots \wedge W_j$ in **2(a)** of Def. 3. We consider the regression of a well-sorted formula

$$InCity(B_1, city, do(drive(T_1, Boston, Toronto), S_0)),$$

where $city$ is a free variable of sort $City$.

$$\begin{aligned}
&\mathcal{R}^{os}\,[InCity(B_1, city, do(drive(T_1, Boston, Toronto), S_0))] \\
&\quad = \mathcal{R}^{os}[(\exists t\!:\!Truck, c_1\!:\!City).drive(T_1, Boston, Toronto) = drive(t, c_1, city) \\
&\qquad \wedge (B_1 = t \vee \exists b\!:\!Box.b = B_1 \wedge On(b, t, S_0))) \ \vee\ InCity(B_1, city, S_0) \\
&\qquad \wedge\neg(\exists t\!:\!Truck, c_1\!:\!City.\ drive(T_1, Boston, Toronto) = drive(t, city, c_1) \\
&\qquad \wedge (B_1 = t \vee \exists b\!:\!Box.b = B_1 \wedge On(b, t, S_0)))] \\
&\quad = \cdots\cdots \\
&\quad = (\exists t\!:\!Truck, c_1\!:\!City).drive(T_1, Boston, Toronto) = drive(t, c_1, city) \\
&\qquad \wedge\exists b\!:\!Box.b = B_1 \wedge On(B_1, T_1, S_0) \ \vee\ InCity(B_1, city, S_0) \\
&\qquad \wedge\neg(\exists t\!:\!Truck, c_1\!:\!City.drive(T_1, Boston, Toronto) = drive(t, city, c_1) \\
&\qquad \wedge\exists b\!:\!Box.b = B_1 \wedge On(b, T_1, S_0)).
\end{aligned}$$

In the above formula, for instance, if we omit the condition $drive(T_1, Boston, Toronto) = drive(t, c_1, city)$ (an example of the component $W_1 \wedge \cdots \wedge W_j$ in **2(a)** of Def. 3), we will lose unification information between the variable $city$ and the constant $Toronto$, and won't be able to maintain logical equivalence between the original formula and its regression.

Given an order-sorted BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$ and the order-sorted regression operator defined above, to show the correctness of the newly defined regression operator, we prove the following theorems similar to the theorems in [Reiter 2001].

THEOREM 5 *If W is a regressable formula wrt $\mathcal{D}$, then $\mathcal{R}^{os}[W]$ is a well-sorted $\mathcal{L}^{OS}$ formula (including $false$) that is uniform in S_0. Moreover,*

$$\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W \equiv \mathcal{R}^{os}[W].$$

THEOREM 6 *If W is a regressable formula wrt $\mathcal{D}$, then*

$$\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W \text{ iff } \mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models^{\mathrm{os}}_{\mathbb{T}} \mathcal{R}^{os}[W].$$

Hence, to reason whether $\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W$ is the same as to compute $\mathcal{R}^{os}[W]$ first and then to reason whether $\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models^{\mathrm{os}}_{\mathbb{T}} \mathcal{R}^{os}[W]$. Besides, according to Th. 5, it is easy to see that the following consequence holds.

COROLLARY 7 *If W_1 and W_2 are regressable formulas in $\mathcal{L}^{OS}$ s.t. $\models^{\mathrm{os}}_{\mathbb{T}} W_1 \equiv W_2$, then*

$$\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} \mathcal{R}^{os}[W_1] \equiv \mathcal{R}^{os}[W_2].$$

6 Order-Sorted Situation Calculus v.s. Reiter's Situation Calculus

Although BATs and regressable formulas in $\mathcal{L}^{OS}$ are based on order-sorted logic, they can be related to BATs and regressable formulas in Reiter's SC $\mathcal{L}_{sc}$.

First, given a well-sorted formula W wrt the sort theory $\mathbb{T}$ of some order-sorted BAT $\mathcal{D}$ (or simply say, wrt $\mathcal{D}$) in $\mathcal{L}^{OS}$, we define what is a *translation* of W in Reiter's SC $\mathcal{L}_{sc}$. Some concepts are introduced here for later convenience. For any sort Q in the language of $\mathcal{L}^{OS}$, we introduce a unary predicate $Q(x)$, which will be true iff x is of sort Q in $\mathcal{L}^{OS}$. Note that we can use same symbols for both sorts and their corresponding unary predicates based on the assumption that all sort symbols are distinct from usual predicate symbols in $\mathcal{L}^{OS}$.

DEFINITION 8 Consider any well-sorted formula ϕ in $\mathcal{L}^{OS}$. A *translation* of ϕ to a (well-sorted) sentence in Reiter's SC, denoted as $tr(\phi)$, is defined recursively as follows:

For every atom $P(\vec{t})$, $tr(P(\vec{t})) \stackrel{def}{=} P(\vec{t})$;
$tr(\neg\phi) \stackrel{def}{=} \neg tr(\phi)$;
$tr((\exists x\!:\!\bot)\phi) \stackrel{def}{=} false$;
$tr((\forall x\!:\!Q)\phi) \stackrel{def}{=} \neg tr((\exists x\!:\!Q.\neg\phi))$;
$tr((\exists x\!:\!Q)\phi) \stackrel{def}{=} (\exists x\!:\!Q)tr(\phi)$, if $Q \in \{Object, Act, Sit\}$;
$tr((\exists x\!:\!\top)\phi) \stackrel{def}{=} (\exists x\!:\!Object)tr(\phi) \vee (\exists x\!:\!Act)tr(\phi) \vee \exists x\!:\!Sit)tr(\phi)$;
$tr((\exists x\!:\!Q)\phi) \stackrel{def}{=} (\exists y\!:\!Object)[Q(y) \wedge tr(\phi(x/y))]$, if $Q \not\in \{\top, \bot, Object, Act, Sit\}$;
$tr(\phi \circ \psi) \stackrel{def}{=} tr(\phi) \circ tr(\psi)$ for $\circ \in \{\supset, \wedge, \vee, \supset, \equiv\}$.

The intuition behind the definition above is obvious, for any well-sorted formula W in $\mathcal{L}^{OS}$, we can always find an "equivalent" formula in Reiter's format. The meaning of equivalence between W and $tr(W)$ is formally given in Lemma 11 below.

