A Logic of Actions Revisited

Yilan Gu
Dept. of Computer Science
University of Toronto
10 King's College Road
Toronto, ON, M5S 3G4, Canada
Email: yilan@cs.toronto.edu

Mikhail Soutchanski
Department of Computer Science
Ryerson University
245 Church Street, ENG281
Toronto, ON, M5B 2K3, Canada
Email: mes@scs.ryerson.ca

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 that takes advantage of well-sorted unification between terms. We show that answering projection queries in our logical theories is sound and complete with respect to that of in Reiter's basic action theories. 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 Introduction

In his influential paper [Hayes, 1971] titled "A Logic of Actions", Pat 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; 1989] developed expressive many-sorted logic and reviewed all previous work in this area. Reasoning about actions based on the situation calculus has been extensively developed in [Reiter, 2001]. However, he 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 situation calculus with many related sorts for objects in the domain. Perhaps, this is because mathematical proofs of these properties are not straightforward. 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. Hence, we consider formulating the situation calculus in an order-sorted (predicate) logic to describe taxonomic information about objects. We are interested in the projection problem (whether a statement is true after executing a sequence of actions) and we would like to use regression to solve this problem [Reiter, 2001]. Note that even if both many-sorted logic and order-sorted logic can be translated to unsorted, using order-sorted logic can bring about significant computational advantages, for example in deduction. This was a primary driving force for [Walther, 1987] and [Cohn, 1987]. We show that regression in 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.

It is well-known that *PDDL* supports typed (sorted) variables and many implemented planners can take advantage of types [Ghallab *et al.*, 1998]. However, to the best of our knowledge, there is no formal logical foundation for sorted reasoning in planning domains. This paper can be considered as a step towards providing this foundation.

2 Background

In general, order-sorted logic (OSL) [Oberschelp, 1962; 1990; Walther, 1987; Schmidt-Schauβ, 1989; Bierle *et al.*, 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 always 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 $\mathcal{T}$ includes 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 $i = 1..n$. 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 $f\!:\!Q_{1..n}\!\rightarrow\!Q$, 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.

For any sort theory $\mathcal{T}$, subsort relation $\leq_{\mathcal{T}}$ 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 $\mathcal{T}$ has greatest lower bound (g.l.b.), then we say that *the sort hierarchy of $\mathcal{T}$ is a meet semi-lattice* [Walther, 1987]. Moreover, a *well-sorted term* (wrt $\mathcal{T}$) is either a sorted variable, or a constant declared in $\mathcal{T}$, 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 $\mathcal{T}$. A *well-sorted atom* (wrt $\mathcal{T}$) 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 $Q'_i \leq_{\mathcal{T}} Q_i$, given that $P\!:\!\vec{Q}_{1..n}$ is in $\mathcal{T}$. A *well-sorted formula* (wrt $\mathcal{T}$) 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 *well-sorted substitution* (wrt $\mathcal{T}$) 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 $E = \{(t_{1,1}, t_{1,2}), \ldots, (t_{n,1}, t_{n,2})\}$, 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 mgu in unsorted logic (i.e., predicate logic without sorts), mgu in OSL can include new weakened variables of sorts which are subsorts of the sorts of unified terms. For example, assume that $E = \{(x, y)\}$, $x \in \mathbf{V}_{Q_1}$, $y \in \mathbf{V}_{Q_2}$ 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. 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 $\mathcal{T}$ 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 the same as in unsorted logic, but the semantics is not defined for ill-sorted terms and formulas. For any well-sorted formula ϕ, a $\mathcal{T}$-interpretation $\mathbb{I} = \langle \mathcal{M}, I \rangle$ is a tuple for a structure $\mathcal{M}$ and an assignment I from the set of free variables to the universe $\mathbf{U}$ 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 $Q_1 \leq_{\mathcal{T}} Q_2$. (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 $f\!:\!\vec{Q}_{1..n}\!\rightarrow\!Q$, $f^{\mathbb{I}} : Q_1^{\mathbb{I}} \times \cdots \times Q_n^{\mathbb{I}} \rightarrow Q^{\mathbb{I}}$ 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 $(f(\vec{t}_{1..n}))^{\mathbb{I}} \overset{def}{=} f^{\mathbb{I}}(t_1^{\mathbb{I}}, \ldots, t_n^{\mathbb{I}})$ for any well-sorted term $f(\vec{t}_{1..n})$. $\mathbb{I}$ is not defined for ill-sorted terms and formulas. (5) If $\mathcal{T}$ includes a declaration for equality symbol "$=$", then $=^{\mathbb{I}}$ must be defined as set $\{(d, d) \mid d\!\in\!\mathbf{U}\}$, i.e., the equality symbol is interpreted by the identity relation on the whole universe. For any sort theory $\mathcal{T}$ and a well-sorted formula ϕ, a structure $\mathcal{M}$ is a *$\mathcal{T}$-model* of ϕ, written as $\mathcal{M} \models^{\text{os}}_{\mathcal{T}} \phi$ iff for every $\mathcal{T}$-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, we say that a $\mathcal{T}$-interpretation $\mathbb{I} = \langle \mathcal{M}, I \rangle$ satisfies ϕ, written as $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi$, if the following conditions (1-7) hold: (1) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} P(\vec{t}_{1..n})$ iff $(t_1^{\mathbb{I}}, \ldots, t_n^{\mathbb{I}}) \in P^{\mathbb{I}}$. (2) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \neg\phi$ iff $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi$ does not hold. (3) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_1 \wedge \phi_2$ iff $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_1$ and $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_2$. (4) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_1 \vee \phi_2$ iff $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_1$ or $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_2$. (5) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi_1 \supset \phi_2$ iff $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \neg\phi_1 \vee \phi_2$. (6) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \forall x\!:\!Q.\phi$ iff for every $d\!\in\!Q^{\mathbb{I}}$, $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi[x/d]$, where $\phi[x/o]$ represent the formula obtained by substituting x with o. (7) $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \exists x : Q.\phi$ iff there is some $d \in Q^{\mathbb{I}}$ s.t. $\mathbb{I} \models^{\text{os}}_{\mathcal{T}} \phi[x/d]$. Given a sort theory $\mathcal{T}$ as the background, a theory Φ including well-sorted sentences only satisfies a well-sorted sentence ϕ, written as $\Phi \models^{\text{os}}_{\mathcal{T}} \phi$, iff every model of Φ is a model of ϕ.

