

Representing a Reference Foundational Ontology of Events in $SR\mathcal{O}IQ$

Alessander Botti Benevides^{a,*}, Jean-Rémi Bourguet^a, Giancarlo Guizzardi^{a,b},
Rafael Peñaloza^b and João Paulo A. Almeida^a

^a *NEMO, Computer Science Department, Federal University of Espírito Santo, Av. Fernando Ferrari, 514, Goiabeiras/Vitória, Brazil*

E-mails: abbenevides@inf.ufes.br, jean-remi.bourguet@ufes.br, gguizzardi@inf.ufes.br, jpalmeida@ieee.org

^b *Faculty of Computer Science, Free University of Bozen-Bolzano, Italy*

E-mails: giancarlo.guizzardi@unibz.it, rafael.penaloza@unibz.it

Abstract. In recent years, there has been a growing interest in the application of foundational ontologies, i.e., formal ontological theories in the philosophical sense, to provide a theoretically sound foundation for improving the theory and practice of conceptual modeling and knowledge representation. This paper addresses one particular foundational theory of events termed UFO-B, which has been successfully employed as a reference model for addressing problems from complex media management, enterprise architecture, software engineering, and modeling of events in petroleum exploration. Despite its success, there is still no formalization of UFO-B in a decidable knowledge representation language that could support reasoning about complex events and event relations. We address this gap by proposing a number of alternative translations from UFO-B’s original axiomatization (in first-order logic and in the Alloy formal language) to the description logic $SR\mathcal{O}IQ$, which is the formal underpinning of OWL 2 DL. Additionally, to support practical applications, we translated these $SR\mathcal{O}IQ$ theories to OWL 2 DL TBoxes, which were validated by showing that all the intended models of UFO-B (the logical models of the UFO-B specification in Alloy) that we generated are consistent with these UFO-B TBoxes. In a sense, the specification in Alloy *implements* the specification in first-order logic, while the OWL 2 TBoxes implement the $SR\mathcal{O}IQ$ specifications. Incidentally, the methodology that we designed for the translation from UFO-B’s original axiomatization in FOL and Alloy to $SR\mathcal{O}IQ$ came to be a key contribution of this work by providing us evidence of the inadequacy of DLs for the specification of comprehensive foundational ontologies.

Keywords: Ontology of Events, Description Logics, $SR\mathcal{O}IQ$, OWL, SAT

1. Introduction

The theory and practice of conceptual modeling and knowledge representation can be improved by means of specially designed *foundational ontologies* (i.e., formal ontological theories in the philosophical sense), which can provide a theoretically sound foundation for the modeling constructs and methodologies of the languages developed in these fields. Foundational ontologies used for these purposes address a number of pervasive domain-independent notions including objects, properties, relations, qualities, and events. The latter ontological category enlarges the variety of entities in the world to include not only its “ordinary citizens” (such as animals, physical objects) but also the “things that happen to or are performed by them” (Casati and Varzi, 2015).

*Corresponding author. Tel. number: +5527996561982. E-mail: abbenevides@inf.ufes.br.

Dealing with the representation of events is key to conceptual modeling and knowledge representation, given the importance of events in cognition, language and, in fact, a large part of human endeavor, which is focused on planning and executing actions, effectively bringing about changes in the world. This has motivated the development of a particular philosophically well-founded ontology of events termed Unified Foundational Ontology – Part B (UFO-B) (Guizzardi et al., 2013). It accounts for events as manifestations of object dispositions, and explicates how objects, situations and events relate to each other.

The latest version of UFO-B (Guizzardi et al., 2013) has been extensively tested in practice and successfully employed as a reference model for addressing problems from enterprise architecture (Jr. et al., 2010; Nardi et al., 2014; Azevedo et al., 2015) and software engineering (Guizzardi et al., 2014; Duarte et al., 2016; de Souza et al., 2017). Earlier versions of UFO-B (Guizzardi et al., 2008; Guizzardi and Wagner, 2004) have also been employed in areas such as complex media management (Pena, 2012; Carolo and Burlamaqui, 2011) and the modeling of events in petroleum exploration (Werlang, 2015). Despite that, there is still no formalization of this ontology in terms of a decidable knowledge representation language that could support reasoning about complex events and event relations. In this paper, we address this gap by proposing a number of translations from UFO-B’s original axiomatization (in first-order logic (FOL) and in the Alloy formal language (Jackson, 2012)) to the Description Logic (DL) *SR_QIQ* (Horrocks et al., 2006), which is the formal underpinning of the OWL 2 Web Ontology Language (OWL 2 DL) (W3C, 2012). We employ the specification of UFO-B in Alloy as an implementation of the UFO-B FOL theory, and the specifications of UFO-B in OWL 2 DL as implementations of the specifications of UFO-B in *SR_QIQ*. By using these technologies, we aim to support the practical application of the foundational ontology UFO-B for knowledge representation in the Semantic Web and to enable decidable reasoning about events based on the well-founded ontology UFO-B.

Incidentally, the methodology that we designed for the translation from UFO-B’s original axiomatization in FOL and Alloy to *SR_QIQ* came to be a key contribution of this work by providing us evidence of the inadequacy of DLs for the specification of comprehensive foundational ontologies. Therefore, while the final artifact—a collection of UFO-B formalizations in *SR_QIQ*—is a core contribution of this paper, the methodology that we used to build this artifact is also worth presenting and occupies a large part of this article.

The remainder of this paper is structured as follows. Section 2 briefly introduces the challenges of dealing with *SR_QIQ*, and presents an overview of our methodology for translating a FOL theory to *SR_QIQ*. In Section 3, we present the formalization of UFO-B in FOL. Section 4 shows a translation of the UFO-B FOL theory to a set of *SR_QIQ* axioms, discussing the challenges in mapping these two formalisms. In Section 5, we explain the *structural restrictions* of *SR_QIQ* (two syntactic constraints that guarantee decidability) and show that the full set of *SR_QIQ* axioms (of Section 4) violates these restrictions. We then discuss the incompatibilities between the various UFO-B *SR_QIQ* axioms, showing an encoding of these incompatibilities in propositional logic and the solutions of the related propositional satisfiability problem (SAT), which define 12 288 UFO-B *SR_QIQ* theories that do not violate the *SR_QIQ* restrictions. In Section 6, these 12 288 theories are evaluated by means of codifications in OWL 2 DL, on which consistency tests are performed regarding a set of automatically generated intended models of UFO-B (logical models of the UFO-B specification in Alloy). This shows that the codifications do not rule out these intended models. Section 7 discusses some related work, while Section 8 presents some final considerations.

This paper follows the same methodology applied by Benevides et al. (2017), but extends their results by (1) performing an ontological analysis of the UFO-B *SR_QIQ* theories (see Section 5.5), and (2)

complementing UFO-B with a more comprehensive presentation and philosophical justification of sub-theories dealing with: (i) the differentiation of roles played by objects inside an event (the so-called processual roles (Loebe, 2007, 2003), see Section 3.7); (ii) qualities and quality structures used to predicate qualitative aspects to events (see Section 3.8); (iii) particular aspects of events of the creation, destruction and modification of objects (see Section 3.9).

2. Challenges and methodology

When creating ontologies, there is a trade-off between being faithful to the domain being captured and guaranteeing desirable computational properties, e.g., decidability and tractability of reasoning. A common approach in ontology engineering is to first employ a highly-expressive formal language, like FOL, to create a theory that captures the domain of interest as well as possible. Such a theory is called a *reference ontology*. The focus of a reference ontology is on representation adequacy: the resulting specifications are intended to be used by humans in tasks such as communication, domain analysis and problem-solving. In this case, an important quality of a reference ontology is its precision, i.e., its ability to rule out unintended models, requiring thus expressiveness of the formalism employed. Then, one can employ a less expressive formalism, like *SR_QIQ*, to (often partially) capture a reference ontology as artifacts with desirable computational properties. These artifacts are called *lightweight ontologies* (Guizzardi, 2006, pp. 36–37).

The goal here is to have *SR_QIQ* formalizations of UFO-B. For building these theories, we have to deal with the difference in expressivity between FOL and *SR_QIQ*. Moreover, while the scope of syntactic constraints in FOL concerns individual formulae, *SR_QIQ* also has syntactic constraints encompassing the whole theory/ontology, the so-called *structural restrictions* (Krötzsch et al., 2012, p. 9). The two structural restrictions—*regularity* and *simplicity*—do not concern each single axiom *per se*, but the structure of the ontology as a whole. These restrictions guarantee that the reasoning algorithms are correct and terminating (Krötzsch et al., 2012, p. 9). In particular, they guarantee that the satisfiability problem of *SR_QIQ*-concepts is decidable (Horrocks et al., 2006, pp. 58, 59).

Concerning the methodology we designed for the translation from FOL to *SR_QIQ*, we first perform an unrestricted mapping of the reference ontology. More specifically, we build a set \mathcal{T} (read “calligraphic-T”) of *SR_QIQ* formulae disconsidering the structural restrictions of *SR_QIQ*. We call \mathcal{T} the *unstructured set* of UFO-B *SR_QIQ* axioms. In later sections, we argue that \mathcal{T} does not fully capture UFO-B because some UFO-B FOL axioms are not expressible in *SR_QIQ* (even by disconsidering the structural restrictions).

Next, we show that \mathcal{T} disobeys both structural restrictions of *SR_QIQ* (viz., regularity and simplicity). The incompatibilities between axioms in \mathcal{T} can be expressed in propositional logic by capturing rules like “the inclusion of axiom (ai) implies the exclusion of axiom (aj).” We call “ \mathcal{I} ” the set of all incompatibility rules expressed in propositional logic.