We would like to show that the order-sorted situation calculus $\mathcal{L}^{OS}$ is correct, or *sound*, in the sense that for any BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$ we can always find a way to represent the BAT in Reiter's situation calculus $\mathcal{L}_{sc}$ (known as the corresponding BAT $\mathcal{D}'$ of $\mathcal{D}$ in $\mathcal{L}_{sc}$) such that for any regressable formula W, it can be entailed by $\mathcal{D}$ iff the translation of W in Reiter's situation calculus $\mathcal{L}_{sc}$ can be entailed by the corresponding BAT $\mathcal{D}'$ in $\mathcal{L}_{sc}$. Later, the corresponding BAT $\mathcal{D}'$ of $\mathcal{D}$ is denoted as $TR(\mathcal{D})$ to remind that it is constructed out of $\mathcal{D}$. That is,

THEOREM 9 **(Soundness)** *For any order-sorted BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$ in $\mathcal{L}^{OS}$, there exists a corresponding BAT $\mathcal{D}'$ (denoted as $TR(\mathcal{D})$ below), such that*

$$\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W \ \textit{ iff } \ TR(\mathcal{D}) \models^{\mathrm{ms}} tr(W)$$

for any regressable sentence (i.e., a query) W.

It is hard to prove Th. 9 directly. Inspired by the *standard relativization* of order-sorted logic to unsorted predicate logic, our general idea of proving Th. 9 is as follows (see the diagram in Fig. 1). In Step 1, we construct a BAT $TR(\mathcal{D})$ (called the *corresponding Reiter's BAT of $\mathcal{D}$* above) in Reiter's situation calculus. In Step 2, we prove that there is an unsorted theory $\mathcal{D}''$ (*strong relativization* of $\mathcal{D}$) and an unsorted first-order sentence W'' (*relativization* of W) such that $\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W$ iff $\mathcal{D}'' \models^{\mathrm{fo}} W''$. In Step 3, we show that $TR(\mathcal{D}) \models^{\mathrm{ms}} tr(W)$ iff $\mathcal{D}''' \models^{\mathrm{fo}} W'''$, for some unsorted theory $\mathcal{D}'''$ (standard relativization of $TR(\mathcal{D})$) and first-order sentence W''' (relativization of $tr(W)$). Finally, in Step 4, we show that $\mathcal{D}''' \models^{\mathrm{fo}} W'''$ iff $\mathcal{D}'' \models^{\mathrm{fo}} W''$.

$$\begin{array}{ccc}
\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} W & \overset{\text{(Step 2)}}{\Longleftrightarrow} & \mathcal{D}'' \models^{\mathrm{fo}} W'' \\
 & & \Updownarrow \text{(Step 4)} \\
TR(\mathcal{D}) \models^{\mathrm{ms}} tr(W) \text{ (Step 1)} & \overset{\text{(Step 3)}}{\Longleftrightarrow} & \mathcal{D}''' \models^{\mathrm{fo}} W'''
\end{array}$$

Figure 1. Diagram of the outline for proving Theorem 9

The following definition of relativization of order-sorted logic to unsorted predicate logic (Def. 10) and the bridge axioms (Def. 12) were given in [Oberschelp 1962; Schmidt 1951; Walther 1987; Schmidt-Schauβ 1989].

DEFINITION 10 For any well-sorted formula ϕ in $\mathcal{L}^{OS}$, $rel(\phi)$, a *relativization* of ϕ, is an unsorted formula defined as:

For every atom $P(\vec{t})$, $rel(P(\vec{t})) \overset{def}{=} P(\vec{t})$;
$rel(\neg\phi) \overset{def}{=} \neg rel(\phi)$;
$rel(\phi \circ \psi) \overset{def}{=} rel(\phi) \circ rel(\psi)$ for $\circ \in \{\wedge, \vee, \supset\}$;
$rel((\forall x\!:\!Q)\phi) \overset{def}{=} (\forall y)[Q(y) \supset rel(\phi[x/y])]$;
$rel((\exists x\!:\!Q)\phi) \overset{def}{=} (\exists y)[Q(y) \wedge rel(\phi[x/y])]$.

Moreover, for any set Set of well-sorted formulas, $rel(Set) = \{rel(\phi) \mid \phi \in Set\}$.

Note that Reiter's SC $\mathcal{L}_{sc}$ is in fact based on many-sorted logic, which is a special case of order-sorted logic, with three disjoint sorts (Act, Sit and $Object$). All formulas in $\mathcal{L}_{sc}$ are well-sorted wrt the sort theory of $\mathcal{L}_{sc}$ with all quantified variables restricted to suitable sorts by default. Hence, the definition of rel can also be applied to any formula in Reiter's SC.

Now, it is straightforward to prove the following lemma (Lemma 11) for rel and tr by structural induction, which shows the equivalence relationship between a well-sorted formula W in $\mathcal{L}^{OS}$ and its translation $tr(W)$ in $\mathcal{L}_{sc}$.

LEMMA 11 *Consider any well-sorted formula W in $\mathcal{L}^{OS}$. Then, given the default assumption that everything in the universe is either an action, or a situation, or an object, we have* $\models^{\text{fo}} rel(tr(W)) \equiv rel(W)$.

DEFINITION 12 For any sort theory $\mathbb{T}$, which includes predicate declarations, function declarations and/or subsort declarations, *the set of bridge axioms* of $\mathbb{T}$, $BA(\mathbb{T})$, is a set of unsorted formulas as follows:

(a) $(\forall x).\, Q_2(x) \supset Q_1(x)$ for each $Q_2 \leq Q_1$ in $\mathbb{T}$;

(b) $Q(c)$ for each $c\!:\!Q$ in $\mathbb{T}$;

(c) $(\forall \vec{x}_{1..n}).\, \bigwedge_{i=1}^{n} Q_i(x_i) \supset Q(f(\vec{x}_{1..n}))$ for each function $f\!:\!\vec{Q}_{1..n} \to Q$ in $\mathbb{T}$.

Note that in particular, when we compute the bridge axioms for a sort theory $\mathbb{T}$ in a given order-sorted BAT $\mathcal{D}$, $Sit(S_0)$ is always included in $BA(\mathbb{T})$ for $S_0\!:\!Sit$ in $\mathbb{T}$, and the axioms of the form (c) are introduced for all functions, including action functions and the special situation function $do(a, s)$.

Based on the definition of relativization and the bridge axioms, the following lemma has been proved in [Schmidt-Schauβ 1989; Walther 1987; Bierle, Hedtstück, Pletat, Schmitt, and Siekmann 1992]: *For any well-sorted sentence ϕ wrt a sort theory $\mathbb{T}$, we have that* $\models^{\text{os}}_{\mathbb{T}} \phi$ *iff* $BA(\mathbb{T}) \models^{\text{fo}} rel(\phi)$.

We then define the standard relativization as follows.