Note that we follow traditional approaches to sorted reasoning, where sort symbols must not occur as predicates in the formulas and there is the closed world assumption about sorts. Alternative approaches, called hybrid, allow to mix sort symbols with application specific predicates (see [Weidenbach, 1996; Cohn, 1989; Bierle *et al.*, 1992]).

Due to the space limitations, we skip the background of the situation calculus. Details can be found in [Reiter, 2001] and we refer to this language as Reiter's situation calculus below. Note that in this paper, we use $\models^{\text{os}}_{\mathcal{T}}$ to represent the logical entailment wrt a sort theory $\mathcal{T}$ in order-sorted logic, $\models^{\text{ms}}$ to represent the logical entailment in Reiter's situation calculus (a many-sorted logic with one standard sort *Object*), and $\models^{\text{fo}}$ to represent the logical entailment in unsorted predicate logic.

3 An Order-Sorted Situation Calculus

In this paper, we consider a modified situation calculus based on order-sorted logic, called *order-sorted situation calculus* 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.

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

• $\mathcal{T}_{\mathcal{D}}$ 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 $Q_i \leq_{\mathcal{T}} Object$ 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 $Q_i \leq_{\mathcal{T}} Object$ 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 $Q_i \leq_{\mathcal{T}} Object$ and $Q_i \neq \bot$ for $i = 1..n$. Note that, when $n = 0$, the declaration is of form $A : Act$ 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 $Q_i \leq_{\mathcal{T}} Object$ and $Q_i \neq \bot$ for each $i = 1..(n+1)$. Note that, when $n = 0$, it is a function declaration for a constant, denoted as $c : Q$ 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.

• **D** is a set of axioms represented using well-sorted sentences wrt $\mathcal{T}_{\mathcal{D}}$, 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. In particular, it includes the unique name axioms for object contants. For clarity, it also includes finitely many *axioms of disjointness for basic sorts* of the form $\forall x : Q_i . \forall y : Q_j .(x \neq y)$ for all distinct *basic sorts* Q_i and Q_j, where Q_i, Q_j are considered basic sorts if $\bot \leq Q_i$ and $\bot \leq Q_j$ are in $\mathcal{T}_{\mathcal{D}}$, and there are no sorts $Q' \neq \bot$, $Q'' \neq \bot$, such that $Q' \leq Q_i$ and $Q'' \leq Q_j$. Notice that these conditions actually are consequences of the semantics of the subsort declarations in the sort theory $\mathcal{T}_{\mathcal{D}}$.

