Definability and Process Ontologies

Michael Grüninger
Department of Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario, Canada
gruninger@mie.utoronto.ca

Abstract

In this paper, we use the notions of relative interpretations and definable models from mathematical logic to compare different ontologies and also to evaluate the limitations of particular ontologies. In particular, we characterize the relationship between the theories within the first-order PSL Ontology and two other ontologies – a first-order theory of time and Reiter's second-order axiomatization of situation calculus.

1 Introduction

Representing activities and the constraints on their occurrences is an integral aspect of commonsense reasoning, particularly in manufacturing, enterprise modelling, and autonomous agents or robots. There have been a variety of process ontologies developed within the artificial intelligence community, particularly in the context of robotics and planning systems.

In this paper, we use the notions of relative interpretations and definable models to compare different process ontologies and also to evaluate the limitations of particular ontologies. In particular, we characterize the relationship between the theories within the first-order PSL Ontology and two other ontologies – a first-order theory of time and Reiter's second-order axiomatization of situation calculus. There are two major kinds of results – relative interpretation theorems that show the conditions under which two ontologies are equivalent, and nondefinability theorems which show that one ontology is in some sense stronger since it is able to define concepts that other ontologies cannot define.

2 Relationships between Theories

Different ontologies within the same language can be compared using the notions of satisfiability, entailment, and independence. More difficult is to compare ontologies that are axiomatized in different languages; in such cases, we need to determine whether or not the lexicon of one ontology can be interpreted in the lexicon of the other ontology. In this section, we review the basic concepts from model theory that will supply us with the techniques for comparing ontologies in different languages.

2.1 Relative Interpretations of Theories

We will adopt the following definition from (Enderton 1972):

Definition 1 *An interpretation π of a theory T_0 with language L_0 into a theory T_1 with language L_1 is a function on the set of parameters of L_0 such that*

1. *π assigns to $\forall$ a formula $\pi_\forall$ of L_1 in which at most v_1 occurs free, such that*
$$T_1 \models (\exists v_1)\, \pi_\forall$$
2. *π assigns to each n-place relation symbol P a formula π_P of L_1 in which at most the variables $v_1, ..., v_n$ occur free.*
3. *π assigns to each n-place function symbol f a formula π_f of L_1 in which at most the variables $v_1, ..., v_n, v_{n+1}$ occur free, such that*
$$T_1 \models (\forall v_1, ..., v_n)\, \pi_\forall(v_1) \wedge ... \wedge \pi_\forall(v_n)$$
$$\supset (\exists x)(\pi_\forall(x) \wedge ((\forall v_{n+1})(\pi_f(v_1, ..., v_{n+1}) \equiv (v_{n+1} = x))))$$
4. *For any sentence σ in L_0,*
$$T_0 \models \sigma \Rightarrow T_1 \models \pi(\sigma)$$

2.2 Definable Interpretations

Relative interpretations specify mappings between theories; we are also interested in specifying mappings between models of the theories. Such an approach will also provide with a means of proving that no relative interpretation exists between two particular theories.

We begin with the notion of definable sets within a structure.

Definition 2 *Let $\mathcal{M}$ be a structure in a language L.*

A set $X \subseteq M^n$ is definable in $\mathcal{M}$ iff there is a formula $\varphi(v_1, ..., v_n, w_1, ..., w_m)$ of L and $\overline{\mathbf{b}} \in M^m$ such that
$$X = \{\overline{\mathbf{a}} \in M^n \,:\, \mathcal{M} \models \varphi(\overline{a}, \overline{b})\}$$

X is A-definable if there is a formula $\psi(\overline{v}, w_1, ..., w_l)$ and $\overline{\mathbf{b}} \in A^l$ such that
$$X = \{\overline{\mathbf{a}} \in M^n \,:\, \mathcal{M} \models \varphi(\overline{\mathbf{a}}, \overline{\mathbf{b}})\}$$

Using this definition, we can adopt the following approach from (Marker 2002):