DEFINITION 13 Consider a sort theory $\mathbb{T}$ in an order-sorted (or many-sorted) logic and a set of well-sorted axioms $\mathcal{K}$ wrt the given sort theory. Then, the *standard relativization of* $\mathcal{K}$, an unsorted theory, is defined as

$$REL(\mathcal{K}) \stackrel{def}{=} rel(\mathcal{K}) \cup BA(\mathbb{T}).$$

In particular, for any BAT $\mathcal{D}_1$ in Reiter's SC $\mathcal{L}_{sc}$ that has a finite set $\mathbb{T}_{\mathcal{D}_1}$ of function declarations and predicate declarations for all predicates and functions appeared in $\mathcal{D}_1$, the *standard relativization of* $\mathcal{D}_1$ is

$$REL(\mathcal{D}_1) \stackrel{def}{=} rel(\mathcal{D}_1 - \phi_\Sigma) \cup BA(\mathbb{T}_{\mathcal{D}_1}) \cup \{rel(\phi_\Sigma)\},$$

where ϕ_Σ is one of the foundational axioms representing the second-order induction axiom

$$(\forall P).P(S_0) \wedge (\forall a, s)[P(s) \supset P(do(a, s))] \supset (\forall s)P(s), \tag{3}$$

and the relativization of ϕ_Σ, $rel(\phi_\Sigma)$, is defined as

$$\begin{aligned}(\forall P).P(S_0) \wedge (\forall a, s)[Act(a) \wedge Sit(s) \supset (P(s) \supset P(do(a, s)))]\\ \supset (\forall s)Sit(s) \supset P(s).\end{aligned} \tag{4}$$

It is easy to see that the standard relativization of a BAT of Reiter's SC is a very slight extension of the standard relativization of a set of well-sorted (first-order) formulas by applying the (standard) relativization function to a (second-order) well-sorted formula. Therefore, similar to the Relativization Theorem proved in [Schmidt-Schauβ 1989], we have:

LEMMA 14 *Consider any regressable formula W with a background BAT $\mathcal{D}$ in Reiter's SC $\mathcal{L}_{sc}$. Then,*

$$\mathcal{D} \models^{\rm ms} W \text{ iff } REL(\mathcal{D}) \models^{\rm fo} rel(W).$$

Now we proceed to Step 1 mentioned in Fig. 1 of the outline.

Consider any order-sorted BAT $\mathcal{D}$. We construct the *corresponding Reiter's BAT of $\mathcal{D}$*, denoted as $TR(\mathcal{D})$, that will be the Reiter's BAT we are looking for in Th. 9. In $TR(\mathcal{D})$,we introduce three new special predicates $SortedObj(x)$ $SortedAct(a)$ and $SortedSit(s)$. Intuitively, for any term t (a, or s, respectively) of sort object (action, or situation, respectively), $SortedObj(t)$ ($SortedAct(a)$, or $SortedSit(s)$, respectively) means that t (a, or s, respectively) needs to be well-sorte with respect to the given sort theory $\mathbb{T}$ in the order-sorted BAT $\mathcal{D}$. Note that the reason why we introduce three different special predicates for well-sorte terms ($SortedObj(x)$, $SortedAct(a)$ and $SortedSit(s)$) is because Reiter's Situation Calculus is a many-sort logic with three sorts only and his BATs have a particular syntactic format. For instance, every formula in an initial theory needs to be uniform in the initial situation S_0, and every SSA has to be of the form $F(\vec{x}, do(a, s)) \equiv \phi_F(\vec{x}, a, s)$. In order to construct a BAT in Reiter' Situation Calculus that satisfies Th. 9, we need to "encode" information about the well-sortedness of terms into the constructed BAT. However, if we introduce onl one predicate to describe well-sortedness, say $sorted(x)$ for any x (including objects, actions and situations) representin x is well-sorted then it would be problematic when we want to axiomatize the property of $sorted$ – it neither can b considered as an axiom in the initial theory (since it is not uniform in S_0), nor can be considered as an SSA (since its last argument is not of sort situation). Notice that in [Reiter 2001], sorted quantifiers are omitted as a convention, because their sorts are always obvious from context. Hence, when we construct the BAT $TR(\mathcal{D})$ in Reiter's SC below, all free variables are implicitly universally sorted-quantified according to their obvious sorts. The declarations for functions and predicates (including for predicates $SortedObj$, $SortedAct$ and $SortedSit$) are always standard, hence are not mentioned here.

• $TR(\mathcal{D})$ includes the standard foundational axioms and the set of unique name axioms for action functions in Reiter's SC.

• The initial theory of $TR(\mathcal{D})$, say $\mathcal{D}'_{S_0}$, includes the following axioms.

1. For any well-sorted sentence $\phi \in \mathcal{D}_{S_0}$, $tr(\phi)$ is in $\mathcal{D}'_{S_0}$.

2. For each declaration $Q_2 \leq Q_1$ in $\mathbb{T}$, add an axiom

$$tr((\forall x\!:\!\top).(\exists y_2\!:\!Q_2.x = y_2) \supset (\exists y_1\!:\!Q_1.x = y_1)).$$

3. For any constant declaration $c : Q$ where $Q \leq_{\mathbb{T}} Object$ and $Q \neq Object$, add an axiom $Q(c)$. Note that other constant declarations will still be kept in the sort theory of $TR(\mathcal{D})$ in language $\mathcal{L}_{sc}$ by default. For example, $S_0 : Sit$, $C : Object$ for any constant object C appeared in $TR(\mathcal{D})$ and $A : Act$ for any constant action function A.

4. For each (situation-independent) function f (including action function) whose declaration is $f : \vec{Q}_{1..n} \rightarrow Q$ in $\mathbb{T}$ ($n \geq 1$), add an axiom $tr((\forall \vec{x}_{1..n} : \vec{Q}_{1..n}).(\exists y : Q).y = f(\vec{x}_{1..n}))$.

5. The axioms related to $SortedObj$, $SortedAct$, and $SortedSit$:

(a) $SortedObj(y) \equiv tr(\bigvee_{i=1}^{k}(\exists x_{i,1} : Q_{i,1}, \ldots, x_{i,n_i} : Q_{i,n_i}).y = f_i(x_{i,1}, \ldots, x_{i,n_i}) \wedge \bigwedge_{j=1}^{n_i} SortedObj(x_{i,j}))$, where $f_1, \ldots, f_k$ (including constant Objects) are all functions other than action functions and do function included in $\mathcal{D}$, and the function declaration for each f_i in $\mathbb{T}$ is $f_i : Q_{i,1} \times \cdots \times Q_{i,n_i} \rightarrow Q_{i,1+n_i}$ (each $Q_{i,j} \leq_{\mathbb{T}} Object$);