Then, we use a SAT solver to find all subsets of \mathcal{T} that obey the incompatibility rules and thus ensure both regularity and simplicity. We call them *structured subsets* of \mathcal{T} . Solving the SAT problem, we find 6 757 902 900 structured subsets of \mathcal{T} , many of which are *not* maximal w.r.t. being structured, i.e., to obeying regularity and simplicity.¹ Such structural non-maximality means that the subset of \mathcal{T} arbitrarily excludes axioms from UFO-B, losing precision without a reasonable computational justification. Because

¹A subset of \mathcal{T} is maximal w.r.t. obeying regularity and simplicity if and only if (iff) the inclusion of any other axiom generates a subset that is not a *SR_QIQ* ontology due to disobeying the regularity or simplicity restrictions.

of this, we are interested in finding the maximally structured subsets, each of which we consider as a possible representation of UFO-B in *SROLQ*. We show that there are 12 288 of such maximally structured subsets. As we discuss later (Section 5.5), selecting one particular subset involves sacrificing one or more axioms of UFO-B.

In order to automate the previous steps, we built a tool that, given a textual representation of \mathcal{T} , derives the incompatibility rules and returns all the maximally structured subsets of \mathcal{T} .

We check these maximally structured subsets of \mathcal{T} by (i) codifying the full UFO-B in Alloy; and (ii) codifying each of the 12 288 theories as a terminology box (TBox) in OWL 2 DL. Then, (iii) we use the Alloy Analyzer² tool to generate a number of models of the full UFO-B Alloy codification; these models can be seen as *intended models* of UFO-B. (iv) each generated model is then transformed to an OWL 2 DL individual assertion box (ABox). (v) by joining each one of the 12 288 TBoxes with each generated ABox, we build a number of OWL 2 DL knowledge bases (KBs), on which consistency tests are performed and yield positive results without any exceptions.

3. The formalization of UFO-B in FOL

First, we present some notational conventions: (i) FOL predicates and functions, and DL concepts and roles are denoted in Uppercase typewriter style, e.g., “Person” and “motherOf;” (ii) we employ restricted quantification *à la Frege*, i.e., “ $\forall x:T(\varphi)$ ” is a schema for “ $\forall x(T(x) \rightarrow \varphi)$,” and “ $\exists x:T(\varphi)$ ” is a schema for “ $\exists x(T(x) \wedge \varphi)$,” and (iii) we keep the indexes of the UFO-B formulae originally presented in Guizzardi et al. (2013), appending a prime symbol “’” to the index whenever we revise such a formula.

UFO-B extended the Unified Foundational Ontology (Guizzardi, 2005, UFO-A) to deal with events. Besides the FOL formalization, Guizzardi et al. (2013) also indicated a publicly available Alloy axiomatization that also formalized subjects not contemplated in that paper. In this section, we present an updated version of UFO-B that also formalizes those subjects. Here, we homogenized the names of relations defined in the original presentation of UFO-B (Guizzardi et al., 2013) by expanding abbreviations and replacing the forms “x-y” or “x_y” with “xY”.

UFO-B is composed of the following sub-theories: the instantiation of universals by individuals (Section 3.1); a definition of a parthood relation for events (Section 3.2); an explanation of the nature of events as manifestations of objects’ dispositions (Section 3.3); constraints on how objects participate in events, and how events depend on objects (Section 3.4); a suggestion of a linear time structure in order to discuss temporal constraints related to event parthood (Section 3.5); the view of events as the entities responsible for *world changes*, a sort of transition between *situations* (Section 3.6); the differentiation of roles played by objects inside an event (the so-called processual roles, Section 3.7); qualities and quality structures used to predicate qualitative aspects to events (Section 3.8); and particular aspects of events of the creation, destruction and modification of objects (Section 3.9).

3.1. Universals and individuals: (U1)–(U5)

First of all, according to UFO-A, a Thing can be either a Set or an Urelement.³ An Urelement can be either an Individual or a Universal (Guizzardi, 2005, Figure 6–12), meaning that Universals are reified in the domain of quantification of our first-order theory (see Figure 1). Usually, the *instantiation*

²<https://github.com/AlloyTools/org.alloytools.alloy/releases>.

³In simple words, an urelement is something that can be an element of a set, but is not a set itself.

- U3** $\forall x:\text{Thing}, y:\text{Event} \text{Universal}(\text{instantiates}(x, y) \rightarrow \text{Event}(x))$
U4 $\forall x:\text{Thing}, y:\text{Participation} \text{Universal}(\text{instantiates}(x, y) \rightarrow \text{Participation}(x))$
U5 $\forall x:\text{Object}, y:\text{ProcessualRole}(\text{instantiates}(x, y) \rightarrow \text{plays}(x, y))$
t1 $\forall x:\text{ProcessualRole}(\exists y:\text{Object}(\text{plays}(y, x)))$

3.2. Event mereology: (M1)–(M9)

Events may be composed of other Events, e.g., “the murder of Caesar” has as parts “the stabbing of Caesar by Brutus” and “Caesar’s death.” Consider a strict partial order (that is, irreflexive (M3), asymmetric (M4) and transitive (M5)) relation `hasPart` between events.⁴ An Event is atomic iff it has no parts (M1), and complex otherwise (M2).

- M1** $\forall e:\text{Event}(\text{AtomicEvent}(e) \leftrightarrow \neg \exists e':\text{Event}(\text{hasPart}(e, e')))$
M2 $\forall e:\text{Event}(\text{ComplexEvent}(e) \leftrightarrow \neg \text{AtomicEvent}(e))$
M3 $\forall e:\text{ComplexEvent}(\neg \text{hasPart}(e, e))$
M4 $\forall e, e':\text{ComplexEvent}(\text{hasPart}(e, e') \rightarrow \neg \text{hasPart}(e', e))$
M5 $\forall e, e':\text{ComplexEvent}, e'':\text{Event}((\text{hasPart}(e, e') \wedge \text{hasPart}(e', e'')) \rightarrow \text{hasPart}(e, e''))$

UFO-B has two notions of overlapping: mereological overlapping and temporal overlapping. Here, we call them `mereologicallyOverlaps` and `temporallyOverlaps`, respectively. The relation of mereological overlapping is usually reflexive and symmetric, holding also for atoms (Casati and Varzi, 1999, p. 37, (3.1),(3.8),(3.9)). `mereologicallyOverlaps` is implicitly defined by (M7'), in terms of `hasPart`, being reflexive and symmetric.

- M7'** $\forall e, e':\text{Event}(\text{mereologicallyOverlaps}(e, e') \leftrightarrow (e = e' \vee \text{hasPart}(e, e') \vee \text{hasPart}(e', e) \vee \exists e'':\text{Event}(\text{hasPart}(e, e'') \wedge \text{hasPart}(e', e''))))$

UFO-B also commits to *weak supplementation* (Casati and Varzi, 1999, p. 39, (P.4)): a whole mereologically differs from its proper parts, i.e., a whole must have at least two non-overlapping parts (M6); and *strong supplementation* (Casati and Varzi, 1999, p. 39, (P.5)) (M8), which yields a theory in which any *two* different entities cannot have all proper parts in common, as represented by the theorem of extensionality (Casati and Varzi, 1999, p. 40, (3.15)) (our (M9)).

- M6** $\forall e:\text{ComplexEvent}, e':\text{Event}(\text{hasPart}(e, e') \rightarrow \exists e'':\text{Event}(\text{hasPart}(e, e'') \wedge \neg \text{mereologicallyOverlaps}(e', e'')))$
M8 $\forall e, e':\text{ComplexEvent}(\forall e'':\text{Event}(\text{hasPart}(e, e'') \rightarrow \text{hasPart}(e', e'')) \rightarrow (e = e' \vee \text{hasPart}(e', e)))$
M9 $\forall e, e':\text{ComplexEvent}(e = e' \leftrightarrow \forall e'':\text{Event}(\text{hasPart}(e, e'') \leftrightarrow \text{hasPart}(e', e'')))$

3.3. Events as manifestations of object dispositions: (D1')–(D9)

AtomicEvents are *manifestations of* (the inverse of `manifestedBy`) unique Dispositions (D2). Dispositions `inheresIn` a unique ConcreteIndividual (D1'). Whenever a Disposition `inheresIn` an Object and is `manifestedBy` an Event, the Event `dependsOn` the Object (D4). An

⁴As noted by Frank Loebe (reviewer), this `hasPart` relation is neither the parthood relation of UFO-A (which holds between objects) nor a general parthood relation. The (co)domain of `hasPart` is `Event`. When unifying UFO-A and UFO-B, `hasPart` should be renamed to `eventHasPart`.