Definition 3 *Let $\mathcal{N}$ be a structure in $\mathcal{L}_0$ and let $\mathcal{M}$ be a structure in $\mathcal{L}$. We say that $\mathcal{N}$ is definable in $\mathcal{M}$ iff we can find a definable subset X of M^n and we can interpret the symbols of $\mathcal{L}_0$ as definable subsets and functions on X so that the resulting structure in $\mathcal{L}_0$ is isomorphic to $\mathcal{N}$.*

The relationship between relative interpretations of theories and definable interpretations of structures is captured in a straightforward way by the following proposition:

Proposition 1 *If there exists an interpretation of T_1 into T_2, then every model of T_1 is definable in some model of T_2.*

Our primary tool for proving that the models of one ontology are not definable in the models of another ontology will be the following proposition from (Marker 2002):

Proposition 2 *Let $\mathcal{M}$ be a structure. If $X \subset M^n$ is A-definable, then every automorphism of $\mathcal{M}$ that fixes the set A pointwise fixes X setwise (that is, if σ is an automorphism of $\mathcal{M}$ and $\sigma(a) = a$ for all $a \in A$, then $\sigma(X) = X$).*

Using this proposition, we can show that a relation is not definable in some structure if there exists an automorphism of the structure that does not preserve the relation.

3 Definability and Time Ontologies

3.1 Linear Time with Endpoints

Consider the ontology $T_{linear-time}$[1] of linear time without endpoints (Hayes 1996). The countable models of this ontology are isomorphic to countably infinite linear orderings with no initial or final element.

Lemma 1 *Let $\mathcal{T}$ be a model of T_{linear_time} that is either discrete or dense.*

The set of automorphisms $Aut(\mathcal{T})$ does not fix any timepoints.

Proof: A model $\mathcal{T}$ of T_{linear_time} is discrete iff it contains a subordering that isomorphic to $\mathbb{Z}$, and $Aut(\mathbb{Z})$ does not fix any elements of $\mathbb{Z}$.

A model $\mathcal{T}$ of T_{linear_time} is dense iff it contains a subordering that is isomorphic to $\mathbb{Q}$, and $Aut(\mathbb{Q})$ does not fix any elements of $\mathbb{Q}$. □

In other words, any two timepoints in $\mathcal{T}$ can be mapped each other by some automorphism of $\mathcal{T}$, whenever $\mathcal{T}$ is either discrete or dense.

[1]The axioms for $T_{linear-time}$ in CLIF (Common Logic Interchange Format) can be found at `http://www.stl.mie.utoronto.ca/colore/linear-time.clif`

3.2 Relationship to PSL-Core

The purpose of PSL-Core ((Gruninger 2004), (Bock & Gruninger 2005)) is to axiomatize a set of intuitive semantic primitives that are adequate for describing the fundamental concepts of manufacturing processes. Consequently, this characterization of basic processes makes few assumptions about their nature beyond what is needed for describing those processes, and it is therefore rather weak in terms of logical expressiveness.

Within PSL-Core [2], there are four kinds of entities required for reasoning about processes – activities, activity occurrences, timepoints, and objects. Activities may have multiple occurrences, or there may exist activities which do not occur at all. Timepoints are linearly ordered, forwards into the future, and backwards into the past. Finally, activity occurrences and objects are associated with unique timepoints that mark the begin and end of the occurrence or object.

Lemma 2 *A model of $T_{pslcore}$ in which the ordering over timepoints is either discrete or dense is not definable in any model of $T_{linear-time}$.*

Proof: Let $\mathcal{T}$ be a model of T_{linear_time} and let $\mathcal{M}$ be a model of $T_{pslcore}$ in which the ordering over timepoints is either discrete or dense.

By Lemma 1, the set of automorphisms $Aut(\mathcal{T})$ does not fix any timepoints, so that for any timepoint $\mathbf{t}$ there exists $\varphi \in Aut(\mathcal{T})$ such that $\varphi(\mathbf{t}) \neq \mathbf{t}$.

Since **beginof** is a function, activity occurrences have unique beginning timepoints, so that we have

$$\langle \mathbf{o}, \mathbf{t} \rangle \in \mathbf{beginof} \Rightarrow \langle \mathbf{o}, \varphi(\mathbf{t}) \rangle \notin \mathbf{beginof}$$

By Proposition 2, the **beginof** function is not definable in $\mathcal{T}$, and hence $\mathcal{M}$ is not definable in $\mathcal{T}$. □

Theorem 1 *There does not exist an interpretation of $T_{pslcore}$ into $T_{linear-time}$.*

Proof: This follows from Proposition 1 and Lemma 1. □

By Theorem 1, we cannot use a time ontology alone to reason about activities and their occurrences.