(b) $SortedAct(a) \equiv tr(\bigvee_{i=1}^{m}(\exists x_{i,1} : Q_{i,1}, \ldots, x_{i,n_i} : Q_{i,n_i}).a = A_i(x_{i,1}, \ldots x_{i,n_i}) \wedge \bigwedge_{j=1}^{n_i} SortedObj(x_{i,j}))$, where $A_1, \ldots, A_m$ (including constant action functions) are all action functions included in $\mathcal{D}$, and the function declaration for each A_i in $\mathbb{T}$ is $A_i : Q_{i,1} \times \cdots \times Q_{i,n_i} \rightarrow Act$ (each $Q_{i,j} \leq_{\mathbb{T}} Object$);

(c) $tr(P(\vec{x}_{1..n}) \supset \bigwedge_{i=1}^{n}(\exists y_i : Q_i.y_i = x_i \wedge SortedObj(x_i)))$ for each situation-independent predicate $P : \vec{Q}_{1..n}$ in $\mathbb{T}$;

(d) $SortedSit(S_0)$;

(e) $tr(F(\vec{x}_{1..n}, S_0) \supset \bigwedge_{i=1}^{n}(\exists y_i : Q_i.y_i = x_i \wedge SortedObj(x_i)))$ for each fluent $F : \vec{Q}_{1..n} \times Sit$ in $\mathbb{T}$. Here, all y_i's are distinct auxiliary variables never appearing in $\vec{x}_{1..n}$.

• For action $A(\vec{x}_{1..n})$ whose precondition axiom in $\mathcal{D}_{ap}$ has the form Eq.(1), we replace it with a precondition axiom in the format of Reiter's SC:

$$Poss(A(\vec{x}_{1..n}), s) \equiv \Pi'_A(\vec{x}_{1..n}, s),$$

where $\Pi'_A(\vec{x}_{1..n}, s)$ is uniform in s, resulting from

$$tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}).(\bigwedge_{i=1}^{n} x_i = y_i \wedge SortedObj(x_i)) \wedge \Pi_A(\vec{y}_{1..n}, s)).$$

Here, all y_i's are distinct auxiliary variables never appearing in $\Pi_A(\vec{x}_{1..n}, s)$.

• The set of successor state axioms of $TR(\mathcal{D})$ now includes the following axioms:

1. For each relational fluent $F(\vec{x}_{1..n}, s)$, whose SSA in $\mathcal{D}_{ss}$ is of the form Eq.(2), we replace it with SSA in the format of Reiter's SC:

$$F(\vec{x}_{1..n}, do(a, s)) \equiv \phi'_F(\vec{x}_{1..n}, a, s),$$

where $\phi'_F(\vec{x}_{1..n}, a, s)$ is uniform in s, resulting from

$tr(SortAct(a) \wedge SortedSit(s) \wedge (\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \bigwedge_{i=1}^{n}(x_i = y_i \wedge SortedObj(x_i)) \wedge \phi_F(\vec{y}_{1..n}, a, s))$.

Here, all y_i's are distinct auxiliary variables never appearing in $\phi_F(\vec{x}_{1..n}, s)$.

2. $SortedSit(do(a, s)) \equiv SortedAct(a) \wedge SortedSit(s)$.

Now, we define a different relativization, the *strong relativization*, for BATs in order-sorted SC $\mathcal{L}^{OS}$ (Def. 15) to help us prove Th. 9 because of the following reasons: (1) We include the sort theory in each order-sorted BAT in the language of $\mathcal{L}^{OS}$, while Reiter's SC mentions sort declarations generally in the signature of $\mathcal{L}_{sc}$. (2) We are not able to use standard relativization to relate order-sorted BATs to BATs in Reiter's SC directly because of the particular syntactic formats of BATs in Reiter's SC. (3) We expect that any predicates (including fluents) will be true only if they are for "reasonable" types of objects in unsorted logic, i.e., for well-sorted terms wrt a given sort theory in order-sorted logic.

DEFINITION 15 For any order-sorted BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$ in $\mathcal{L}^{OS}$, besides introducing unary predicates that correspond to sorts in $\mathbb{T}$, same as the special new predicates introduced in the corresponding Reiter's BAT of $\mathcal{D}$, $TR(\mathcal{D})$, we also use $SortedObj(x)$,

($SortedAct(a)$ and $SortedSit(s)$, respectively) to represent that t (a, or s, respectively) is well-sorted with respect to the given sort theory $\mathbb{T}$ in the order-sorted BAT $\mathcal{D}$.

The *strong relativization of* $\mathcal{D}$ is an unsorted theory defined as

$$REL_S(\mathcal{D}) \stackrel{def}{=} rel_S(\mathbf{D}) \cup BA(\mathbb{T}),$$

where $rel_S(\mathbf{D})$ is a set of axioms including the following axioms.

(a) $(\forall \vec{x}_{1..n}).\bigwedge_{i=1}^{n} Q'_i(x_i) \supset Q'_{n+1}(f(\vec{x}_{1..n}))$, where each Q'_j ($j = 1..n{+}1$) is a predicate in $\{Action, Situation, Object\}$ and its corresponding sort Q'_j satisfies $Q_j \leq_{\mathbb{T}} Q'_j$, for any function $f : \vec{Q}_{1..n} \to Q_{n+1}$ in $\mathbb{T}$ (including constant functions, action functions and $do(a, s)$)[1].

(b) all axioms in $rel(\mathcal{D}_{S_0} \cup \Sigma - \{\phi_\Sigma\})$, where ϕ_Σ is Axiom (3).

(c) the relativization of Axiom (3), i.e., Axiom (4).

(d) $$(\forall \vec{x}_{1..n}, \vec{y}_{1..n}).\bigwedge_{i=1}^{n} \big(Object(x_i) \wedge Object(y_i)\big) \supset \big(A(\vec{x}_{1..n}) = A(\vec{y}_{1..n}) \supset \bigwedge_{i=1}^{n} x_i = y_i\big)$$
for each action function symbol A.

(e) $(\forall \vec{x}_{1..n}, \vec{y}_{1..m}).\bigwedge_{i=1}^{n} Object(x_i) \wedge \bigwedge_{j=1}^{m} Object(y_i) \supset A(\vec{x}_{1..n}) \neq B(\vec{y}_{1..m})$ for any two distinct action function symbols A and B.

(f) $(\forall y).Object(y) \supset [SortedObj(y) \equiv \bigvee_{i=1}^{k} (\exists x_{i,1}, \ldots, x_{i,n_i}).y = f_i(x_{i,1}, \ldots, x_{i,n_i}) \wedge (\bigwedge_{j=1}^{n_i} Q_{i,j}(x_{i,j}) \wedge SortedObj(x_{i,j}))]$, where $f_1, \ldots, f_k$ (including constant Objects) are *all* functions other than action functions and do function included in $\mathcal{D}$, and the function declaration for each f_i in $\mathbb{T}$ is $f_i : Q_{i,1} \times \cdots \times Q_{i,n_i} \to Q_{i,n_i+1}$ (each $Q_{i,j} \leq_{\mathbb{T}} Object$). Note that for any i, if $n_i = 0$ (i.e., f_i is a constant object), there are no quantifiers for variables $x_{i,1}, \ldots, x_{i,n_i}$ at the front and $\bigwedge_{j=1}^{n_i} Q_{i,j}(x_{i,j}) \wedge SortedObj(x_{i,j})) \equiv true$.