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 of the form

$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, s : Sit).Poss(A(\vec{x}_{1..n}), s) \equiv \phi_A(\vec{x}_{1..n}, s), \quad (1)$$

where $\phi_A(\vec{x}_{1..n}, s)$ 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

$$(\forall \vec{x}_{1..n} : \vec{Q}_{1..n}, a : Act, s : Sit).$$
$$F(\vec{x}_{1..n}, do(a, s)) \equiv \psi_F(\vec{x}_{1..n}, a, s), \quad (2)$$

where $\psi_F(\vec{x}_{1..n}, a, s)$ 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. $\mathcal{T}_{\mathcal{D}}$ includes following subsort declarations:

$MovObj \leq Object$, $\bot \leq City$, $\bot \leq Box$, $\bot \leq Truck$,
$Truck \leq MovObj$, $City \leq Object$, $Box \leq MovObj$,

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 $On(o, t, s)$. 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:

$B_1 : Box$,	$B_2 : Box$,	$T_1 : Truck$,
$T_2 : Truck$,	$Pasadena : City$,	$Boston : City$.

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.

4 Order-Sorted Regression and Reasoning

We now consider the central reasoning mechanism in the order-sorted situation calculus. 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 $\mathcal{T}_{\mathcal{D}}$; (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 would like to 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. 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
$\mathcal{R}^{os}[\circ W_1] = \circ\mathcal{R}^{os}[W_1]$ 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.

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

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

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 1). Here are some examples.

Example 3 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 $\mathcal{D}$).

Given an order-sorted BAT $\mathcal{D} = (\mathcal{T}_{\mathcal{D}}, \mathbf{D})$ 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 that of in [Reiter, 2001].

Theorem 1 *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, $\mathbf{D} \models^{os}_{\mathcal{T}_{\mathcal{D}}} W \equiv \mathcal{R}^{os}[W]$.*

Theorem 2 *If W is a regressable formula wrt $\mathcal{D}$, then*
$$\mathbf{D} \models^{os}_{\mathcal{T}_{\mathcal{D}}} W \text{ iff } \mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models^{os}_{\mathcal{T}_{\mathcal{D}}} \mathcal{R}^{os}[W].$$

Hence, to reason whether $\mathbf{D} \models^{os}_{\mathcal{T}_{\mathcal{D}}} W$ is the same as to compute $\mathcal{R}^{os}[W]$ first and then to reason whether $\mathcal{D}_{S_0} \cup \mathcal{D}_{una} \models^{os}_{\mathcal{T}_{\mathcal{D}}} \mathcal{R}^{os}[W]$. Besides, according to Theorem 1, it is easy to see that the following consequence holds.

Corollary 1 *If W_1 and W_2 are regressable formulas in $\mathcal{L}^{OS}$ s.t. $\models^{os}_{\mathcal{T}_{\mathcal{D}}} W_1 \equiv W_2$, then $\mathbf{D} \models^{os}_{\mathcal{T}_{\mathcal{D}}} \mathcal{R}^{os}[W_1] \equiv \mathcal{R}^{os}[W_2]$.*

Intuitively, Corollary 1 states that the regressed results of two logically equivalent regressable formulas (possibly having different syntactic forms only) are still equivalent.

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

Although BATs and regressable formulas in $\mathcal{L}^{OS}$ are based on OSL, they can be related to BATs and regressable formulas in Reiter's situation calculus as stated in Theorem 3.

Theorem 3 (Soundness) *For any BAT $\mathcal{D}$ and any query W in order-sorted situation calculus $\mathcal{L}^{OS}$, there exists a corresponding BAT $\mathcal{D}'$ and a corresponding query W' in Reiter's situation calculus s.t.*
$$\mathbf{D} \models^{os}_{\mathcal{T}_{\mathcal{D}}} W \text{ iff } \mathcal{D}' \models^{ms} W'.$$

Intuitively, we would like to show that the order-sorted situation calculus $\mathcal{L}^{OS}$ is correct, or *sound*, in the sense that for any query in $\mathcal{L}^{OS}$ that can be answered in its background BAT in $\mathcal{L}^{OS}$, we always can find a way to represent the BAT and the query in Reiter's situation calculus $\mathcal{L}_{sc}$ s.t. the corresponding query in $\mathcal{L}_{sc}$ can be answered wrt the corresponding BAT in $\mathcal{L}_{sc}$.

It is hard to prove Theorem 3 directly. Inspired by the *standard relativization* of OSL to unsorted (first-order) logic, our general idea of proving Theorem 3 is as follows. In Step 1, we prove that there is an unsorted theory $\mathcal{D}''$ (via *strong relativization*) and an unsorted first-order sentence W'' (via *relativization*) s.t. $\mathbf{D} \models^{\mathrm{os}}_{\mathcal{T}_\mathcal{D}} W$ iff $\mathcal{D}'' \models^{\mathrm{fo}} W''$. In Step 2, we construct a BAT $\mathcal{D}'$ (called the *corresponding Reiter's BAT of* $\mathcal{D}$ below) and a regressable formula W' (called the *translation of* W below) in Reiter's situation calculus, s.t. $\mathcal{D}' \models^{\mathrm{ms}} W'$ iff $\mathcal{D}''' \models^{\mathrm{fo}} W'''$, for some unsorted theory $\mathcal{D}'''$ (via standard relativization) and sentence W''' (via relativization). Finally, in Step 3, we show that $\mathcal{D}''' \models^{\mathrm{fo}} W'''$ iff $\mathcal{D}'' \models^{\mathrm{fo}} W''$.