4 Definability and Situation Calculus

In this section, we characterize the relationship between Reiter's second-order axiomatization of the situation calculus and three core theories within the first-order PSL Ontology.

4.1 Axiomatization of Situation Calculus

Consider the theory $T_{sitcalc}$ which is Reiter's second-order axiomatization of the situation calculus ((Reiter 2001), (Levesque *et al.* 1997)). Let $T_{sittime}$ be Pinto's axiomatization of time for situation trees ((Pinto & Reiter 1995)) and let $T_{sitfluent}$ be Pinto's axiomatization of the *holds* relation[3].

[2]The axiomatization of PSL-Core (also referred to as $T_{pslcore}$) in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/psl_core.html`

[3]The axioms of $T_{sitcalc}$ in CLIF can be found at `http://stl.mie.utoronto.ca/colore/sitcalc.clif`.

4.2 Relationship to PSL-Core

Theorem 2 *There exists an interpretation of* $T_{pslcore}$ *into* $T_{sitcalc} \cup T_{sittime}$.

Proof: Suppose

$$\pi_{\mathbf{occurrence\_of}}(s, a) = ((\exists s_1)\ s = do(a, s_1))$$

$$\pi_{\mathbf{activity}}(a) = ((\exists s_1, s_2)\ s = do(a, s_1))$$

$$\pi_{\mathbf{activity\_occurrence}}(s) = ((\exists a, s_1)\ s = do(a, s_1))$$

$$\pi_{\mathbf{timepoint}}(t) = ((\exists s)\ (start(s) = t))$$

$$\pi_{\mathbf{beginof}}(s, t) = ((start(s) = t))$$

$$\pi_{\mathbf{endof}}(s, t) = ((\exists a)\ (end(s, a) = t))$$

It is straightforward to verify that these mappings and the axioms of $T_{sitcalc} \cup T_{sittime}$ entail the axioms of $T_{pslcore}$. □

Of course, it is not surprising to see that there exists an interpretation of $T_{pslcore}$ into $T_{sitcalc} \cup T_{sittime}$, since the theory $T_{pslcore}$ was designed to be the weakest process ontology that is shared by other process ontologies.

4.3 Relationship to Occurrence Trees

Within the PSL Ontology, the theory $T_{occtree}$ extends the theory of $T_{pslcore}$[4]. An occurrence tree is a partially ordered set of activity occurrences, such that for a given set of activities, all discrete sequences of their occurrences are branches of the tree.

An occurrence tree contains all occurrences of *all* activities; it is not simply the set of occurrences of a particular (possibly complex) activity. Because the tree is discrete, each activity occurrence in the tree has a unique successor occurrence of each activity. Every sequence of activity occurrences has an initial occurrence (which is the root of an occurrence tree).

Although occurrence trees characterize all sequences of activity occurrences, not all of these sequences will intuitively be physically possible within the domain. We therefore consider the subtree of the occurrence tree that consists only of *possible* sequences of activity occurrences; this subtree is referred to as the legal occurrence tree.

Occurrence trees are closely related to situation trees, which are the models of Reiter's axiomatization of situation calculus; the following theorems make this intuition more precise.

Theorem 3 *There exists an interpretation of* $T_{occtree} \cup T_{pslcore}$ *into* $T_{sitcalc} \cup T_{sittime}$.

The axioms of $T_{sittime}$ can be found at
`http://stl.mie.utoronto.ca/colore/sittime.clif`.
The axioms of $T_{sitfluent}$ can be found at
`http://stl.mie.utoronto.ca/colore/sitfluent.clif`.