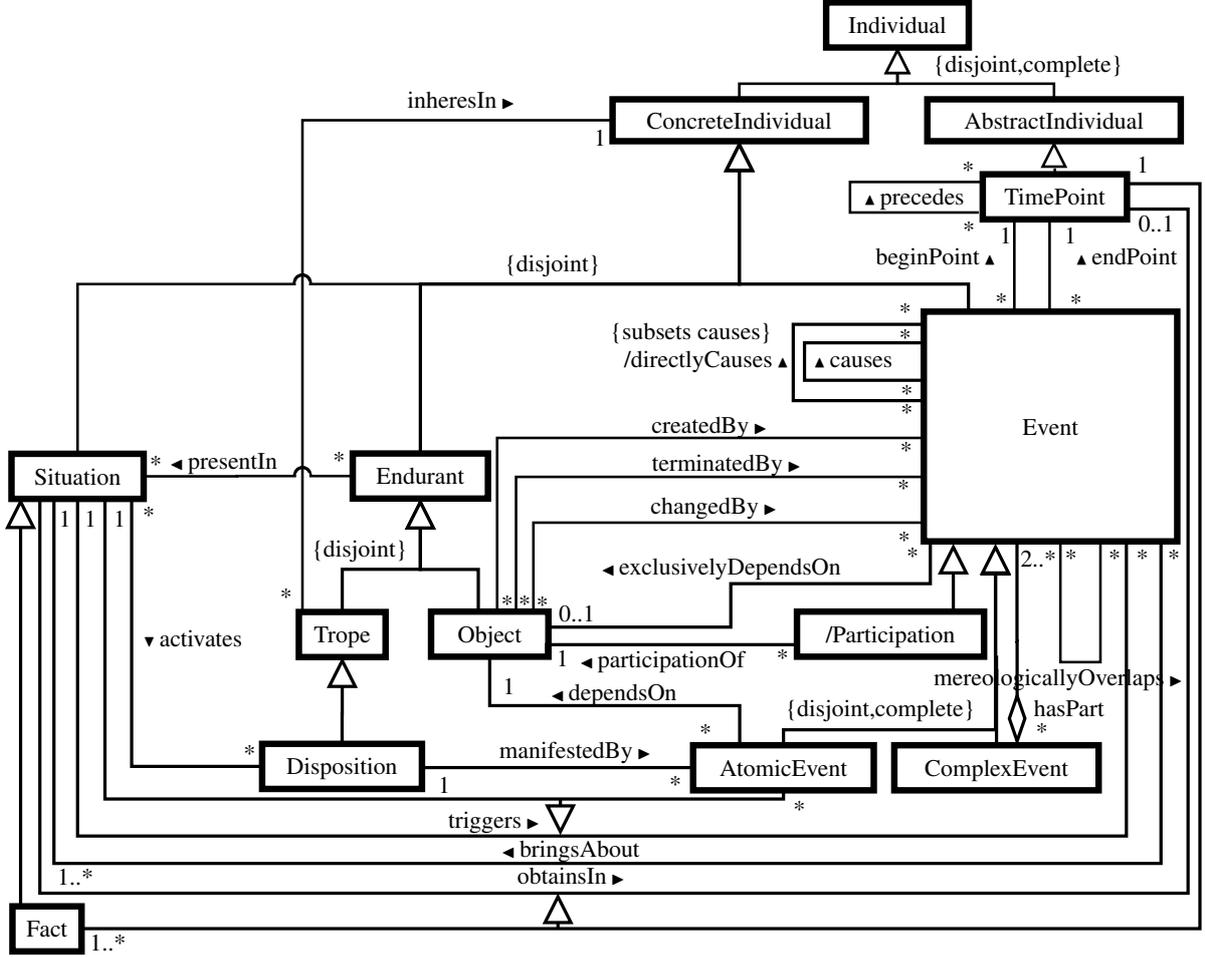


Figure 2. Individuals (based on the figures in Guizzardi et al. (2013)).

Endurant can be presentIn a Situation (see Figure 2). A Situation (Section 3.6) triggers an AtomicEvent iff there is a Disposition that is *activated by* (the inverse of *activates*) the Situation and that is manifestedBy the Event (D3).⁵

D1' $\forall d:\text{Disposition}(\exists!o:\text{ConcreteIndividual}(\text{inheresIn}(d,o)))$

D2 $\forall e:\text{AtomicEvent}(\exists!d:\text{Disposition}(\text{manifestedBy}(d,e)))$

D3 $\forall s:\text{Situation}, e:\text{AtomicEvent}(\text{triggers}(s,e) \leftrightarrow \exists d:\text{Disposition}(\text{activates}(s,d) \wedge \text{manifestedBy}(d,e)))$

D4 $\forall d:\text{Disposition}, e:\text{AtomicEvent}, o:\text{Object}(\text{manifestedBy}(d,e) \wedge \text{inheresIn}(d,o) \rightarrow \text{dependsOn}(e,o))$

⁵Originally, UFO-B assumed that Dispositions inheresIn a unique Object. Here, since we have introduced ConcreteIndividuals and our theory of Situations is quite simple, we are more liberal on the nature of the relation inheresIn. For instance: (i) is the life of an object a manifestation of its disposition to exist in time? (ii) is the holding of a situation from t_0 to t_1 a manifestation of its disposition to hold? (iii) is the happening of an event from t_0 to t_1 a manifestation of its disposition to happen?

Since UFO-A ensured that Objects *bear* (the inverse of *inheresIn*) at least one Trope, here we enforce that both Objects and Events bear at least one Trope (D5). Moreover, since the domain of *inheresIn* (Trope) is not disjoint from its codomain (ConcreteIndividual), we explicitly assert the asymmetry (D6) and intransitivity of *inheresIn* (D7).⁶

- D5** $\forall x:\text{ConcreteIndividual}((\text{Object}(x) \vee \text{Event}(x)) \rightarrow \exists y:\text{Trope}(\text{inheresIn}(y, x)))$
D6 $\forall c, c':\text{ConcreteIndividual}(\text{inheresIn}(c, c') \rightarrow \neg \text{inheresIn}(c', c))$
D7 $\forall c, c', c'':\text{ConcreteIndividual}((\text{inheresIn}(c, c') \wedge \text{inheresIn}(c', c'')) \rightarrow \neg \text{inheresIn}(c, c''))$

If a Disposition is manifestedBy an Event, the Disposition is presentIn the Situation that triggers the Event and also in the Situation that the Event bringsAbout (D8). Our point of view is that a Disposition d must temporally exist at the endPoint of its manifestations, being present in some Situation that obtainsIn this TimePoint. This is because we assume that the Event of the destruction of d cannot happen while d is manifesting itself. The closest the destruction of d can get to a manifestation of d is when the endPoint of the manifestation is the beginPoint of the destruction, when d still temporally exists. On the other hand, one could assume that dispositions can manifest themselves while being destroyed. In this case, “bringsAbout(e, s') \rightarrow presentIn(d, s')” should be removed from (D8). We have no strong objections against this point of view. Finally, the bearer of a Disposition—the entity in which the Disposition *inheresIn*—is presentIn in all Situations which the Disposition is presentIn (D9).

- D8** $\forall e:\text{Event}, d:\text{Disposition}(\text{manifestedBy}(d, e) \rightarrow (\forall s, s':\text{Situation}(\text{triggers}(s, e) \rightarrow \text{presentIn}(d, s)) \wedge (\text{bringsAbout}(e, s') \rightarrow \text{presentIn}(d, s')))))$
D9 $\forall d:\text{Disposition}, o:\text{Object}(\text{inheresIn}(d, o) \rightarrow (\forall s:\text{Situation}(\text{presentIn}(d, s) \rightarrow \text{presentIn}(o, s))))$

3.4. On the participation of objects in events: (P1)–(P5)

Objects participate not only in the AtomicEvents that are manifestations of their Dispositions, but also in the ComplexEvents that have such manifestations as parts. For instance, Caesar participates not only in “Caesar’s death,” but also in “the murder of Caesar.” AtomicEvents dependsOn unique Objects (P1). *exclusivelyDependsOn* generalizes *dependsOn* to ComplexEvents, whereas these relations are equivalent on AtomicEvents (P2). By composing the relations *hasPart* and *exclusivelyDependsOn*, ComplexEvents *exclusivelyDependsOn* the Objects that its parts *exclusivelyDependsOn* (P3). A Participation is an Event that *exclusivelyDependsOn* a unique Object (P4); this yields an orthogonal way to partition Events that collects all and only the ways a specific Object contributes to the manifestation of the atomic (proper and non-proper) parts of an Event. A *participationOf* relation from Participations to Objects can be implicitly defined by means of *exclusivelyDependsOn* (P5).

- P1** $\forall e:\text{AtomicEvent} \exists! o:\text{Object}(\text{dependsOn}(e, o))$
P2 $\forall e:\text{AtomicEvent}, o:\text{Object}(\text{exclusivelyDependsOn}(e, o) \leftrightarrow \text{dependsOn}(e, o))$

⁶*inheresIn* is also acyclic, and finite acyclicity can be represented in FOL by means of a little set theory (7 axioms). However, exposing such axioms would deviate the focus of this paper into more technicalities. Nevertheless, we enforced the finite acyclicity of *inheresIn* in our Alloy codification of UFO-B.

- P3** $\forall e:\text{ComplexEvent}, o:\text{Object}$
 $\text{exclusivelyDependsOn}(e, o) \leftrightarrow$
 $\forall e':\text{Event}(\text{hasPart}(e, e') \rightarrow \text{exclusivelyDependsOn}(e', o))$
- P4** $\forall e:\text{Event}(\text{Participation}(e) \leftrightarrow \exists! o:\text{Object}(\text{exclusivelyDependsOn}(e, o)))$
- P5** $\forall o:\text{Object}, p:\text{Participation}$
 $\text{participationOf}(p, o) \leftrightarrow \text{exclusivelyDependsOn}(p, o)$

3.5. Temporal relations between events: (T1)–(T14')

The set of TimePoints is totally ordered by the relation precedes (T1)–(T4), and the temporal extension of Events and Situations is described by means of the functional relations beginPoint and endPoint (T5'), (T6'). Allen's temporal relations (1983), which are not present in Figure 2, are formalized by means of (T7')–(T13'). The temporal extent of an Event (improperly) includes the temporal extent of all the (proper) parts of the Event (T14').