(g) $(\forall a).Action(a) \supset [SortedAct(a) \equiv \bigvee_{i=1}^{m} (\exists x_{i,1}, \ldots, x_{i,n_i}).a = A_i(x_{i,1}, \ldots, x_{i,n_i}) \wedge (\bigwedge_{j=1}^{n_i} Q_{i,j}(x_{i,j}) \wedge SortedObj(x_{i,j}))]$, where $A_1, \ldots, A_m$ (including constant action functions) are *all* action functions included in $\mathcal{D}$, and the function declaration for each A_i in $\mathbb{T}$ is $A_i : Q_{i,1} \times \cdots \times Q_{i,n_i} \to Action$ (each $Q_{i,j} \leq_{\mathbb{T}} Object$). Note that for any i, if $n_i = 0$ (i.e., A_i is a constant action function), there are no quantifiers for variables $x_{i,1}, \ldots, x_{i,n_i}$ at the front and $\bigwedge_{j=1}^{n_i} Q_{i,j}(x_{i,j}) \wedge SortedObj(x_{i,j})) \equiv true$.

(h) $(\forall \vec{x}_{1..n}).\bigwedge_{i=1}^{n} Object(x_i) \supset [P(\vec{x}_{1..n}) \supset \bigwedge_{i=1}^{n} Q_i(x_i) \wedge SortedObj(x_i)]$ for each situation-independent predicate $P : \vec{Q}_{1..n}$ in $\mathbb{T}$.

(i) $SortedSit(S_0)$.

(j) $$(\forall a, s).Action(a) \wedge Situation(s) \supset [SortedSit(do(a, s)) \equiv SortedAct(a) \wedge SortedSit(s)].$$

[1] In particular, when $n = 0$, $f(\vec{x}_{1..n})$ is a constant function c, and we have $Q'(f)$, where Q' is a predicate in $\{Action, Situation, Object\}$ and its corresponding sort Q' satisfies $Q \leq_{\mathbb{T}} Q'$, for $c : Q$ in $\mathbb{T}$ (including constant action functions and the initial situation S_0).

(k) $(\forall \vec{x}_{1..n}, a, s). \bigwedge_{i=1}^{n} Object(x_i) \wedge Action(a) \wedge Situation(s) \supset [F(\vec{x}_{1..n}, do(a,s)) \equiv \bigwedge_{i=1}^{n} Q_i(x_i) \wedge SortedObj(x_i) \wedge SortedAct(a) \wedge SortedSit(s) \wedge rel(\phi_F(\vec{x}_{1..n}, a, s))]$ for each fluent F, whose SSA in $\mathcal{D}$ is of the form Axiom (2).

(l) $(\forall \vec{x}_{1..n}). \bigwedge_{i=1}^{n} Object(x_i) \supset [F(\vec{x}_{1..n}, S_0) \supset \bigwedge_{i=1}^{n} Q_i(x_i) \wedge SortedObj(x_i)]$ for each fluent $F : \vec{Q}_{1..n} \times Situation$ in $\mathbb{T}$.

(m) $(\forall \vec{x}_{1..n}, s). \bigwedge_{i=1}^{n} Object(x_i) \wedge Situation(s) \supset [Poss(A(\vec{x}_{1..n}), s) \equiv \bigwedge_{i=1}^{n}(Q_i(x_i) \wedge SortedObj(x_i)) \wedge SortedSit(s) \wedge rel(\Pi_A(\vec{x}_{1..n}, s))]$ for each n-ary action function A, whose precondition axiom in $\mathcal{D}$ is of the form Axiom (1).

We can also prove a relativization theorem as follows for the strong relativization similar to the Relativization Theorem proved in [Schmidt-Schauβ 1989].

LEMMA 16 *Consider any regressable formula W with a background BAT $\mathcal{D} = (\mathbb{T}, \mathbf{D})$ in order-sorted SC $\mathcal{L}^{OS}$. Then,*

$$\mathbf{D} \models^{\text{os}}_{\mathbb{T}} W \textit{ iff } REL_S(\mathcal{D}) \models^{\text{fo}} rel(W).$$

We can prove Step 2 in Fig. 1 using Lemma 16. Because Reiter's SC is a many-sorted logical language that has particular syntactic formats for precondition axioms and SSAs, we cannot use rel to relate $\mathcal{D}$ in $\mathcal{L}^{OS}$ with a Reiter's BAT directly. It is also the reason why strong relativization is introduced.

Proof of Th. 9. Overall, in Fig. 1, in Step 1, $TR(\mathcal{D})$ is constructed as above. Let $\mathcal{D}'' = REL_S(\mathcal{D})$ and $W'' = rel(W)$, and Step 2 can be proved by using Lemma 16. Let $W''' = rel(tr(W))$ and $\mathcal{D}''' = REL(\mathcal{D}')$ (see Def. 13), and Step 3 can be proved by using Lemma 14. Finally, Step 4 is true according to Lemma 11.

It is important to notice that any query in $\mathcal{L}^{OS}$ has to be well-sorted wrt a given background order-sorted BAT $\mathcal{D}$; while, in general, a query that can be answered in the corresponding Reiter's BAT of $\mathcal{D}$ are not necessarily well-sorted wrt $\mathcal{D}$. Below, Th. 17 shows that for any query that can be answered in $TR(\mathcal{D})$, it can be answered in $\mathcal{D}$ in a "well-sorted" way too. Proof details are omitted due to the space limitations. But, we provide some examples below to illustrate the statement.

THEOREM 17 **(Completeness)** *Let $\mathcal{D}$ be an order-sorted BAT in $\mathcal{L}^{OS}$, and $TR(\mathcal{D})$ be its corresponding Reiter's BAT. Consider any regressable formula W in Reiter's SC, in which there is no appearance of special predicates $SortedObj$, $SortedAct$ or $SortedSit$, W can be translated to a (well-sorted) formula wrt $\mathcal{D}$, denoted as $os(W)$ below, such that*

$$TR(\mathcal{D}) \models^{\text{ms}} tr(os(W)) \equiv W.$$

Furthermore, we have

$$TR(\mathcal{D}) \models^{\text{ms}} W \textit{ iff } \mathbf{D} \models^{\text{os}}_{\mathbb{T}} os(W)$$

when W is a regressable sentence wrt $TR(\mathcal{D})$.