[4] The axioms of $T_{occtree}$ in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/occtree.th.html`

Proof: Suppose

$$\pi_{\mathbf{earlier}}(s_1, s_2) = s_1 < s_2$$

$$\pi_{\mathbf{generator}}(a) = (\exists s_1, s_2)\ s = do(a, s_1)$$

$$\pi_{\mathbf{arboreal}}(s) = (\exists a, s_1)\ s = do(a, s_1)$$

$$\pi_{\mathbf{successor}}(a, s) = do(a, s)$$

$$\pi_{\mathbf{initial}}(s) = (s = do(a, S_0))$$

$$\pi_{\mathbf{legal}}(s) = (executable(s))$$

It is straightforward to verify that these mappings and the axioms of $T_{sitcalc} \cup T_{sittime}$ entail the axioms of $T_{occtree} \cup T_{pslcore}$. □

What of the converse direction – does there exist an interpretation of $T_{sitcalc} \cup T_{sittime}$ into $T_{occtree} \cup T_{pslcore}$. The primary difference between $T_{occtree}$ and $T_{sitcalc}$ is the existence of models of $T_{occtree}$ that are occurrence trees with branches that are not isomorphic to the standard models of the theory $Th(\mathbb{N}, 0, S, <)$; such trees cannot be isomorphic to situation trees.

Definition 4 *WFAS is the first-order axiom schema*

$$(\forall s)\ (\phi(s) \wedge arboreal(s))$$

$$\supset ((\exists x)\phi(x) \wedge earlier(x, s) \wedge ((\forall y) earlier(y, x) \supset \neg\phi(y)))$$

for any first-order formula $\phi(x)$.

This axiom schema is equivalent to saying that all first-order definable sets of elements in an occurrence tree are well-founded.

Theorem 4 *Let* ACA *be a sentence of the form*

$$(\forall a, s_1, s_2)\ (s_2 = do(a, s_1)) \supset (a = A_1) \vee ... \vee (a = A_n)$$

There exists an interpretation of $T_{sitcalc} \cup ACA$ *into* $T_{occtree} \cup T_{pslcore} \cup WFAS$.

Proof: (Sketch) Suppose

$$\pi_{<}(S_0, s_2) = (\exists s)\, initial(s) \wedge (earlier(s, s_2) \vee (s = s_2)$$

$$(s_1 \neq S_0) \Rightarrow \pi_{<}(s_1, s_2) = earlier(s_1, s_2)$$

$$\pi_{\mathbf{do}}(a, S_0) = (\exists s)\ initial(s) \wedge occurrence\_of(s, a)$$

$$(s_1 \neq S_0) \Rightarrow \pi_{\mathbf{do}}(a, s) = successor(a, s)$$

$$\pi_{\mathbf{executable}}(s) = (legal(s))$$

Since the interpretation of theories is specified with respect to first-order entailment, we only need to show that the first-order consequences are preserved by the interpretation.

The techniques introduced in (Doets 1989) and (Backofen, Rogers, & Vijay-Shanker 1995) can be used to show that the models of $T_{occtree} \cup T_{pslcore} \cup WFAS$ are elementarily equivalent to models of $T_{sitcalc} \cup ACA$. □

4.4 Relationship to Discrete States

Most applications of process ontologies are used to represent dynamic behaviour in the world so that intelligent agents may make predictions about the future and explanations about the past. In particular, these predictions and explanations are often concerned with the state of the world and how that state changes. The PSL core theory T_{disc_state} is intended to capture the basic intuitions about states and their relationship to activities[5].

Within the PSL Ontology, state is changed by the occurrence of activities. Intuitively, a change in state is captured by a state that is either achieved or falsified by an activity occurrence. Furthermore, state can only be changed by the occurrence of activities. Thus, if some state holds after an activity occurrence, but after an activity occurrence later along the branch it is false, then an activity must occur at some point between that changes the state. This also leads to the requirement that the state holding after an activity occurrence will be the same state holding prior to any immediately succeeding occurrence, since there cannot be an activity occurring between the two by definition.