$$\begin{tikzcd}
\mathbf{D} \models^{\mathrm{os}}_{\mathcal{T}_\mathcal{D}} W \arrow[r, Leftrightarrow, "\text{(Step 1)}"] & \mathcal{D}'' \models^{\mathrm{fo}} W'' \arrow[d, Leftrightarrow, "\text{(Step 3)}"] \\
\mathcal{D}' \models^{\mathrm{ms}} W' \arrow[r, Leftrightarrow, "\text{(Step 2)}"] & \mathcal{D}''' \models^{\mathrm{fo}} W'''
\end{tikzcd}$$

Fig 1. Diagram of the Outline for Proving Theorem 3

To prove Theorem 3, we first define some concepts and prove Lemma 1 for later convenience. First, 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 $\mathcal{L}^{OS}$.

Definition 1 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 all formulas in $\mathcal{L}_{sc}$ are well-sorted wrt the sort theory of $\mathcal{L}_{sc}$. Hence, the definition of rel can also be applied to any formula or a set of formulas in Reiter's situation calculus.

Definition 2 For any sort theory $\mathcal{T}_\mathcal{D}$ in $\mathcal{L}^{OS}$, *the set of bridge axioms* of $\mathcal{T}_\mathcal{D}$, $BA(\mathcal{T}_\mathcal{D})$, is a set of the following formulas:
(a) $(\forall x).\, Q_2(x) \supset Q_1(x)$ for each $Q_2 \leq Q_1 \in \mathcal{T}_\mathcal{D}$;
(b) $Q(c)$ for each $c\!:\!Q \in \mathcal{T}_\mathcal{D}$;
(c) $(\forall \vec{x}_{1..n}).\, \bigwedge_{i=1}^{n} Q_i(x_i) \supset Q(f(\vec{x}_{1..n}))$ for each $f : \vec{Q}_{1..n} \rightarrow Q \in \mathcal{T}_\mathcal{D}$.

Moreover, let $Sorted(x)$ be an auxiliary predicate that does not appear in $\mathcal{D}$: it is a purely technical device used for proving Theorem 3. The set of *strong bridge axioms* of $\mathcal{T}_\mathcal{D}$, $SBA(\mathcal{T}_\mathcal{D})$, is also a set of unsorted axioms $BA(\mathcal{T}_\mathcal{D}) \cup sba(\mathcal{T}_\mathcal{D})$, where $sba(\mathcal{T}_\mathcal{D})$ includes the following axioms:
(d) $(\forall \vec{x}_{1..n}).P(\vec{x}_{1..n}) \supset \bigwedge_{i=1}^{n} Q_i(x_i) \wedge Sorted(x_i)$ for each $P\!:\!\vec{Q}_{1..n} \in \mathcal{T}_\mathcal{D}$;
(e) $(\forall \vec{x}_{1..n}).Q(f(\vec{x}_{1..n})) \wedge Sorted(f(\vec{x}_{1..n})) \supset \bigwedge_{i=1}^{n}(Q_i(x_i) \wedge Sorted(x_i))$ for each $f\!:\!\vec{Q}_{1..n} \rightarrow Q \in \mathcal{T}_\mathcal{D}$.

Intuitively, $Sorted(t)$ means that term t is well-sorted (wrt $\mathcal{D}$). (a functional term is well-sorted and of its own sort, respectively), then all its arguments should be well-sorted and of the corresponding sorts wrt the predicate declaration (the function declaration, respectively). Note that although $Sorted$ may satisfy other characterizing axioms than axioms in (d) and (e) according to its intuitive meaning, but adding axioms in (d) and (e) to the strong relativization theory of $\mathcal{D}$ defined below is enough for us to prove Theorem 3.