- T1** $\forall t:\text{TimePoint}(\neg \text{precedes}(t, t))$
- T2** $\forall t, t':\text{TimePoint}(\text{precedes}(t, t') \rightarrow \neg \text{precedes}(t', t))$
- T3** $\forall t, t', t'':\text{TimePoint}((\text{precedes}(t, t') \wedge \text{precedes}(t', t'')) \rightarrow \text{precedes}(t, t''))$
- T4** $\forall t, t':\text{TimePoint}((t \neq t') \rightarrow (\text{precedes}(t, t') \vee \text{precedes}(t', t)))$
- T5'** $\forall e:\text{Event} \exists t, t':\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{endPoint}(e, t'))$
- T6'** $\forall e:\text{Event}, t, t':\text{TimePoint}((\text{beginPoint}(e, t) \wedge \text{endPoint}(e, t')) \rightarrow \text{precedes}(t, t'))$
- T7'** $\forall e, e':\text{Event}(\text{before}(e, e') \leftrightarrow$
 $\exists t, t':\text{TimePoint}(\text{endPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{precedes}(t, t'))$
- T8'** $\forall e, e':\text{Event}(\text{meets}(e, e') \leftrightarrow \exists t:\text{TimePoint}(\text{endPoint}(e, t) \wedge \text{beginPoint}(e', t)))$
- T9'** $\forall e, e':\text{Event}(\text{temporallyOverlaps}(e, e') \leftrightarrow$
 $\exists t_0, \dots, t_3:\text{TimePoint}(\text{beginPoint}(e, t_0) \wedge \text{beginPoint}(e', t_1) \wedge$
 $\text{endPoint}(e, t_2) \wedge \text{endPoint}(e', t_3) \wedge \text{precedes}(t_0, t_1) \wedge \text{precedes}(t_1, t_2) \wedge$
 $\text{precedes}(t_2, t_3)))$
- T10'** $\forall e, e':\text{Event}(\text{starts}(e, e') \leftrightarrow$
 $\exists t, t', t'':\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t) \wedge \text{endPoint}(e, t') \wedge$
 $\text{endPoint}(e', t'') \wedge \text{precedes}(t', t''))$
- T11'** $\forall e, e':\text{Event}(\text{during}(e, e') \leftrightarrow$
 $\exists t_0, \dots, t_3:\text{TimePoint}(\text{beginPoint}(e, t_0) \wedge \text{beginPoint}(e', t_1) \wedge$
 $\text{endPoint}(e, t_2) \wedge \text{endPoint}(e', t_3) \wedge \text{precedes}(t_1, t_0) \wedge \text{precedes}(t_2, t_3)))$
- T12'** $\forall e, e':\text{Event}(\text{finishes}(e, e') \leftrightarrow$
 $\exists t, t', t'':\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{endPoint}(e, t'') \wedge$
 $\text{endPoint}(e', t'') \wedge \text{precedes}(t', t))$
- T13'** $\forall e, e':\text{Event}(\text{equals}(e, e') \leftrightarrow$
 $\exists t, t':\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t) \wedge \text{endPoint}(e, t') \wedge$
 $\text{endPoint}(e', t'))$
- T14'** $\forall e, e':\text{Event}(\text{hasPart}(e, e') \rightarrow$
 $((\exists t:\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t)) \vee$
 $\exists t, t':\text{TimePoint}(\text{beginPoint}(e, t) \wedge \text{beginPoint}(e', t') \wedge \text{precedes}(t, t'))) \wedge$
 $(\exists t:\text{TimePoint}(\text{endPoint}(e, t) \wedge \text{endPoint}(e', t)) \vee$
 $\exists t, t':\text{TimePoint}(\text{endPoint}(e, t) \wedge \text{endPoint}(e', t') \wedge \text{precedes}(t', t))))$

3.6. World changes and situations: (S1')–(S8)

All Events are *triggered by* a unique Situation (S3), and every Event bringsAbout a unique Situation (S4). If a Situation triggers an Event, the Situation obtainsIn the same TimePoint that is the beginPoint of the Event (S1'). Similarly, if an Event bringsAbout a Situation, the Situation obtainsIn the same TimePoint that is the endPoint of the Event (S2'). Facts are Situations that obtain (S5). An Event e directlyCauses an Event e' “by means of” a Situation that was *brought about by* (the inverse of bringsAbout) e and that triggers e' (S6). On (S7), causes should be the transitive closure of directlyCauses, what is inexpressible in FOL. Here, we mitigate this issue by adding a new axiom to the original UFO-B in order to constrain causes to be asymmetric (S8).

- S1'** $\forall s:\text{Situation}, e:\text{Event}(\text{triggers}(s, e) \rightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t) \wedge \text{beginPoint}(e, t)))$
S2' $\forall s:\text{Situation}, e:\text{Event}(\text{bringsAbout}(e, s) \rightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t) \wedge \text{endPoint}(e, t)))$
S3 $\forall e:\text{Event}(\exists!s:\text{Situation}(\text{triggers}(s, e)))$
S4 $\forall e:\text{Event}(\exists!s:\text{Situation}(\text{bringsAbout}(e, s)))$
S5 $\forall s:\text{Situation}(\text{Fact}(s) \leftrightarrow \exists t:\text{TimePoint}(\text{obtainsIn}(s, t)))$
S6 $\forall e, e':\text{Event}(\text{directlyCauses}(e, e') \leftrightarrow \exists s:\text{Situation}(\text{bringsAbout}(e, s) \wedge \text{triggers}(s, e')))$
S7 $\forall e, e':\text{Event}(\text{causes}(e, e') \leftrightarrow (\text{directlyCauses}(e, e') \vee \exists e'':\text{Event}(\text{causes}(e, e'') \wedge \text{causes}(e'', e'))))$
S8 $\forall e, e':\text{Event}(\text{causes}(e, e') \rightarrow \neg \text{causes}(e', e))$

3.7. Processual roles: (R1)–(R2)

This section deals with an interesting aspect of the interplay between Endurants and Events, the so-called ProcessualRoles (Loebe, 2007, 2003). In UFO-B, ProcessualRoles synchronically classify Endurants w.r.t. the specific ways endurants participate in Events. This means that a synchronic classification is inferred from a diachronic (historical) classification. For instance, *mary* is a student now because she is studying now.⁷ Different from ProcessualRoles, *relational roles* (see Guizzardi (2005)) are inferred from the holding of a specific relationship. For instance, *mary* is a student now because there is an enrollment relationship between *mary* and a university that holds now. Notice how different the two meanings of the student roles are in these two examples. It is of fundamental importance to provide an ontological foundation that allows modelers and ontology engineers to disambiguate between ProcessualRoles and relational roles. For instance, while *mary qua* worker in the relational sense has many rights and obligations and is what matters for the human-resources department, it is *mary qua* worker in the processual sense that accomplishes goals that are important for her supervisor. In this scenario, an information system for the human-resources department deals with qua-individuals playing relational roles, while an information system for the production department deals with qua-individuals playing processual roles.

As said before, the relation plays is an instantiation relation from Objects to ProcessualRoles. The key notion here is that an Object plays a ProcessualRole inducedBy a ParticipationUniversal, where ParticipationUniversals are Universals that classify Events. A ProcessualRole

⁷We avoid here the difficult task of providing an epistemological explanation for *ongoing events*.

is inducedBy a unique ParticipationUniversal; and ParticipationUniversals must induce at least one ProcessualRole.

Most importantly, we provide both a necessary and a sufficient condition for an Object to plays a ProcessualRole. On the sufficient condition, if a ProcessualRole r is inducedBy a ParticipationUniversal u and an Object o is the participant of an AtomicEvent p that instantiates u , or p is partOf a Participation that instantiates u , then o plays the ProcessualRole r (R1). Concerning the necessary condition for an Object to plays a ProcessualRole, if an Object o plays a ProcessualRole r , then there is an AtomicEvent p that instantiates a ParticipationUniversal u that induces r , or p is partOf a Participation that instantiates a ParticipationUniversal u that induces r (R2).

R1 $\forall o:\text{Object}, p:\text{AtomicEvent}, u:\text{ParticipationUniversal}, r:\text{ProcessualRole}(\text{dependsOn}(p, o) \wedge \text{inducedBy}(r, u) \wedge (\text{instantiates}(p, u) \vee \exists c:\text{Participation}(\text{hasPart}(c, p) \wedge \text{instantiates}(c, u)))) \rightarrow \text{plays}(o, r))$

R2 $\forall o:\text{Object}, r:\text{ProcessualRole}(\text{plays}(o, r) \rightarrow (\exists p:\text{AtomicEvent}, u:\text{ParticipationUniversal}(\text{dependsOn}(p, o) \wedge \text{inducedBy}(r, u) \wedge (\text{instantiates}(p, u) \vee (\exists c:\text{Participation}(\text{hasPart}(c, p) \wedge \text{instantiates}(c, u))))))))$

3.8. Qualities and quality structures: (Q1)–(Q3)

Since Events are immutable, their Qualities cannot change. Although “static,” such Qualities may *endure* through time, which suggests that Qualities of Events must be Endurants. Examples of Qualities inhering in Events include: location, duration, the path of a movement, and the cost of an activity.

As shown in Figure 1, Qualities are Tropes disjoint with Dispositions. Both Qualities and Dispositions must inhereIn a ConcreteIndividual.

Qualities provide an ontologically parsimonious approach to properties: john and mary have their own immutable and nontransferable weight qualities, which may change by means of *quale substitution*. Weight qualities are *bona fide* entities, e.g., john’s-weight and mary’s-weight, which inheresIn john and mary, respectively, and instantiates the QualityUniversal Weight. QualityUniversal is a Universal and is disjoint with ObjectUniversal, and with EventUniversal. Moreover, given a QualityUniversal, a ConcreteIndividual can bear at most one Quality that instantiates this universal (Q1).

Q1 $\forall x, y:\text{Quality}, z:\text{ConcreteIndividual}, w:\text{QualityUniversal}(\text{inheresIn}(x, z) \wedge \text{inheresIn}(y, z) \wedge \text{instantiates}(x, w) \wedge \text{instantiates}(y, w) \rightarrow x = y)$

Quales are AbstractIndividuals that are memberOf QualityStructures, which are Sets. A Quale is memberOf a unique QualityStructure, while a QualityStructure must have at least one Quale.