Theorem 5 *There exists an interpretation of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$ into $T_{sitcalc} \cup T_{sittime} \cup T_{sitfluent}$.*

Proof: Suppose

$$(s \neq S_0) \Rightarrow \pi_{\mathbf{holds}}(f, s) = holds(f, s)$$

$$\pi_{\mathbf{prior}}(f, s) = (((\forall s, s', a)\ s = do(a, s') \supset holds(f, s)) \\ \wedge(((\exists s, s', a)\ s = do(a, s')) \vee holds(f, S_0)))$$

It is straightforward to verify that these mappings and the axioms of $T_{sitcalc} \cup T_{sittime} \cup T_{sitfluent}$ entail the axioms of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$. □

The interpretation of situation calculus into the PSL Ontology requires an additional assumption that the set of fluents in any model be finite and bounded.

Theorem 6 *Let FCA be a sentence of the form*

$$(\forall f, s)\ holds(f, s) \supset (f = F_1) \vee ... \vee (f = F_m)$$

There exists an interpretation of $T_{sitcalc} \cup T_{sittime} \cup T_{sitfluent} \cup ACA \cup FCA$ into $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore} \cup WFAS$.

Proof: (Sketch) Suppose

$$\pi_{\mathbf{holds}}(f, s) = holds(f, s)$$

$$\pi_{\mathbf{holds}}(f, S_0) = (\exists s)\ initial(s) \wedge prior(f, s)$$

As with Theorem 4, the techniques introduced in (Doets 1989) and (Backofen, Rogers, & Vijay-Shanker 1995) can be used to show that the models of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore} \cup WFAS$ are elementarily equivalent to models of $T_{sitcalc} \cup T_{sittime} \cup T_{sitflent} \cup ACA \cup FCA$. □

[5]The axioms of T_{disc_state} in CLIF can be found at http://www.mel.nist.gov/psl/psl-ontology/part12/disc_state.th.html

Although $T_{sitcalc} \cup T_{sittime} \cup T_{sitfluent}$ cannot be interpreted into $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$ without the axiom schema, we can show that the two theories are equivalent with respect to a restricted class of first-order sentences.

Theorem 7 *Let $Q(s)$ be a simple state formula in the language of $T_{sitcalc}$ and let $Q'(s)$ be the the image of the formula under the interpretation into $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$.*

For any model $\mathcal{M}$ of $T_{sitcalc} \cup T_{sittime} \cup T_{sitfluent}$ there exists a model $\mathcal{N}$ of $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$ such that

$$Th(\mathcal{M}) \models (\forall s)\ Q(s) \Leftrightarrow Th(\mathcal{N}) \models (\forall s)\ Q'(s)$$

and

$$Th(\mathcal{M}) \models (\exists s)\ Q(s) \Leftrightarrow Th(\mathcal{N}) \models (\exists s)\ Q'(s)$$

Proof: (Sketch) Axioms 6 and 7 of T_{disc_state} are logically equivalent to the instantiation of the axiom schema $WFAS$ for positive and negative *holds* literals, respectively. Since simple state formulae are finite boolean combinations of positive and negative *holds* literals with the same activity occurrence variable, the instantiation of $WFAS$ for a simple state formula is logically equivalent to a finite boolean combination of sentences that are entailed by $T_{disc\_state} \cup T_{occtree} \cup T_{pslcore}$. □

The first sentence in Theorem 7 corresponds to the classical planning problem, while the second sentence corresponds to the entailment of state constraints. By this theorem, the PSL Ontology entails the same set of plans and state constraints as $T_{sitcalc}$.

5 Nondefinability Theorems

In this section, we show that the remaining core theories in the PSL Ontology cannot be interpreted in $T_{sitcalc} \cup T_{sittime}$.

5.1 Automorphisms of Situation Trees

All of the nondefinability theorems rest on the characterization of the automorphisms of situation trees and the failure of these automorphisms to preserve the sets that correspond to the extensions of the functions and relations in models of the PSL Ontology. We introduce three lemmas that characterize properties of the automorphisms of situation trees which will be used in later proofs.

Lemma 3 *Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$ and let $Aut(\mathcal{R})$ be the set of automorphisms of $\mathcal{R}$.*

For any $\varphi \in Aut(\mathcal{R})$ and any element $\mathbf{o}$ of the situation tree, $\mathbf{o}$ and $\varphi(\mathbf{o})$ must be on different branches of the situation tree.

Lemma 4 *Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.*

The set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of situations that are the successors of a situation in the tree.