Definition 3 For any order-sorted BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$, *the strong relativization of* $\mathcal{D}$, an unsorted theory, is defined as
$$REL_S(\mathcal{D}) \overset{def}{=} rel(\mathbf{D}) \cup SBA(\mathcal{T}_\mathcal{D}).$$
Consider any BAT $\mathcal{D}_1$ in Reiter's situation calculus $\mathcal{L}_{sc}$, which has a finite set $\mathcal{T}_{\mathcal{D}_1}$ of function declarations and predicate declarations for all predicates and functions appeared in $\mathcal{D}_1$. The *standard relativization of* $\mathcal{D}_1$, an unsorted theory, is defined as
$$REL(\mathcal{D}_1) \overset{def}{=} rel(\mathcal{D}_1) \cup BA(\mathcal{T}_{\mathcal{D}_1}).$$

The reasons for differences between the two cases in Def. 3 are that (1) we include the sort theory in each BAT of order-sorted situation calculus, while Reiter's situation calculus mentions sort declarations generally in the signature of $\mathcal{L}_{sc}$, and (2) we need strong relativization for order-sorted BATs and only need standard relativization for Reiter's BATs to prove Theorem 3. In comparison to the standard relativization, the strong relativization adds additional axioms of the form (d) and (e) in Def. 2. They are based on the sort theory that includes one and only one declaration for each predicate P or for each function f, respectively. We can also prove a relativization theorem as follows for the strong relativization similar to the Sort Theorem proved in [Walther, 1987] and/or the relativization theorem proved in [Schmidt-Schauβ, 1989].

Lemma 1 *Consider any regressable formula W with a background BAT $\mathcal{D}$ in order-sorted situation calculus $\mathcal{L}^{OS}$. Then,*
$$\mathbf{D} \models^{\mathrm{os}}_{\mathcal{T}_\mathcal{D}} W \textit{ iff } REL_S(\mathcal{D}) \models^{\mathrm{fo}} rel(W).$$

We therefore can prove Step 1 in Fig. 1 using Lemma 1. Because Reiter's situation calculus is a many-sorted logical language with special 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. To construct a Reiter's BAT $\mathcal{D}'$ and a regressable formula W' that satisfy the theorem, we first define another translation function $tr(W)$ as follows.

Definition 4 Consider any well-sorted formula ϕ in $\mathcal{L}^{OS}$. A *translation* of ϕ to a (well-sorted) sentence in Reiter's situation calculus, denoted as $tr(\phi)$, is defined recursively as follows:

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

The translation function tr defined above is a mapping from well-sorted formulas wrt the sort theory of some BAT $\mathcal{D}$ (or, wrt $\mathcal{D}$ for simplicity) in $\mathcal{L}^{OS}$ to well-sorted formulas in $\mathcal{L}_{sc}$. Moreover, it is easy to prove by structural induction the following lemma for rel and tr, which will be useful for proving Theorem 3.

Lemma 2 *Consider any well-sorted formula ϕ in $\mathcal{L}^{OS}$. Then,* $\models^{\mathrm{fo}} rel(tr(\phi)) \equiv rel(\phi)$.

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 Theorem 3. 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 situation calculus below, all free variables are implicitly universally sorted-quantified according to their obvious sorts. The function and predicate declarations are always standard, hence are not mentioned here.

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

• The initial theory of $TR(\mathcal{D})$, say $\mathcal{D}'_{S_0}$, includes the following axioms. Note that for axioms in items **(3)**–**(5)** below, predicate $Sorted$ is auxiliary wrt $\mathcal{D}$ and each x_i is universally quantified with a default sort $Object$ (Q_i itself, respectively) if $Q_i \leq_{\mathcal{T}} Object$ ($Q_i \not\leq_{\mathcal{T}} Object$, respectively).

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 $\mathcal{T}_{\mathcal{D}}$, 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 each declaration $f : \vec{Q}_{1..n} \rightarrow Q$ in $\mathcal{T}_{\mathcal{D}}$ ($n \geq 1$), add an axiom $tr((\forall \vec{x}_{1..n} : \vec{Q}_{1..n}).(\exists y : Q).y = f(\vec{x}_{1..n}))$.

We also add an axiom

$$Q(f(\vec{x}_{1..n})) \wedge Sorted(f(\vec{x}_{1..n})) \supset$$
$$tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \textstyle\bigwedge_{i=1}^{n} (x_i = y_i \wedge Sorted(x_i)))$$

if $Q \leq_{\mathcal{T}} Object$ and $Q \neq Object$, or add an axiom

$$((\exists y : Q).y = f(\vec{x}_{1..n}) \wedge Sorted(y)) \supset$$
$$tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \textstyle\bigwedge_{i=1}^{n} (x_i = y_i \wedge Sorted(x_i)))$$

otherwise.

4. For each situation-independent predicate declaration $P : \vec{Q}_{1..n}$, add an axiom
$P(\vec{x}_{1..n}) \supset tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \bigwedge_{i=1}^{n} (x_i = y_i \wedge Sorted(x_i)))$.

5. For each fluent declaration $F : \vec{Q}_{1..n} \times Sit$, add an axiom
$F(\vec{x}_{1..n}, S_0) \supset tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \bigwedge_{i=1}^{n} (x_i = y_i \wedge Sorted(x_i)))$.