QualityStructures are associatedWith exactly one QualityUniversal, while a QualityUniversal is associated with at least one QualityStructure. These associatedWith relations serve

to constrain the kind of values (Quales) that a quality that is instance of a certain QualityUniversal can have. First, given a Quality x that instantiates a QualityUniversal y associated with a QualityStructure z , there is exactly one Quale q that is memberOf z and is qualeOf x (Q2). Second, if a Quale q is a qualeOf a Quality x , then there is a QualityStructure z associatedWith a QualityUniversal y and s.t. q is memberOf z and x instantiates y (Q3).

$$\text{Q2 } \forall x:\text{Quality}, y:\text{QualityUniversal}, z:\text{QualityStructure}(\text{instantiates}(x, y) \wedge \text{associatedWith}(z, y)) \rightarrow \exists! q:\text{Quale}(\text{memberOf}(q, z) \wedge \text{qualeOf}(q, x))$$

$$\text{Q3 } \forall x:\text{Quality}, q:\text{Quale}(\text{qualeOf}(q, x) \rightarrow \exists y:\text{QualityUniversal}, z:\text{QualityStructure}(\text{associatedWith}(z, y) \wedge \text{memberOf}(q, z) \wedge \text{instantiates}(x, y)))$$

Continuing our example, the quale 71kg is a qualeOf of both john's-weight and mary's-weight now. The fact that john and mary have the same weight now means that the qualities john's-weight and mary's-weight point at the same 71kg quale now (by means of qualeOf). john (or mary) changes his weight by means of *quale substitution*, e.g., the qualeOf relationship from john's-weight to 71kg ceases to hold, and a qualeOf relationship from john's-weight to 70kg starts to hold.⁸ Notice, however, that Qualities of Events are like tropes in the classical sense⁹ given that these Qualities can never change because Events are immutable.

Finally, TimePoint is a specialization of AbstractIndividual and may be related to a temporal structure specializing Set, but TimePoint is not a Quale because it is the qualeOf no Quality. Making TimePoint the qualeOf a Quality commits to the idea that *existence at a time* is a property, which is refused by most philosophers. For example, Armstrong (2004) claims that it is a mistake to recognize states of affairs with the form “ a 's existence”, because this will turn existence into a property of a .

3.9. The creation, termination and change of objects: (C1)–(C3)

In this section, we introduce three relations from Objects to Events—viz., createdBy, terminatedBy (meaning “destroyed by”) and changedBy—on which we impose no cardinality constraints to avoid necessitating an infinite regress in the creation of Objects, what would result in all models of UFO-B being infinite, and also to allow for eternal objects in infinite models.

Concerning createdBy, an Object createdBy an Event cannot be presentIn a Situation that triggers the Event, but must be presentIn a Situation brought about by the Event, and the Event or one of its proper parts must dependsOn the Object (C1).

$$\text{C1 } \forall o:\text{Object}, e:\text{Event}(\text{createdBy}(o, e) \rightarrow \forall s, s':\text{Situation}(\text{presentIn}(o, s) \wedge \text{triggers}(e, s) \rightarrow \text{presentIn}(o, s') \wedge \text{dependsOn}(e, s')))$$

⁸Note that by “relationship” we mean a tuple from a relation.

⁹While a Quality can change, in classical trope theory a trope cannot change. For instance, consider that both john and mary have 70Kg now. Classical trope theorists would say that john bears his own immutable and nontransferable 70kg-ness trope, which *exactly resembles* but is numerically different from the 70kg-ness trope of mary. john (or mary) change his weight by means of *trope substitution*, e.g., john's 71kg-ness ceases to exist, and a 70kg-ness trope starts inhering in john. Classical tropes are then similar to reified qualeOf relationships, in the sense that in a classical trope-based approach, each qualeOf relationship would be modeled by means of a unique trope. See Daly (1994) for a short introduction.

$$\begin{aligned}
& (\text{triggers}(s, e) \wedge \text{bringsAbout}(e, s')) \rightarrow \\
& (\neg \text{presentIn}(o, s) \wedge \text{presentIn}(o, s') \wedge (\text{dependsOn}(e, o) \vee \exists e': \text{AtomicEvent}(\text{hasPart}(e, e') \wedge \text{dependsOn}(e', o))))))
\end{aligned}$$

Regarding `terminatedBy`, an Object `terminatedBy` an Event must be `presentIn` a Situation that triggers the Event, but cannot be `presentIn` a Situation brought about by the Event, and the Event or one of its proper parts must `dependsOn` the Object (C2).

$$\begin{aligned}
\mathbf{C2} \quad & \forall o:\text{Object}, e:\text{Event}(\text{terminatedBy}(o, e) \rightarrow \\
& \forall s, s':\text{Situation}(\text{triggers}(s, e) \wedge \text{bringsAbout}(e, s') \rightarrow \\
& (\text{presentIn}(o, s) \wedge \neg \text{presentIn}(o, s') \wedge (\text{dependsOn}(e, o) \vee \exists e': \text{AtomicEvent}(\text{hasPart}(e, e') \wedge \text{dependsOn}(e', o))))))
\end{aligned}$$

Concerning `changedBy`, an Object `changedBy` an Event must be `presentIn` all Situations that triggers the Event or that are brought about by the Event; the Event or one of its proper parts must `dependsOn` the Object; and in order to explain the change, a Disposition must start or cease to be `presentIn` the Situations mentioned before (C3). Another possibility would be to explain the change in the object by means of a change in one of its dispositions, instead of an interchange of dispositions. However, one now faces the problem of explaining changes in dispositions.

$$\begin{aligned}
\mathbf{C3} \quad & \forall o:\text{Object}, e:\text{Event}(\text{changedBy}(o, e) \rightarrow \\
& \forall s, s':\text{Situation}(\text{triggers}(s, e) \wedge \text{bringsAbout}(e, s') \rightarrow \\
& (\text{presentIn}(o, s) \wedge \text{presentIn}(o, s') \wedge (\text{dependsOn}(e, o) \vee \exists e': \text{AtomicEvent}(\text{hasPart}(e, e') \wedge \text{dependsOn}(e', o)))) \wedge \\
& \exists d:\text{Disposition}(\text{inheresIn}(d, o) \wedge ((\neg \text{presentIn}(d, s) \wedge \text{presentIn}(d, s')) \vee \\
& (\text{presentIn}(d, s) \wedge \neg \text{presentIn}(d, s'))))))))
\end{aligned}$$

Finally, note that our UFO-B FOL theory focuses on legibility, and we may have accidentally added (non-trivial) theorems as axioms. This potentially affects the performance of reasoning with our UFO-B *SR_{OLQ}* theories. We leave an optimization of the theories presented here for future work.

4. The unstructured set of UFO-B *SR_{OLQ}* axioms

In this section, we map each UFO-B FOL axiom to *SR_{OLQ}*, disregarding the structural restrictions of *SR_{OLQ}*. The result is \mathcal{T} , the *unstructured set* of UFO-B *SR_{OLQ}* axioms (Section 4.2). Let us introduce, in Section 4.1, a bit of the *SR_{OLQ}* syntax required for \mathcal{T} .

4.1. Concepts and roles in *SR_{OLQ}*

In DL, unary predicates can be represented by means of *concepts*, and binary predicates by *roles*. Let \mathbf{C} be a set of *concept names*¹⁰ and \mathbf{R} be a set of *role names*, where \mathbf{C} and \mathbf{R} are disjoint. Let “ $\overset{\sim}{\leq}$ ” denote \leq , =, or \geq .

Definition 1 (Role). A role is either a role name R from \mathbf{R} or the inverse R^{-1} of a role name R from \mathbf{R} .¹¹ The universal role U is a role name in \mathbf{R} .

Definition 2 (Concept). Concepts can be built by means of the operators of complement “ \neg ,” union “ \sqcup ,” intersection “ \sqcap ,” universal restriction “ \forall ,” existential restriction “ \exists ,” qualified cardinality restriction “ \leq_n ,” and local reflexivity “Self.” The top concept “ \top ” can be seen as “thing,” the most general predicate. The bottom concept “ \perp ” can be seen as a type with no instances, and it is useful in the specification of negative constraints, e.g., one can state that *Object* is pairwise disjoint with *Event* by stating that their intersection specializes “ \perp .”

The set of concepts is the smallest superset of \mathbf{C} containing “ \top ,” “ \perp ,” “ $\neg C$,” “ $C \sqcup D$,” “ $C \sqcap D$,” “ $\forall R.C$,” “ $\exists R.C$,” “ $\leq_n R.C$,” and “ $\exists R.\text{Self}$,” where C, D are concepts, R is a role, and n is a numeral denoting a natural number.

In DL, the terminological knowledge is expressed by a TBox, which contains axioms referring to concepts and roles.

Definition 3 (General concept inclusion axiom). Subsumptions between concepts can be represented by means of general concept inclusions (GCIs), which use the operator “ \sqsubseteq .” For instance, “*Person* \sqsubseteq *Animal*.” If C and D are concepts, then “ $C \sqsubseteq D$ ” is a GCI.

Definition 4 (Role assertion). One can specify certain constraints on roles by means of role assertions. For instance, if R and S are roles, then symmetry “ $\text{Sym}(R)$,” asymmetry “ $\text{Asy}(R)$,”¹² transitivity “ $\text{Tra}(R)$,” reflexivity “ $\text{Ref}(R)$,” irreflexivity “ $\text{Irr}(R)$,”¹³ and pairwise disjointness “ $\text{Dis}(R, S)$ ” are role assertions.

Definition 5 (Role composition). Role composition is represented by means of the construct “ \circ ,” e.g., the “maternal grandfather of” relation can be specified as “*fatherOf* \circ *motherOf*.”¹⁴ A string “ $\tau \circ R$ ” is a role composition iff R is a role, and τ is a role or role composition.