Lemma 5 *Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.*

The set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of actions in $\mathcal{R}$.

5.2 Relationship to Subactivities

The theory $T_{subactivity}$ in PSL Ontology uses the *subactivity* relation to capture the basic intuitions for the composition of activities[6]. This relation is a discrete partial ordering, in which primitive activities are the minimal elements.

Lemma 6 *A model $\mathcal{M}$ of $T_{subactivity} \cup T_{pslcore}$ with nonprimitive activities is not definable in any model of $T_{sitcalc} \cup T_{sittime}$.*

Proof: We will show that the **subactivity** relation in $\mathcal{M}$ is not definable in any model of $T_{sitcalc} \cup T_{sittime}$.

Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.

By Lemma 5, the set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of actions in $\mathcal{R}$; thus, there exists $\varphi \in Aut(\mathcal{R})$ and distinct actions $\mathbf{a_1}, \mathbf{a_2}$ such that $\varphi(\mathbf{a_1}) = \mathbf{a_2}$. By the following axiom of $T_{subactivity}$

$$(\forall a_1, a_2) subactivity(a_1, a_2) \wedge subactivity(a_2, a_1) \supset (a_1 = a_2)$$

we have

$$\langle \mathbf{a_1}, \mathbf{a_2} \rangle \in \mathbf{subactivity} \Rightarrow \langle \varphi(\mathbf{a_1}), \varphi(\mathbf{a_2}) \rangle \notin \mathbf{subactivity}$$

By Proposition 2, the **subactivity** relation is not definable in $\mathcal{R}$, and hence $\mathcal{M}$ is not definable in $\mathcal{R}$. □

Theorem 8 *There does not exist an interpretation of $T_{subactivity} \cup T_{pslcore}$ into $T_{sitcalc} \cup T_{sittime}$.*

Proof: This follows from Lemma 6 and Lemma 1. □

5.3 Relationship to Atomic Activities

The primary motivation behind the core theory T_{atomic} in the PSL Ontology is to capture intuitions about the occurrence of concurrent activities[7]. The core theory T_{atomic} introduces the function *conc* that maps any two atomic activities to the activity that is their concurrent composition. Essentially, an atomic activity corresponds to some set of primitive activities, so that every concurrent activity is equivalent to the composition of a set of primitive activities.

Lemma 7 *A model $\mathcal{M}$ of $T_{atomic} \cup T_{subactivity} \cup T_{pslcore}$ with nonatomic activities is not definable in any model of $T_{sitcalc} \cup T_{sittime}$.*

Proof: We will show that the **conc** function and **atomic** relation in $\mathcal{M}$ are not definable in any model of $T_{sitcalc} \cup T_{sittime}$.

Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.

By Lemma 5, the set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of actions in $\mathcal{R}$; thus there exists $\varphi \in Aut(\mathcal{R})$ and actions $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}$ such that $\mathbf{a_3} = \mathbf{conc}(\mathbf{a_1}, \mathbf{a_2})$ and

$$\varphi(\mathbf{a_1}) = \mathbf{a_1},\ \varphi(\mathbf{a_2}) = \mathbf{a_2},\ \varphi(\mathbf{a_3}) = \mathbf{a_3}$$

It is easy to see that

$$\varphi(\mathbf{conc}(\mathbf{a_1}, \mathbf{a_2})) \neq \mathbf{conc}(\varphi(\mathbf{a_1}), \varphi(\mathbf{a_2}))$$

There also exists $\varphi \in Aut(\mathcal{R})$ and distinct actions $\mathbf{a_1}, \mathbf{a_2}$ such that $\varphi(\mathbf{a_1}) = \mathbf{a_2}$ and

$$\langle \mathbf{a_1}, \mathbf{a_2} \rangle \in \mathbf{subactivity}$$

$$\langle \mathbf{a_1} \rangle \in \mathbf{atomic}, \langle \mathbf{a_2} \rangle \notin \mathbf{atomic}$$

By the following axiom of $T_{subactivity}$

$$(\forall a_1, a_2) subactivity(a_1, a_2) \wedge subactivity(a_2, a_1) \supset (a_1 = a_2)$$

we have

$$\langle \mathbf{a} \rangle \in \mathbf{atomic} \Rightarrow \langle \varphi(\mathbf{a}) \rangle \notin \mathbf{atomic}$$