6. For any constant declaration $c : Q$ where $Q \leq_{\mathcal{T}} Object$ and $Q \neq Object$, add an axiom $Q(c)$. Note that other constant declarations will still be kept in the sort theory of $\mathcal{L}_{sc}$ by default (e.g., $S_0 : Sit$).

• 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 situation calculus:

$$Poss(A(\vec{x}_{1..n}), s) \quad \equiv \quad \phi'_A(\vec{x}_{1..n}, s) \tag{3}$$

where $\phi'_A(\vec{x}_{1..n}, s)$ is a $\mathcal{L}_{sc}$ formula uniform in s, resulting from $tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}).(\bigwedge_{i=1}^{n} x_i = y_i) \wedge \phi_A(\vec{y}_{1..n}, s))$. Here, all y_i's are distinct auxiliary variables never appearing in $\phi_A(\vec{x}_{1..n}, s)$.

• 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 situation calculus:

$$F(\vec{x}_{1..n}, do(a, s)) \quad \equiv \quad \psi'_F(\vec{x}_{1..n}, a, s) \tag{4}$$

where $\psi'_F(\vec{x}_{1..n}, a, s)$ is a $\mathcal{L}_{sc}$ formula uniform in s, resulting from $tr((\exists \vec{y}_{1..n} : \vec{Q}_{1..n}). \bigwedge_{i=1}^{n} x_i = y_i \wedge \psi_F(\vec{y}_{1..n}, a, s))$. Here, all y_i's are distinct auxiliary variables never appearing in $\psi_F(\vec{x}_{1..n}, s)$.

Let $\mathcal{D}' = TR(\mathcal{D})$, $W' = tr(W)$, we then can prove Theorem 3 by following the ideas presented in Fig. 1. Details are omitted due to the space limitations.

Example 4 Consider the BAT $\mathcal{D}$ from Example 1. The axioms in $TR(\mathcal{D})$ are mostly obvious. Due to the space limitations, we just provide examples of a precondition axiom and an SSA in $TR(\mathcal{D})$:

$$\begin{aligned}
&Poss(load(x,t),s) \equiv Box(x) \wedge Truck(t) \wedge \neg On(x,t,s) \wedge \\
&\qquad (\exists y.City(y) \wedge InCity(x,y,s) \wedge InCity(t,y,s)), \\
&InCity(d,c,do(a,s)) \equiv MovObj(d) \wedge City(c) \wedge \\
&\qquad [(\exists t, c_1.Truck(t) \wedge City(c_1) \wedge a = drive(t,c_1,c) \\
&\qquad \wedge (d = t \vee \exists b.Box(b) \wedge b = d \wedge On(b,t,s))) \\
&\qquad \vee InCity(d,c,s) \wedge \\
&\qquad \neg(\exists t, c_1.Truck(t) \wedge City(c_1) \wedge a = drive(t,c,c_1) \\
&\qquad \wedge (d = t \vee \exists b.Box(b) \wedge b = d \wedge On(b,t,s)))].
\end{aligned}$$

It is important to notice that all queries $\mathcal{L}^{OS}$ have to be well-sorted wrt the given background order-sorted BAT $\mathcal{D}$; while, in general, the queries that can be answered in the corresponding Reiter's BAT of $\mathcal{D}$ are not necessarily well-sorted wrt $\mathcal{D}$. Below, Theorem 4 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.

Theorem 4 (Completeness) *Let $\mathcal{D}$ be an order-sorted BAT in $\mathcal{L}^{OS}$, and $TR(\mathcal{D})$ be its corresponding Reiter's BAT. Then, for any query W in Reiter's situation calculus, W can be translated to a (well-sorted) query wrt $\mathcal{D}$, denoted as $os(W)$ below, s.t. $TR(\mathcal{D}) \models^{\mathrm{ms}} tr(os(W)) \equiv W$. Furthermore, we have $TR(\mathcal{D}) \models^{\mathrm{ms}} W$ iff $\mathbf{D} \models^{\mathrm{os}}_{\mathcal{T}_{\mathcal{D}}} os(W)$.*

To prove Theorem 4, we first define some new concepts and prove a lemma.

Definition 5 Let $\mathcal{D}$ be a BAT in the order-sorted situation calculus $\mathcal{L}^{OS}$, and $TR(\mathcal{D})$ be its corresponding Reiter's BAT. Any term t in Reiter's situation calculus 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 $\mathcal{T}_{\mathcal{D}}$ (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 $\mathcal{T}_{\mathcal{D}}$, for every i ($i = 1..n$), t_i either is a variable or is a non-variable term of sort Q'_i wrt $\mathcal{D}$ and $Q'_i \leq_{\mathcal{T}} Q_i$ in $\mathcal{T}_{\mathcal{D}}$ (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 situation calculus (can be $t_1 = t_2$), which is well-sorted wrt $TR(\mathcal{D})$, is a *possibly sortable atom wrt* $\mathcal{D}$, if for every i, t_i either is a variable or is a non-variable term s.t.:
(a) it is possibly sortable wrt $\mathcal{D}$; and
(b) $P : \vec{Q}_{1..n}$ is in $\mathcal{T}_{\mathcal{D}}$ ($=: \top \times \top$, respectively), the sort of t_i is Q'_i wrt $\mathcal{D}$ and $Q'_i \leq_{\mathcal{T}} Q_i$ wrt $\mathcal{D}$.