We provide an example to illustrate some axioms in the corresponding Reiter's BAT of an order-sorted BAT in Th. 9. We also give some examples to illustrate the idea of Th. 17.

EXAMPLE 18 Consider the BAT $\mathcal{D}$ from Example 1. Most of the axioms in $TR(\mathcal{D})$ are obvious and we just provide examples of the axiom of $SortedObj$, a precondition axiom and an SSA:

$SortedObj(x) \equiv x = B_1 \vee x = B_2 \vee x = Boston \vee x = Toronto$
$\quad \vee x = T_1 \vee x = T_2 \vee \exists y. City(y) \wedge x = twinCity(y),$
$Poss(load(x,t), s) \equiv Box(x) \wedge Truck(t) \wedge \neg On(x,t,s) \wedge$
$\quad (\exists y. City(y) \wedge InCity(x,y,s) \wedge InCity(t,y,s)),$
$InCity(d,c,do(a,s)) \equiv MovObj(d) \wedge City(c) \wedge$
$\quad [(\exists t, c_1. Truck(t) \wedge City(c_1) \wedge a = drive(t, c_1, c)$
$\quad \wedge (d = t \vee \exists b. Box(b) \wedge b = d \wedge On(b,t,s))) \vee InCity(d,c,s) \wedge$
$\quad \neg(\exists t, c_1. Truck(t) \wedge City(c_1) \wedge a = drive(t, c, c_1)$
$\quad \wedge (d = t \vee \exists b. Box(b) \wedge b = d \wedge On(b,t,s)))].$

Now, let $On(twinCity(Boston), T_1, s)$ (denoted as W_3) be a regressable formula in $\mathcal{L}_{sc}$, where s is a variable of sort situation. According to the way $TR(\mathcal{D})$ is constructed, we have $TR(\mathcal{D}) \models^{\rm ms} On(o,t,s) \supset Box(o)$. Then, for any situation s, if $TR(\mathcal{D}) \models^{\rm ms} On(twinCity(Boston), T_1, s)$, we need to have $TR(\mathcal{D}) \models^{\rm ms} Box(twinCity(Boston))$, which in fact does not hold according to the axioms in $TR(\mathcal{D})$. Hence, $TR(\mathcal{D}) \models^{\rm ms} W_3 \equiv os(W_3)$, where $os(W_3) = false$.

Let W_4 be $\forall s. \exists c. \neg InCity(B_1, twinCity(c), s)$, which is a regressable sentence in $\mathcal{L}_{sc}$, where $c : Object$ and $s : Sitcuation$ hold by default. Then, $os(W_4)$ is $\forall s : Sit. \exists c : Object. \neg(\exists c_1 : City. c_1 = c \wedge InCity(B_1, twinCity(c_1), s))$. Since $TR(\mathcal{D}) \models^{\rm ms} InCity(B_1, twinCity(c), s) \supset City(c)$, it is easy to see $TR(\mathcal{D}) \models^{\rm ms} W_4 \equiv tr(os(W_4))$.

7 Computational Advantages of $\mathcal{L}^{OS}$

Given any BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$, it is easy to see that Reiter's regression operator $\mathcal{R}$ [Reiter 2001] still can be applied to (well-sorted) regressable formulas (wrt $\mathcal{D}$). Moreover, one can prove that $\mathcal{R}[W]$ is a formula in $\mathcal{L}^{OS}$ uniform in S_0 and $\mathbf{D} \models^{\rm os}_{\mathbb{T}} W \equiv \mathcal{R}[W]$. However, using the order-sorted regression operator $\mathcal{R}^{os}$ sometimes can give us computational advantages in comparison to using Reiter's regression operator $\mathcal{R}$. But first of all, we show that the computational complexity of using $\mathcal{R}^{os}$ is no worse than that of $\mathcal{R}$.

For the regression operator $\mathcal{R}$ that can be used either in $\mathcal{L}^{OS}$ or in $\mathcal{L}_{sc}$ ($\mathcal{R}^{os}$ used in $\mathcal{L}^{OS}$, respectively), we can construct a *regression tree* rooted at W for any regressable query W in either language. Each node in a regression tree of $\mathcal{R}[W]$ ($\mathcal{R}^{os}[W]$, respectively) corresponds to a sub-formula computed by regression, and each edge corresponds to one step of regression according to the definition of the regression operator. In the worst case scenario, for any query W in $\mathcal{L}^{OS}$, the regression tree of $\mathcal{R}^{os}[W]$ will have the same number of nodes as the regression tree of $\mathcal{R}[W]$ (and linear to the number of nodes in the regression tree of $\mathcal{R}[tr(W)]$ wrt $TR(\mathcal{D})$). Moreover, based on the assumption that our sort theory of $\mathcal{D}$ is simple with empty equational theory, whose corresponding sort hierarchy is a meet semi-lattice, finding a unique (well-sorted) mgu takes the same time as in the unsorted case [Schmidt-Schauβ 1989; Jouannaud and Kirchner 1991; Weidenbach 1996]. Hence, the overall computational complexity of building the regression tree of $\mathcal{R}^{os}[W]$ is at most linear to the size of Reiter's regression tree.

THEOREM 19 *Consider any regressable sentence W with a background BAT $\mathcal{D}$ in order-sorted SC $\mathcal{L}^{OS}$. Then, in the worst case scenario, the complexity of computing $\mathcal{R}^{os}[W]$ is the same as that of computing $\mathcal{R}[W]$, which is also the same as the complexity of computing $\mathcal{R}[tr(W)]$ in $TR(\mathcal{D})$, the corresponding Reiter's BAT.*

On the other hand, under some circumstances, the regression of a query in $\mathcal{L}^{OS}$ using $\mathcal{R}^{os}$ instead of $\mathcal{R}$ will give us computational advantages. Consider any query (i.e., a regressable sentence) W with a background BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$. Then, the computation of $\mathcal{R}^{os}[W]$ wrt $\mathcal{D}$ can sometimes terminate earlier than that of $\mathcal{R}[W]$ wrt $\mathcal{D}$, and also earlier than the computation of $\mathcal{R}[tr(W)]$ wrt $TR(\mathcal{D})$. In particular, we have:

PROPERTY 20 *Consider an order-sorted BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$, at least one of the SSAs in $\mathcal{D}$ is not context-free, and any regressable formula W of the syntactic form $t_{1,1} = t_{1,2} \wedge \ldots \wedge t_{m,1} = t_{m,2} \wedge W_1$. Let the size of W (including the length of the terms in W) be n. If there is no well-sorted mgu for equalities between terms $\{\langle t_{i,1}, t_{i,2}\rangle \mid i = 1..m\}$, then in the worst-case scenario, computing $\mathcal{R}^{os}[W]$ runs in $O(n)$, while computing $\mathcal{R}[W]$ with respect to $\mathcal{D}$ ($\mathcal{R}[tr(W)]$ with respect to $TR(\mathcal{D})$) runs in time $O(2^n)$. Moreover, the size of the resulting formula of $\mathcal{R}^{os}[W]$, which is $false$, is always constant, while the size of the resulting formula using $\mathcal{R}$ is $O(2^n)$.*