By Proposition 2, the **conc** function and **atomic** relation are not definable in $\mathcal{R}$, and hence $\mathcal{M}$ is not definable in $\mathcal{R}$. □

Theorem 9 *There does not exist an interpretation of $T_{atomic} \cup T_{subactivity} \cup T_{occtree} \cup T_{pslcore}$ into $T_{sitcalc} \cup T_{sittime}$.*

Proof: This follows from Lemma 7 and Lemma 1. □

5.4 Relationship to Complex Activities

The core theory $T_{complex}$ characterizes the relationship between the occurrence of a complex activity and occurrences of its subactivities[8]. Occurrences of complex activities correspond to sets of occurrences of subactivities; in particular, these sets are subtrees of the occurrence tree. An activity tree consists of all possible sequences of atomic subactivity occurrences beginning from a root subactivity occurrence. In a sense, activity trees are a microcosm of the occurrence tree, in which we consider all of the ways in which the world unfolds in the context of an occurrence of the complex activity.

Lemma 8 *A model $\mathcal{M}$ of $T_{complex} \cup T_{atomic} \cup T_{subactivity} \cup T_{pslcore}$ with nonatomic activities such that not all activity occurrences are elements of nontrivial activity trees is not definable in any model of $T_{sitcalc} \cup T_{sittime}$.*

Proof: We will show that the **root** and **min_precedes** relations in $\mathcal{M}$ are not definable in any model of $T_{sitcalc} \cup T_{sittime}$.

Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.

By Lemma 4, the set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of situations that are the successors of any situation in the tree.

There exists $\varphi_1 \in Aut(\mathcal{R})$ such that for any $\mathbf{s_1}, \mathbf{s_2}$ that are successors of the same element of the situation tree such that $\varphi_1(\mathbf{s_1}) = \mathbf{s_2}$ and such that $\mathbf{s_1}$ is not

[6]The axioms of $T_{subactivity}$ in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/subactivity.th.html`

[7]The axioms of T_{atomic} in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/atomic.th.html`

[8]The axioms of $T_{complex}$ in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/complex.th.html`

an element of any nontrivial activity tree and $\mathbf{s_2}$ is an element of a nontrivial activity tree.

If $\mathbf{s_2}$ is a root of an activity tree, then there exists $\varphi_1 \in Aut(\mathcal{R})$ such that

$$\langle \mathbf{s}, \mathbf{a} \rangle \in \mathbf{root} \Rightarrow \langle \varphi_1(\mathbf{s}), \mathbf{a} \rangle \notin \mathbf{root}$$

If $\mathbf{s_2}$ is not a root of an activity tree, then there exists $\varphi_2 \in Aut(\mathcal{R})$ such that

$$\langle \mathbf{s_1}, \mathbf{s_2}, \mathbf{a} \rangle \in \mathbf{min\_precedes} \Rightarrow$$

$$\langle \varphi_2(\mathbf{s_1}), \varphi_2(\mathbf{s_2}), \varphi_2(\mathbf{a}) \rangle \notin \mathbf{min\_precedes}$$

By Proposition 2, the **root** and **min_precedes** relations are not definable in $\mathcal{R}$, and hence $\mathcal{M}$ is not definable in $\mathcal{R}$. □

Theorem 10 *There does not exist an interpretation of* $T_{complex} \cup T_{atomic} \cup T_{subactivity} \cup T_{occtree} \cup T_{pslcore}$ *into* $T_{sitcalc} \cup T_{sittime}$.

Proof: This follows from Lemma 8 and Lemma 1. □

5.5 Relationship to Complex Activity Occurrences

Within $T_{complex}$, complex activity occurrences correspond to activity trees, and consequently occurrences of complex activities are not elements of the legal occurrence tree. The axioms of the core theory T_{actocc} ensure complex activity occurrences correspond to branches of activity trees[9]. Each complex activity occurrence has a unique atomic root occurrence and each finite complex activity occurrence has a unique atomic leaf occurrence. A subactivity occurrence corresponds to a sub-branch of the branch corresponding to the complex activity occurrence.