Definition 6 (Role inclusion axiom). Subsumption between binary relations can be represented by means of role inclusion axioms (RIAs), which use the operator “ \sqsubseteq .” For instance, “*motherOf* \sqsubseteq *parentOf*.” If R is a role and τ is a role or role composition not including the universal role U , then “ $\tau \sqsubseteq R$ ” is a RIA.

¹⁰By “name,” we mean a finite string.

¹¹We use the symbols “ \ulcorner ” and “ \urcorner ” to denote quasi-quotation. Moreover, note that the role inversion operator “ $^{-1}$ ” can only be applied to role names. So, one cannot write “ R^{-1} ” meaning “ R .”

¹²The construct “Asy” is called “antisymmetry” in Horrocks et al. (2006, p. 59), but we think that the provided semantics of “Asy” is of asymmetry.

¹³Note that irreflexivity and asymmetry usually refer to a relation over a set, i.e., a relation where the domain and the codomain are the same. However, both *SR_{OLQ}* operators “Irr” and “Asy” can be applied to a role having different domain and codomain.

¹⁴In Horrocks et al. (2006, p. 58), role composition is denoted by the juxtaposition of roles, e.g., “*fatherOfmotherOf*.” Here, we use the symbol “ \circ ” for clarity.

Note that $\lceil \text{Sym}(R) \rceil$ is equivalent to $\lceil R^- \sqsubseteq R \rceil$, $\lceil \text{Tra}(R) \rceil$ is to $\lceil R \circ R \sqsubseteq R \rceil$, and $\lceil \text{Irr}(R) \rceil$ is to $\lceil \top \sqsubseteq \neg \exists R.\text{Self} \rceil$. Finally, while $\lceil \text{Ref}(R) \rceil$ is equivalent to $\lceil \top \sqsubseteq \exists R.\text{Self} \rceil$, the form $\lceil \text{Ref}(R) \rceil$ is preferred since $SR\mathcal{O}IQ$ poses no structural constraint on it (see Section 8).

Definition 7 (TBox). A role box (RBox) is a finite set of role assertions or RIAs. A concept box (CBox) is a finite set of GCIs. A TBox is the union of an RBox and a CBox.

Since we do not use *nominals* in this work (the “ \mathcal{O} ” in “ $SR\mathcal{O}IQ$ ”), one could argue that our theories actually fit in the smaller logic $SRIQ$ (Horrocks et al., 2005). However, the presentation of $SRIQ$ in Horrocks et al. (2005) does not provide the construct for asymmetry “Asy,” which we use, and that cannot be enforced by means of RIAs (Horrocks et al., 2006, p. 59). $SRIQ$ provides constructs for symmetry “Sym” and transitivity “Tra” (which can be expressed by means of RIAs), and reflexivity “Ref,” irreflexivity “Irr” and pairwise disjointness “Dis” (which cannot be expressed by means of RIAs) (Horrocks et al., 2005, p. 5).

4.2. The set \mathcal{T}

First, we present axioms that correspond to Figure 1 and Figure 2, then we discuss the axioms of the different subtheories: Universals and individuals (U1)–(U5); Event mereology (M1)–(M9); Events as manifestations of object dispositions (D1’)–(D9); On the participation of objects in events (P1)–(P5); Temporal relations between events (T1)–(T14’); World changes and situations (S1’)–(S8); Processual roles (R1)–(R2); Qualities and quality structures (Q1)–(Q3); and The creation, termination and change of objects (C1)–(C3).

4.2.1. Formalization of Figures 1 and 2

Below, we specify the subsumptions between classes or relations in Figure 1 and Figure 2.

- | | |
|---|---|
| a1 Urelement $\sqsubseteq \top$ | a15 Object \sqsubseteq Endurant |
| a2 Set $\sqsubseteq \top$ | a16 Trope \sqsubseteq Endurant |
| a3 Universal \sqsubseteq Urelement | a17 Disposition \sqsubseteq Trope |
| a4 Individual \sqsubseteq Urelement | a18 Quality \sqsubseteq Trope |
| a5 QualityUniversal \sqsubseteq Universal | a19 TimePoint \sqsubseteq AbstractIndividual |
| a6 ObjectUniversal \sqsubseteq Universal | a20 Quale \sqsubseteq AbstractIndividual |
| a7 EventUniversal \sqsubseteq Universal | a21 QualityStructure \sqsubseteq Set |
| a8 ProcessualRole \sqsubseteq ObjectUniversal | a22 plays \sqsubseteq instantiates |
| a9 ParticipationUniversal \sqsubseteq EventUniversal | a23 Situation \sqsubseteq ConcreteIndividual |
| a10 ConcreteIndividual \sqsubseteq Individual | a24 Fact \sqsubseteq Situation |
| a11 AbstractIndividual \sqsubseteq Individual | a25 AtomicEvent \sqsubseteq Event |
| a12 Event \sqsubseteq ConcreteIndividual | a26 ComplexEvent \sqsubseteq Event |
| a13 Endurant \sqsubseteq ConcreteIndividual | a27 directlyCauses \sqsubseteq causes |
| a14 Participation \sqsubseteq Event | |

The pairwise disjointness constraints between classes:

- | | |
|--|---|
| a28 (Urelement \sqcap Set) $\sqsubseteq \perp$ | a30 (ConcreteIndividual \sqcap AbstractIndividual) $\sqsubseteq \perp$ |
| a29 (Universal \sqcap Individual) $\sqsubseteq \perp$ | a31 (QualityUniversal \sqcap ObjectUniversal) $\sqsubseteq \perp$ |

- | | |
|---|--|
| a32 (QualityUniversal \sqcap EventUniversal) $\sqsubseteq \perp$ | a37 (Disposition \sqcap Quality) $\sqsubseteq \perp$ |
| a33 (ObjectUniversal \sqcap EventUniversal) $\sqsubseteq \perp$ | a38 (Endurant \sqcap Situation) $\sqsubseteq \perp$ |
| a34 (Endurant \sqcap Event) $\sqsubseteq \perp$ | a39 (Event \sqcap Situation) $\sqsubseteq \perp$ |
| a35 (TimePoint \sqcap Quale) $\sqsubseteq \perp$ | a40 (AtomicEvent \sqcap ComplexEvent) $\sqsubseteq \perp$ |
| a36 (Object \sqcap Trope) $\sqsubseteq \perp$ | |

The completeness constraints that, together with pairwise disjointness, define class partitions:

- a41** $\top \sqsubseteq (\text{Urelement} \sqcup \text{Set})$
a42 $\text{Urelement} \sqsubseteq (\text{Universal} \sqcup \text{Individual})$
a43 $\text{Individual} \sqsubseteq (\text{ConcreteIndividual} \sqcup \text{AbstractIndividual})$
a44 $\text{Event} \sqsubseteq (\text{AtomicEvent} \sqcup \text{ComplexEvent})$

We formalize the domain, codomain and cardinality constraints of relations according to their orientations in the class diagrams presented in Figure 1 and Figure 2. Below, the domain constraints.

- | | |
|---|---|
| a45 $\exists \text{instantiates}.\top \sqsubseteq \text{Individual}$ | a58 $\exists \text{createdBy}.\top \sqsubseteq \text{Object}$ |
| a46 $\exists \text{memberOf}.\top \sqsubseteq \text{Quale}$ | a59 $\exists \text{terminatedBy}.\top \sqsubseteq \text{Object}$ |
| a47 $\exists \text{inducedBy}.\top \sqsubseteq \text{ProcessualRole}$ | a60 $\exists \text{changedBy}.\top \sqsubseteq \text{Object}$ |
| a48 $\exists \text{qualeOf}.\top \sqsubseteq \text{Quale}$ | a61 $\exists \text{activates}.\top \sqsubseteq \text{Situation}$ |
| a49 $\exists \text{plays}.\top \sqsubseteq \text{Object}$ | a62 $\exists \text{exclusivelyDependsOn}.\top \sqsubseteq \text{Event}$ |
| a50 $\exists \text{associatedWith}.\top \sqsubseteq \text{QualityStructure}$ | a63 $\exists \text{participationOf}.\top \sqsubseteq \text{Participation}$ |
| a51 $\exists \text{inheresIn}.\top \sqsubseteq \text{Trope}$ | a64 $\exists \text{dependsOn}.\top \sqsubseteq \text{AtomicEvent}$ |
| a52 $\exists \text{precedes}.\top \sqsubseteq \text{TimePoint}$ | a65 $\exists \text{mereologicallyOverlaps}.\top \sqsubseteq \text{Event}$ |
| a53 $\exists \text{beginPoint}.\top \sqsubseteq \text{Event}$ | a66 $\exists \text{hasPart}.\top \sqsubseteq \text{ComplexEvent}$ |
| a54 $\exists \text{endPoint}.\top \sqsubseteq \text{Event}$ | a67 $\exists \text{manifestedBy}.\top \sqsubseteq \text{Disposition}$ |
| a55 $\exists \text{directlyCauses}.\top \sqsubseteq \text{Event}$ | a68 $\exists \text{triggers}.\top \sqsubseteq \text{Situation}$ |
| a56 $\exists \text{causes}.\top \sqsubseteq \text{Event}$ | a69 $\exists \text{bringsAbout}.\top \sqsubseteq \text{Event}$ |
| a57 $\exists \text{presentIn}.\top \sqsubseteq \text{Endurant}$ | a70 $\exists \text{obtainsIn}.\top \sqsubseteq \text{Situation}$ |

The codomain constraints:

- | | |
|--|---|
| a71 $\top \sqsubseteq \forall \text{instantiates}.\text{Universal}$ | a81 $\top \sqsubseteq \forall \text{directlyCauses}.\text{Event}$ |
| a72 $\top \sqsubseteq \forall \text{memberOf}.\text{QualityStructure}$ | a82 $\top \sqsubseteq \forall \text{causes}.\text{Event}$ |
| a73 $\top \sqsubseteq \forall \text{inducedBy}.\text{ParticipationUniversal}$ | a83 $\top \sqsubseteq \forall \text{presentIn}.\text{Situation}$ |
| a74 $\top \sqsubseteq \forall \text{qualeOf}.\text{Quality}$ | a84 $\top \sqsubseteq \forall \text{createdBy}.\text{Event}$ |
| a75 $\top \sqsubseteq \forall \text{plays}.\text{ProcessualRole}$ | a85 $\top \sqsubseteq \forall \text{terminatedBy}.\text{Event}$ |
| a76 $\top \sqsubseteq \forall \text{associatedWith}.\text{QualityUniversal}$ | a86 $\top \sqsubseteq \forall \text{changedBy}.\text{Event}$ |
| a77 $\top \sqsubseteq \forall \text{inheresIn}.\text{ConcreteIndividual}$ | a87 $\top \sqsubseteq \forall \text{activates}.\text{Disposition}$ |
| a78 $\top \sqsubseteq \forall \text{precedes}.\text{TimePoint}$ | a88 $\top \sqsubseteq \forall \text{exclusivelyDependsOn}.\text{Object}$ |
| a79 $\top \sqsubseteq \forall \text{beginPoint}.\text{TimePoint}$ | a89 $\top \sqsubseteq \forall \text{participationOf}.\text{Object}$ |
| a80 $\top \sqsubseteq \forall \text{endPoint}.\text{TimePoint}$ | a90 $\top \sqsubseteq \forall \text{dependsOn}.\text{Object}$ |

a91 $\top \sqsubseteq \forall \text{mereologicallyOverlaps.Event}$ **a94** $\top \sqsubseteq \forall \text{triggers.Event}$
a92 $\top \sqsubseteq \forall \text{hasPart.Event}$ **a95** $\top \sqsubseteq \forall \text{bringsAbout.Situation}$
a93 $\top \sqsubseteq \forall \text{manifestedBy.AtomicEvent}$ **a96** $\top \sqsubseteq \forall \text{obtainsIn.TimePoint}$

The domain cardinality constraints:

a97 Individual $\sqsubseteq (\geq 1 \text{instantiates.Universal})$
a98 Quale $\sqsubseteq (= 1 \text{memberOf.QualityStructure})$
a99 ProcessualRole $\sqsubseteq (= 1 \text{inducedBy.ParticipationUniversal})$
a100 QualityStructure $\sqsubseteq (= 1 \text{associatedWith.QualityUniversal})$
a101 Trope $\sqsubseteq (= 1 \text{inheresIn.ConcreteIndividual})$
a102 Event $\sqsubseteq (= 1 \text{beginPoint.TimePoint})$
a103 Event $\sqsubseteq (= 1 \text{endPoint.TimePoint})$
a104 Event $\sqsubseteq (\leq 1 \text{exclusivelyDependsOn.Object})$
a105 Participation $\sqsubseteq (= 1 \text{participationOf.Object})$
a106 AtomicEvent $\sqsubseteq (= 1 \text{dependsOn.Object})$
a107 ComplexEvent $\sqsubseteq (\geq 2 \text{hasPart.Event})$
a108 Event $\sqsubseteq (= 1 \text{bringsAbout.Situation})$
a109 Situation $\sqsubseteq (\leq 1 \text{obtainsIn.TimePoint})$
a110 Fact $\sqsubseteq (= 1 \text{obtainsIn.TimePoint})$

The codomain cardinality constraints:

a111 Universal $\sqsubseteq (\geq 1 \text{instantiates}^-. \text{Individual})$
a112 QualityStructure $\sqsubseteq (\geq 1 \text{memberOf}^-. \text{Quale})$
a113 ParticipationUniversal $\sqsubseteq (\geq 1 \text{inducedBy}^-. \text{ProcessualRole})$
a114 Quality $\sqsubseteq (\geq 1 \text{qualeOf}^-. \text{Quale})$
a115 ProcessualRole $\sqsubseteq (\geq 1 \text{plays}^-. \text{Object})$
a116 QualityUniversal $\sqsubseteq (\geq 1 \text{associatedWith}^-. \text{QualityStructure})$
a117 Object \sqcup Event $\sqsubseteq (\geq 1 \text{inheresIn}^-. \text{Trope})$
a118 AtomicEvent $\sqsubseteq (= 1 \text{manifestedBy}^-. \text{Disposition})$
a119 Event $\sqsubseteq (= 1 \text{triggers}^-. \text{Situation})$
a120 TimePoint $\sqsubseteq (\geq 1 \text{obtainsIn}^-. \text{Situation})$

We have the domain constraint (a55) even though it is entailed by $\{(a27), (a56)\}$; and similarly for the codomain constraint (a81), which is entailed by $\{(a27), (a82)\}$. This is because we do not know (yet) if the structural restrictions make (a27) incompatible with other axioms. For the same reason, we keep (a92) when we introduce (a130), even though $\{(a91), (a130)\}$ entails (a92). Similarly for (a92) being also entailed by $\{(a65), (a131)\}$; (a90) being entailed by $\{(a88), (a140)\}$; and (a89) being entailed by $\{(a88), (a143)\}$.

4.2.2. Universals and individuals: (a121)–(a124)

According to UFO-A, the classes `Urelement` and `Set` partition `Thing`, while `Individual` and `Universal` partition `Urelement` (Guizzardi, 2005, Figure 6–12). One can see in (a1), (a2) and (a41) that we mapped `Thing` to the “ \top ” DL concept.

Universals in UFO-A (Guizzardi, 2005, Axiom 22, p. 221) are assumed to satisfy the *principle of instantiation* (Armstrong, 1989, p. 75), which means that it must be possible (in a modal sense) for a universal to have instances. We partially capture this assumption by means of (a111).

Concerning the relation `instantiates` and its specializations, (U1)–(U4) can be specified in *SR_{OLQ}* as (a121)–(a124), respectively. (U5), the constraint that all `instantiates` relationships from `Objects` to `ProcessualRoles` are `plays` relationships, seems inexpressible in *SR_{OLQ}* (we can only think of a solution using variables). For this reason, we add an axiom to guarantee that all `ProcessualRoles` are played by at least one `Object` (a115), which is a theorem of our first-order theory (t1).

a121	$\exists \text{instantiates. QualityUniversal} \sqsubseteq \text{Quality}$	(U1)
a122	$\exists \text{instantiates. ObjectUniversal} \sqsubseteq \text{Object}$	(U2)
a123	$\exists \text{instantiates. EventUniversal} \sqsubseteq \text{Event}$	(U3)
a124	$\exists \text{instantiates. ParticipationUniversal} \sqsubseteq \text{Participation}$	(U4)

Moreover, given that the domain of `instantiates` (`Individual`) is disjoint with its codomain (`Universal`, see (a29)), then `instantiates` is (implicitly) asymmetric and hence irreflexive. We will see in Section 5.1 that, due to the structural restrictions of *SR_{OLQ}*, asserting either $\lceil \text{Irr}(R) \rceil$ or $\lceil \text{Asy}(R) \rceil$ require R to be *simple*. Since stating the disjointness of the domain and codomain of R (as in (a29)) has no consequences w.r.t. structural restrictions, we only state $\lceil \text{Irr}(R) \rceil$ or $\lceil \text{Asy}(R) \rceil$ when the domain of R is not disjoint with its codomain.

4.2.3. Event mereology: (a125)–(a133)

The inexpressibility of mereological notions in *SR_{OLQ}* was already recognized in (Horrocks et al., 2006, p. 67), as its syntactic constraints forbid roles to be asymmetric and transitive. For (M1), the axiom (a25) already guarantees that `AtomicEvents` are `Events`, and we capture the necessary condition (indicated by “–”) of `AtomicEvent` in (a125), and the sufficient condition (indicated by “+”) in (a126). An absence of “–” or “+” means equivalence. (M2) is guaranteed by (a40) and (a44). (M4) is captured by (a127), which also captures (M3) since asymmetry entails irreflexivity.¹⁵ (M5) is captured by (a128).

a125	$\text{AtomicEvent} \sqsubseteq \neg \exists \text{hasPart. Event}$	(M1)–	a127	$\text{Asy}(\text{hasPart})$	(M3),(M4)
a126	$\text{Event} \sqcap \neg \exists \text{hasPart. Event} \sqsubseteq \text{AtomicEvent}$	(M1)+	a128	$\text{Tra}(\text{hasPart})$	(M5)

For (M7’), (a129)–(a132) represent the sufficient conditions for `mereologicallyOverlaps`. The necessary condition for this role seems inexpressible, as the right-hand side of a role inclusion axiom (RIA) can only have a role name. Finally, we assert the symmetry of `mereologicallyOverlaps` by (a133).

a129	$\text{Event} \sqsubseteq \exists \text{mereologicallyOverlaps. Self}$	(M7’)+
a130	$\text{hasPart} \sqsubseteq \text{mereologicallyOverlaps}$	(M7’)+
a131	$\text{hasPart}^- \sqsubseteq \text{mereologicallyOverlaps}$	(M7’)+

¹⁵Since asymmetry entails irreflexivity, and both “Irr” and “Asy” obey the very same *SR_{OLQ}* syntactic constraints (as we will see in Section 5.1, “Irr” and “Asy” only apply to simple roles), then whenever a role is asymmetric, there is no benefit in also asserting its irreflexivity.

a132 $\text{hasPart} \circ \text{hasPart}^- \sqsubseteq \text{mereologicallyOverlaps}$ (M7')+
a133 $\text{Sym}(\text{mereologicallyOverlaps})$ (M7)'-

(M6) seems inexpressible even by assuming a *SRIOQ* definition of *mereologicallyOverlaps*. Intuitively, (M6) would be captured by the expressions “ $\text{hasAtLeastTwoDisjointParts} \equiv \text{hasPart} \circ \text{notMereologicallyOverlap} \circ \text{hasPart}^-$ ” and “ $\text{Ref}(\text{hasAtLeastTwoDisjointParts})$,” which enforce the existence of a path from a whole, then to a part, then to a non-overlapping entity that is also a part, so that the path comes to a cycle by an inverse of the parthood relation. However, the first formula is excluded by the syntax of *SRIOQ*, which disallows role compositions on the right-hand side of role inclusions. Moreover, the role *notMereologicallyOverlap* used in this composition is inexpressible in this logic. One could think of trying “ $\text{Dis}(\text{mereologicallyOverlaps}, \text{notMereologicallyOverlap})$,” but *SRIOQ* cannot express that one relation would be the set-complement of the other, i.e., the meaning of the non-*SRIOQ* formula “ $U \sqsubseteq \text{mereologicallyOverlaps} \sqcup \text{notMereologicallyOverlap}$.” More precisely, *SRIOQ* does not allow complementation of roles. Finally, it seems that neither (M8) nor (M9) are expressible, since there is no way to capture an equality on “generic individuals,” as there are no first-order variables in this language. Since we captured neither weak supplementation (M6) nor strong supplementation (M8) in *SRIOQ*, (a107) poses at least a cardinality constraint ensuring that every *ComplexEvent* has at least two parts.