According to the definition of Reiter's regression operator, the equalities will be kept and regression will be further performed on W_1 (or on $tr(W_1)$ in $TR(\mathcal{D})$, respectively), which in general takes exponential time wrt the length of W_1 and causes exponential blow-up in the size of the formula. Once Reiter's regression has terminated, a theorem prover will find that the resulting formula is false either because there is no mgu for terms when reasoning is performed in $\mathcal{L}^{OS}$ (or, due to the clash between sort related predicates when reasoning in $\mathcal{L}_{sc}$, respectively). Hence, using the order-sorted regression operator can sometimes prune brunches of the regression tree built by $\mathcal{R}$ exponentially (wrt the size of the regressed formula), and therefore make regression terminated exponentially faster.

We provide an example below to show the computational advantage of using $\mathcal{R}^{os}$. This example also illustrates the the class of conjunctive queries in Property 20 is common in regression and leads to significant savings if regression trees are pruned earlier on.

EXAMPLE 21 Consider the BAT $\mathcal{D}$ from Example 1. Let W_5 be a $\mathcal{L}^{OS}$ query, i.e., a (well-sorted) regressable sentence,

$$InCity(T_1, Toronto, do(drive(T_1, Boston, Toronto), S_1)),$$

where S_1 is a well-sorted ground situation term that involves a long sequence of actions. According to the SSA of $InCity$, at the branch of computing $\mathcal{R}^{os}[\exists b : Box.b = T_1 \wedge On(b, t, S_1)]$ in the regression tree, since there is no well-sorted mgu for (b, T_1), the application of order-sorted regression equals to $false$ immediately. However, using Reiter's regression operator (no matter in $\mathcal{D}$ or in $TR(\mathcal{D})$), his operator will keep doing useless regression on $On(b, t, S_1)$ until getting (a potentially huge) sub-formula uniform in S_0. Once his regression has terminated, such sub-formula will also be proved equivalent to $false$ wrt the initial theory ($\mathcal{D}_{S_0}$ or $TR(\mathcal{D})_{S_0}$, respectively) using a theorem prover, for the same reason as above.

In addition, since our sort theory of a BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$ is finite and it has one and only one declaration for each function and predicate symbol, for any query W (wrt $TR(\mathcal{D})$) in $\mathcal{L}_{sc}$, it takes linear time (wrt the length of the query) to find a well-sorted formula $os(W)$ in $\mathcal{L}^{OS}$ that satisfies Th. 17. But, reasoning whether $\mathbf{D} \models^{\mathrm{os}}_{\mathbb{T}} os(W)$ (starting from finding $os(W)$) sometimes can terminate exponentially earlier than finding whether $TR(\mathcal{D}) \models^{\mathrm{ms}} W$. In particular, we study a certain class of formulas.

DEFINITION 22 Let $\mathcal{D}$ be a BAT in the order-sorted SC $\mathcal{L}^{OS}$, and $TR(\mathcal{D})$ be its corresponding Reiter's BAT. Any term t in Reiter's SC is a *possibly sortable term wrt $\mathcal{D}$*, if one of the following conditions holds:
(1) t is a variable of sort Act, $Object$ or Sit in $\mathcal{L}_{sc}$;
(2) t is a constant c, and $c\!:\!Q$ in $\mathbb{T}$ (we say that the sort of c is Q wrt $\mathcal{D}$); or,
(3) t is of form $f(\vec{x}_{1..n})$, function declaration $f:\vec{Q}_{1..n}\rightarrow Q$ in $\mathbb{T}$, for every i ($i=1..n$), t_i either is a variable or is a non-variable possibly sortable term of sort Q'_i wrt $\mathcal{D}$ and $Q'_i \leq_{\mathbb{T}} Q_i$ in $\mathbb{T}$ (we say that the sort of $f(\vec{t}_{1..n})$ is Q wrt $\mathcal{D}$).

Similarly, any atom $P(\vec{t}_{1..n})$ in Reiter's SC (well-sorted wrt $TR(\mathcal{D})$) and P is not $SortedObj$, $SortedAct$ or $SortedSit$, is a *possibly sortable atom wrt $\mathcal{D}$*, if for every i, t_i either is a variable or is a non-variable term of sort Q'_i wrt $\mathcal{D}$ satisfying that:
(a) it is a possibly sortable term wrt $\mathcal{D}$; and
(b) $P\!:\!\vec{Q}_{1..n}$ is in $\mathbb{T}$ and $Q'_i \leq_{\mathbb{T}} Q_i$ wrt $\mathcal{D}$.

Any regressable formula W in Reiter's SC (well-sorted wrt $TR(\mathcal{D})$) is *possibly sortable wrt $\mathcal{D}$* if every atom in W is possibly sortable wrt $\mathcal{D}$.

Note that the predicate of a possibly sortable atom can be equality, $Poss$ or any predicate appeared in $\mathcal{D}$.

Given any $\mathcal{D}$ in order-sorted SC, it is easy to see that every atom (term, respectively) in $TR(\mathcal{D})$ that can be considered as well-sorted wrt $\mathcal{D}$ is always a possibly sortable atom (term, respectively); while a possibly sortable atom (term, respectively) is not necessarily well-sorted wrt $\mathcal{D}$. We provide some simple examples of the terms and atoms defined in Def. 22.

EXAMPLE 23 We continue with Example 18. The query

$$\exists x.\exists y.InCity(x, twinCity(y), do(load(B_1, T_1), S_0))$$

in $TR(\mathcal{D})$ is ill-sorted wrt $\mathcal{D}$, but is possibly sortable wrt $\mathcal{D}$. The query

$$\exists x.InCity(x, twinCity(B_1), do(load(B_1, T_1), S_0))$$

in $TR(\mathcal{D})$ is not possibly sortable wrt $\mathcal{D}$, because $twinCity(B_1)$ is not a possibly sortable term wrt $TR(\mathcal{D})$.

Now, we have the following property.