Lemma 9 *A model* $\mathcal{M}$ *of* $T_{actocc} \cup T_{complex} \cup T_{atomic} \cup T_{subactivity} \cup T_{pslcore}$ *with occurrences of nonatomic activities is not definable in any model of* $T_{sitcalc} \cup T_{sittime}$.

Proof: We will show that the **subactivity_occurrence** relation in $\mathcal{M}$ is not definable in any model of $T_{sitcalc} \cup T_{sittime}$.

Let $\mathcal{R}$ be a model of $T_{sitcalc} \cup T_{sittime}$.

By Lemma 4, the set of automorphisms $Aut(\mathcal{R})$ of a situation tree is transitive on the set of situations that are the successors of any situation in the tree. Furthermore, $Aut(\mathcal{R})$ only acts on elements of the situation tree, so that it fixes occurrences of complex activities.

By Lemma 3, any $\varphi \in Aut(\mathcal{R})$ maps elements of a branch of the situation tree to another branch of the situation tree; however, the axioms of T_{actoc} entail that all subactivity occurrences of a complex activity occurrences must be on the same branch of the tree.

Thus, for any activity occurrence $\mathbf{o_1}$ that is an element of the situation tree and any complex activity occurrence $\mathbf{o_2}$, there exists $\varphi \in Aut(\mathcal{R})$ such that

$$\langle \mathbf{o_1}, \mathbf{o_2} \rangle \in \mathbf{subactivity\_occurrence} \Rightarrow$$

$$\langle \varphi(\mathbf{o_1}), \mathbf{o_2} \rangle \notin \mathbf{subactivity\_occurrence}$$

By Proposition 2, the **subactivity_occurrence** relation is not definable in $\mathcal{R}$, and hence $\mathcal{M}$ is not definable in $\mathcal{R}$. □

Theorem 11 *There does not exist an interpretation of* $T_{actocc} \cup T_{complex} \cup T_{atomic} \cup T_{subactivity} \cup T_{occtree} \cup T_{pslcore}$ *into* $T_{sitcalc} \cup T_{sittime}$.

Proof: This follows from Lemma 9 and Lemma 1. □

6 Summary

In this paper we have characterized the relationship between the PSL Ontology and two other ontologies – a time ontology and Reiter's second-order axiomatization of situation calculus. With the addition of a first-order axiom schema and the restriction to finite domains of activities and fluents, the PSL Ontology is elementarily equivalent to the situation calculus axiomatization. Furthermore, the core theories in PSL Ontology that axiomatize subactivities and complex activities are not definable in the situation calculus.

References

Backofen, R.; Rogers, J.; and Vijay-Shanker, K. 1995. A first-order axiomatization of the theory of finite trees. *Journal of Logic, Language, and Information* 4:5–39.

Bock, C., and Gruninger, M. 2005. PSL: A semantic domain for flow models. *Software and Systems Modeling* 4:209–231.

Doets, K. 1989. Monadic π^1_1-theories of π^1_1-properties. *Notre Dame Journal of Formal Logic* 30:224–240.

Enderton, H. 1972. *Mathematical Introduction to Logic*. Academic Press.

Gruninger, M. 2004. Ontology of the Process Specification Language. In Staab, S., and Studer, R., eds., *Handbook of Ontologies in Information Systems*. Springer-Verlag.

Hayes, P. 1996. A catalog of temporal theories. Technical Report UIUC-BI-AI-96-01, University of Illinois.

Levesque; H., Reiter, R.; Lesperance, Y.; Lin, F.; and Scherl, R. 1997. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming* 31:92–128.

Marker, D. 2002. *Model Theory: An Introduction*. Springer-Verlag.

Pinto, J., and Reiter, R. 1995. Reasoning about time in the situation calculus. *Annals of Mathematics and Artificial Intelligence* 14:2510–268.

Reiter, R. 2001. *Knowlege in action: logical foundations for specifying and implementing dynamical systems*. Cambridge, MA, USA: MIT Press.

[9]The axioms of T_{actocc} in CLIF can be found at `http://www.mel.nist.gov/psl/psl-ontology/part12/actocc.th.html`