Given any $\mathcal{D}$ in order-sorted situation calculus, 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 $\mathcal{D}$.

Lemma 3 *Let $\mathcal{D}$ be a BAT in the order-sorted situation calculus $\mathcal{L}^{OS}$, and $TR(\mathcal{D})$ be its corresponding Reiter's BAT. Then, for any atom $P(\vec{t}_{1..n})$ (can be $t_1 = t_2$) that is well-sorted in $\mathcal{L}_{sc}$ but not possibly sortable wrt $\mathcal{D}$, we have $TR(\mathcal{D}) \models^{\rm ms} P(\vec{t}_{1..n}) \equiv false$.*

Now we define a function which transforms a formula in $\mathcal{L}_{sc}$ wrt $TR(\mathcal{D})$ to a well-sorted formula in $\mathcal{L}^{OS}$ wrt $\mathcal{D}$.

Definition 6 Let $\mathcal{D}$ be a BAT in the order-sorted situation calculus $\mathcal{L}^{OS}$, $TR(\mathcal{D})$ be its corresponding Reiter's BAT and W be a regressable sentence in $\mathcal{L}_{sc}$ wrt the background BAT $TR(\mathcal{D})$. Then, function $os(W)$ is defined recursively as follows.

1. If W is either of the form $(\forall x)W_1$, $(\exists x)W_1$, where the default sort of x is Q (either $Object$, Act or Sit) in $TR(\mathcal{D})$, then $os((\forall x)W_1) \overset{def}{=} (\forall x : Q)os(W_1)$, and $os(\exists x.W_1) \overset{def}{=} (\exists x\!:\!Q)os(W_1)$.
2. If W is one of the form $\neg W_1$, $W_1 \wedge W_2$, $W_1 \vee W_2$, then $os(\neg W_1) \overset{def}{=} \neg os(W_1)$, $os(W_1 \wedge W_2) \overset{def}{=} os(W_1) \wedge os(W_2)$, $os(W_1 \vee W_2) \overset{def}{=} os(W_1) \vee os(W_2)$.
3. If W is atomic and not possibly sortable, then $W \overset{def}{=} false$.
4. If W is atomic and possibly sortable, assume that $var(W) = \langle x_1, \cdots, x_n \rangle$ is the vector of free variables appeared from left to right in W (including repeated ones). For each $i = 1..n$, suppose that x_i appears as an argument of a function f_i in some term or as an argument of a predicate P_i in W. Let Q_i be the sort appeared in the k_i-th position of the declaration of f_i (P_i, respectively), if x_i appears in the k_i-th position of f_i (P_i, respectively) in W. Then, let $I_W = \{i \,|\, x_i \in var(W), Q_i \leq_{\mathcal{T}} Object, Q_i \neq Object\}$, and $\vec{y}\!:\!\vec{Q} = \{y_i\!:\!Q_i \,|\, i \in I_W\}$, where y_i's are auxiliary variables never appeared in W and each y_i is distinct from others. And, $os(W) \overset{def}{=} (\exists \vec{y} : \vec{Q})(W_0 \wedge \bigwedge_{i \in I_W} x_i = y_i)$, where W_0 is obtained from substituting each x_i with y_i for $i \in I_W$.

Proof sketch for Theorem 4. First, for any query W in Reiter's situation calculus, let $W' = os(W)$. By using structural induction and Lemma 3, it is easy to prove that W' is a well-sorted query wrt $\mathcal{D}$ in OSL and $TR(\mathcal{D}) \models^{\rm ms} W \equiv tr(W')$. Then, by Theorem 3 and $TR(\mathcal{D}) \models^{\rm ms} W \equiv tr(W')$, it is easy to see that $\mathbf{D} \models^{\rm os}_{\mathcal{T}_\mathcal{D}} W'$ iff $TR(\mathcal{D}) \models^{\rm ms} tr(W')$ iff $TR(\mathcal{D}) \models^{\rm ms} W$. Proof details are omitted due to the space limitations. But, we provide some examples below to illustrate the statement.