PROPERTY 24 *Assume that $W = F(\vec{t}, do([\alpha_1, \cdots, \alpha_n], S_0))$ is an atomic fluent in $\mathcal{L}_{sc}$ that is not possibly sortable with respect to $\mathcal{D}$. Then, it takes at most linear time (with respect to the length of the whole formula) to terminate reasoning $TR(\mathcal{D}) \models^{\text{ms}} W$ by checking whether W is possibly sortable and computing the corresponding $os(W)$ (which is $false$). However, in the worst-case scenario, it takes exponential time (with respect to the length of the whole formula) to determine $TR(\mathcal{D}) \models^{\text{ms}} W$ by using the usual regression in Reiter's SC.*

Note that the worst-case scenario mentioned in Property 24 often happens when a BAT is not context-free. That is, it is common that the usual regression operator leads to a regressed query whose length is exponential in the length of the original formula. Furthermore, even the corresponding $os(W)$ of any query W is not $false$, according to the previous discussion, we sometimes still can gain further computational advantages during computing $\mathcal{R}^{os}[os(W)]$ when reasoning by order-sorted regression in $\mathcal{L}^{OS}$ instead of reasoning by regression in $\mathcal{L}_{sc}$.

8 Conclusions

We propose a logical theory for reasoning about actions wrt a taxonomy of objects based on OSL. We also define a regression-based reasoning mechanism that takes advantages of sort theories, and discuss the computational advantages. It is well-known that *PDDL* supports typed (sorted) variables and many implemented planners can take advantage of types. [Classen, Eyerich, Lakemeyer, and Nebel 2007] propose formal semantics for the typed ADL subset of PDDL using ES, a dialect of SC, where types are represented using unary predicates. Our work also contributes towards a formal logical foundation of PDDL, but in a different way using order-sorted logic. A possible future work can be extending our logic to hybrid order-sorted logic [Cohn 1989; Bierle, Hedtstück, Pletat, Schmitt, and Siekmann 1992; Weidenbach 1996]. Another possibility is to consider efficient reasoning in our framework by identifying specialized classes of queries or decidable fragments [Abadi, Rabinovich, and Sagiv 2007].

References

Abadi, A., A. M. Rabinovich, and M. Sagiv [2007]. Decidable fragments of many-sorted logic. In *LPAR*, Volume 4790 of *Lecture Notes in Computer Science*, pp. 17–31. Springer.

Bacchus, F., J. Y. Halpern, and H. J. Levesque [1999]. Reasoning about noisy sensors and effectors in the situation calculus. *Artif. Intell. 111*(1-2), 171–208.

Bierle, C., U. Hedtstück, U. Pletat, P. H. Schmitt, and J. Siekmann [1992]. An order-sorted logic for knowledge representation systems. *Artif. Intell. 55*(2-3), 149–191.

Brachman, R. J. and H. J. Levesque [1982]. Competence in knowledge representation. In *AAAI*, pp. 189–192.

Brachman, R. J., H. J. Levesque, and R. Fikes [1983]. Krypton: Integrating terminology and assertion. In *AAAI*, pp. 31–35.

Classen, J., P. Eyerich, G. Lakemeyer, and B. Nebel [2007]. Towards an integration of Golog and planning. In *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*. AAAI Press.

Cohn, A. G. [1987]. A more expressive formulation of many sorted logic. *J. Autom. Reason. 3*(2), 113–200.

Cohn, A. G. [1989]. Taxonomic reasoning with many sorted logics. *Artificial Intelligence Review 3*(2-3), 89–128.

Goguen, J. A. and J. Meseguer [1987]. Remarks on remarks on many-sorted equational logic. *SIGPLAN Notices 22*(4), 41–48.

Hayes, P. J. [1971]. A logic of actions. *Machine Intelligence 6*, 495–520.

Herbrand, J. [1971]. *Logical Writings*. Cambridge: Harvard University Press. Warren D. Goldfarb (ed.).

Jouannaud, J.-P. and C. Kirchner [1991]. Solving equations in abstract algebras: A rule-based survey of unification. In *Computational Logic - Essays in Honor of Alan Robinson*, pp. 257–321. MIT Press.

Lakemeyer, G. and H. J. Levesque [2011]. A semantic characterization of a useful fragment of the situation calculus with knowledge. *Artif. Intell. 175*(1), 142–164.

Levesque, H. and G. Lakemeyer [2008]. Chapter 23 Cognitive Robotics. In V. L. Frank van Harmelen and B. Porter (Eds.), *Handbook of Knowledge Representation*, Volume 3 of *Foundations of Artificial Intelligence*, pp. 869 – 886. Elsevier.

Levesque, H., R. Reiter, Y. Lespérance, F. Lin, and R. Scherl [1997]. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming 31*, 59–84.

Levesque, H. J. [1986]. Making believers out of computers. *Artif. Intell. 30*(1), 81–108.

Levesque, H. J. [1994]. Knowledge, action, and ability in the situation calculus. In R. Fagin (Ed.), *TARK*, pp. 1–4. Morgan Kaufmann.

McSkimin, J. R. [1976]. *The use of semantic information in deductive question-answering systems. PhD Thesis*. Ph.D. thesis, University of Maryland at College Park, College Park, MD, USA.

McSkimin, J. R. and J. Minker [1977]. The use of a semantic network in a deductive question-answering system. In *IJCAI'77: Proceedings of the 5th international joint conference on Artificial intelligence*, San Francisco, CA, USA, pp. 50–58. Morgan Kaufmann Publishers Inc.

Oberschelp, A. [1962]. Untersuchungen zur mehrsortigen quantorenlogik (in German). *Mathematische Annalen 145*, 297–333.

Oberschelp, A. [1990]. Order sorted predicate logic. In *Sorts and Types in Artificial Intelligence*, Volume 418 of *Lecture Notes in Computer Science*, pp. 8–17. Springer.

Reiter, R. [1977a]. An approach to deductive question-answering. BBN Technical Report 3649 (Accession Number : ADA046550), Bolt Beranek and Newman, Inc.

Reiter, R. [1977b]. An approach to deductive question-answering systems. *SIGART Bull.* (61), 41–43.

Reiter, R. [2001]. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press.

Scherl, R. B. and H. J. Levesque [2003]. Knowledge, action, and the frame problem. *Artif. Intell. 144*(1-2), 1–39.

Schmidt, A. [1938]. Über deduktive theorien mit mehreren soften von grunddingen. *Mathematische Annalen 115*, 485–506.

Schmidt, A. [1951]. Die Zulässigkeit der Behandlung mehrsortiger Theorien mittels der üblichen einsortigen prädikatenlogik. *Mathematische Annalen 123*, 187–200.

Schmidt-Schauβ, M. [1989]. *Computational aspects of an order-sorted logic with term declarations*. New York: Springer-Verlag.

Walther, C. [1987]. *A many-sorted calculus based on resolution and paramodulation*. San Francisco: Morgan Kaufmann.

Wang, H. [1952]. Logic of many sorted theories. *Symbolic Logic 17*(2), 105–116.

Weidenbach, C. [1996]. Unification in sort theories and its applications. *Annals of Math. and AI 18*(2/4), 261–293.