Example 5 Here are simple examples of computing $os(W)$ from W in $\mathcal{L}_{sc}$. Consider the $TR(\mathcal{D})$ in Example 4. Let $On(Boston, T_1, S_1)$ (denoted as W_3) be a query in $\mathcal{L}_{sc}$, where S_1 is some situation instance. According to the way $TR(\mathcal{D})$ is constructed, we have $TR(\mathcal{D}) \models^{\rm ms} On(o, t, s) \supset Box(o)$ and $TR(\mathcal{D}) \models^{\rm ms} \neg Box(Boston)$. So, $TR(\mathcal{D}) \models^{\rm ms} W_3 \equiv false$. Hence, $os(W_3) \overset{def}{=} false$.
Let W_4 be $\forall s.\exists o.\neg InCity(o, Pasadena, s)$, which is also a query in $\mathcal{L}_{sc}$, where $o : Object$ and $s : Sit$ hold by default. Then, $os(W_4)$ is $\forall s : Sit.\exists o : Object.\neg(\exists b : MovObj.b = o \wedge InCity(b, Pasadena, s))$, since $TR(\mathcal{D}) \models^{\rm ms} InCity(o, c, s) \supset MovObj(o) \wedge City(c)$. And it is easy to prove that $TR(\mathcal{D}) \models^{\rm ms} W_4 \equiv tr(os(W_4))$.

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

In this section, we discuss the advantages of using OSL and the order-sorted regression operator based on it.

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 $\mathbf{D} \models^{\rm os}_{\mathcal{T}_\mathcal{D}} W \equiv \mathcal{R}[W]$. 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 5 *Consider any regressable sentence W with a background BAT $\mathcal{D}$ in order-sorted situation calculus $\mathcal{L}^{OS}$. 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 the corresponding Reiter's BAT $TR(\mathcal{D})$.*

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 the following property.

Theorem 6 *Let a regressable formula W have the syntactic form $t_{1,1} = t_{1,2} \wedge \ldots \wedge t_{m,1} = t_{m,2} \wedge W_1$, with any background order-sorted BAT $\mathcal{D}$ in $\mathcal{L}^{OS}$. 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, then Computing $\mathcal{R}^{os}[W]$ runs in time $O(n)$, while computing $\mathcal{R}[W]$ wrt $\mathcal{D}$ ($\mathcal{R}[tr(W)]$ wrt $TR(\mathcal{D})$) 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 in $O(2^n)$.*

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 save computation time significantly.

Example 6 Consider the BAT $\mathcal{D}$ from Example 1. Let W_5 be a $\mathcal{L}^{OS}$ query (i.e., a (well-sorted) regressable sentence) $InCity(T_1, Pasadena, do(drive(T_1, Boston, Pasadena), S_1))$, 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 Theorem 4. But, reasoning whether $\mathbf{D} \models^{\rm os}_{\mathcal{T}_\mathcal{D}} os(W)$ (starting from finding $os(W)$) sometimes can terminate earlier than finding whether $TR(\mathcal{D}) \models^{\rm ms} W$. In particular, we have

Theorem 7 *Assume that $W = F(\vec{t}, do([\alpha_1, \cdots, \alpha_n], S_0))$ is an atomic fluent instance in $\mathcal{L}_{sc}$ that includes an ill-sorted ground term wrt $\mathcal{D}$ (e.g., W_3 in Example 5). Then, it takes at most linear time to terminate reasoning by computing the corresponding $os(W)$ (which is $false$).*

Observe that reasoning about $TR(\mathcal{D}) \models^{\rm ms} W$ directly, for the formula W mentioned in Theorem 7, using regression $\mathcal{R}$ could result in a exponentially large regression tree when computing $\mathcal{R}[W]$. Also, the size of the resulting formula can be exponentially larger than that of W. Moreover, it still needs further computational steps to find whether $TR(\mathcal{D})_{S_0} \cup TR(\mathcal{D})_{una} \models^{\rm ms} \mathcal{R}[W]$.

7 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 of our theory. One possible future work can be extending our logic to hybrid order-sorted logic [Cohn, 1989; Bierle *et al.*, 1992; Weidenbach, 1996]. Another possibility is to consider efficient reasoning in our framework by identifying specialized classes of queries or decidable fragments [Abadi *et al.*, 2007]. Finally, we are planning to work on an efficient implementation of our theory.

References

[Abadi *et al.*, 2007] Aharon Abadi, Alexander Moshe Rabinovich, and Mooly Sagiv. Decidable fragments of many-sorted logic. In *LPAR*, volume 4790 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2007.

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

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

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

[Ghallab *et al.*, 1998] M. Ghallab, a. Howe, C. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL the planning domain definition language. Technical report, Yale Center for Computational Vision and Control, Technical Report CVC TR-98-003/DCS TR-1165, 1998.

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

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

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

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

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

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

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

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

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

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

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

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