4.2.4. Events as manifestations of object dispositions: (a134)–(a139)

(D1') is guaranteed by (a17) and (a101), while (D2) is captured by (a118). As the right-hand side of a RIA can only have a role name, it seems that only the sufficient condition of (D3) can be formalized as (a134). (D4) is captured by (a135). (D5) is guaranteed by (a117). (D6) is captured by (a136). On (D7), since *SRIOQ* provides no role negation, it seems impossible to capture intransitivity. (D8) is captured by (a137) and (a138), while (D9) is captured by (a139).

a134 $\text{activates} \circ \text{manifestedBy} \sqsubseteq \text{triggers}$ (D3) +
a135 $\text{manifestedBy}^- \circ \text{inheresIn} \sqsubseteq \text{dependsOn}$ (D4)
a136 $\text{Asy}(\text{inheresIn})$
a137 $\text{manifestedBy} \circ \text{triggers}^- \sqsubseteq \text{presentIn}$ (D8) –
a138 $\text{manifestedBy} \circ \text{bringsAbout} \sqsubseteq \text{presentIn}$ (D8) –
a139 $\text{inheresIn}^- \circ \text{presentIn} \sqsubseteq \text{presentIn}$ (D9)

4.2.5. On the participation of objects in events: (a140)–(a142)

(P1) is guaranteed by (a106). For (P2), it seems that *SRIOQ* cannot express the subset of the *exclusivelyDependsOn* relation that has *AtomicEvent* as its domain. As the domain of *dependsOn* is *AtomicEvent*, it seems that only the necessary condition for *dependsOn* (a140) is expressible, but not the opposite role inclusion. Similarly for (P3) and the subset of the *exclusivelyDependsOn* relation that has *ComplexEvent* as its domain. On (P4), the necessary condition for being a *Participation* is captured by {(a14),(a141)}, while the sufficient condition for being a *Participation* is captured by (a142).

a140 $\text{dependsOn} \sqsubseteq \text{exclusivelyDependsOn}$ (P2)–
a141 $\text{Participation} \sqsubseteq =\text{lexclusivelyDependsOn.Object}$ (P4)–
a142 $\text{Event} \sqcap (= \text{lexclusivelyDependsOn.Object}) \sqsubseteq \text{Participation}$ (P4)+

Concerning (P5), since *participationOf* and *exclusivelyDependsOn* have the same codomain—*Object*—and the domain of *participationOf* (*Participation*) specializes the domain of *exclusivelyDependsOn* (*Event*), the left-to-right implication can be captured in (a143). On the other hand, it

seems impossible to capture the right-to-left implication, i.e., `exclusivelyDependsOn` with the domain constrained to `Participation` specializes `participationOf`.

a143 `participationOf` \sqsubseteq `exclusivelyDependsOn` (P5)+

4.2.6. Temporal relations between events: (a144)–(a174)

Batsakis et al. (2017) discuss the impossibility of expressing Allen’s time interval relations (Allen, 1983) in *SR_{OTQ}*. Therefore, it seems that the best one can do is to provide partial axiomatizations of UFO-B’s temporal relations between Events. (Guizzardi et al., 2013, p. 331) informally defines the domain and codomain of these relations as `Event` (a144)–(a157).

a144 \exists <code>before</code> . \top \sqsubseteq <code>Event</code>	a148 \exists <code>starts</code> . \top \sqsubseteq <code>Event</code>	a152 \exists <code>finishes</code> . \top \sqsubseteq <code>Event</code>
a145 \top \sqsubseteq \forall <code>before</code> . <code>Event</code>	a149 \top \sqsubseteq \forall <code>starts</code> . <code>Event</code>	a153 \top \sqsubseteq \forall <code>finishes</code> . <code>Event</code>
a146 \exists <code>meets</code> . \top \sqsubseteq <code>Event</code>	a150 \exists <code>during</code> . \top \sqsubseteq <code>Event</code>	a154 \exists <code>equals</code> . \top \sqsubseteq <code>Event</code>
a147 \top \sqsubseteq \forall <code>meets</code> . <code>Event</code>	a151 \top \sqsubseteq \forall <code>during</code> . <code>Event</code>	a155 \top \sqsubseteq \forall <code>equals</code> . <code>Event</code>
a156 \exists <code>temporallyOverlaps</code> . \top \sqsubseteq <code>Event</code>	a157 \top \sqsubseteq \forall <code>temporallyOverlaps</code> . <code>Event</code>	

On `precedes`, (T2) (and (T1)) is captured by (a158), while (T3) is by (a159). Totality (T4) seems inexpressible (at the time, we can only think of a solution using variables). (T5’) is guaranteed by (a102) and (a103). (T6’) is captured by (a160).

a158 <code>Asy</code> (<code>precedes</code>)	(T1),(T2)	a159 <code>Tra</code> (<code>precedes</code>) (T3)
a160 <code>beginPoint</code> ⁻ \circ <code>endPoint</code> \sqsubseteq <code>precedes</code>		(T6’)
a161 <code>endPoint</code> \circ <code>precedes</code> \circ <code>beginPoint</code> ⁻ \sqsubseteq <code>before</code>		(T7’)+
a162 <code>endPoint</code> \circ <code>beginPoint</code> ⁻ \sqsubseteq <code>meets</code>		(T8’)+

Concerning the Allen relations (T7’)–(T13’), their necessary conditions seem inexpressible because the right-hand side of a RIA can only have a role name. The sufficient conditions for `before` (T7’) and `meets` (T8’) are captured by (a161) and (a162), respectively. The sufficient conditions for `temporallyOverlaps` (T9’), `starts` (T10’), `during` (T11’), `finishes` (T12’), and `equals` (T13’), seem inexpressible, since they would require a conjunction of role compositions, see formulae (f1)–(f5), respectively. Finally, (T14’) would break even more syntactic constraints, see (f6).

f1	(<code>beginPoint</code> \circ <code>precedes</code> \circ <code>beginPoint</code> ⁻) \sqcap (<code>endPoint</code> \circ <code>precedes</code> ⁻ \circ <code>beginPoint</code> ⁻) \sqcap (<code>endPoint</code> \circ <code>precedes</code> \circ <code>endPoint</code> ⁻) \sqsubseteq <code>temporallyOverlaps</code>	(T9’)+
f2	(<code>beginPoint</code> \circ <code>beginPoint</code> ⁻) \sqcap (<code>endPoint</code> \circ <code>precedes</code> \circ <code>endPoint</code> ⁻) \sqsubseteq <code>starts</code>	(T10’)+
f3	(<code>beginPoint</code> \circ <code>precedes</code> ⁻ \circ <code>beginPoint</code> ⁻) \sqcap (<code>endPoint</code> \circ <code>precedes</code> \circ <code>endPoint</code> ⁻) \sqsubseteq <code>during</code>	(T11’)+
f4	(<code>endPoint</code> \circ <code>endPoint</code> ⁻) \sqcap (<code>beginPoint</code> \circ <code>precedes</code> ⁻ \circ <code>beginPoint</code> ⁻) \sqsubseteq <code>finishes</code>	(T12’)+
f5	(<code>beginPoint</code> \circ <code>beginPoint</code> ⁻) \sqcap (<code>endPoint</code> \circ <code>endPoint</code> ⁻) \sqsubseteq <code>equals</code>	(T13’)+
f6	<code>hasPart</code> \sqsubseteq (((<code>beginPoint</code> \circ <code>beginPoint</code> ⁻) \sqcup (<code>beginPoint</code> \circ <code>precedes</code> \circ <code>beginPoint</code> ⁻)) \sqcap ((<code>endPoint</code> \circ <code>endPoint</code> ⁻) \sqcup (<code>endPoint</code> \circ <code>precedes</code> ⁻ \circ <code>endPoint</code> ⁻)))	(T14’)

Since we could capture only a small fraction of the definitions of the Allen relations, we can at least pose some role assertions to capture some properties of these relations: `before`, `starts` and `during` are asymmetric (a163),(a165),(a167) and transitive (a164),(a166),(a168); `meets`, `temporallyOverlaps`,

and finishes are asymmetric (a169),(a170),(a171); and equals is an equivalence relation on its domain, i.e., reflexive (a172),¹⁶ symmetric (a173) and transitive (a174).

a163 Asy(before)	a167 Asy(during)	a171 Asy(finishes)
a164 Tra(before)	a168 Tra(during)	a172 Event $\sqsubseteq \exists \text{equals.Self}$
a165 Asy(starts)	a169 Asy(meets)	a173 Sym(equals)
a166 Tra(starts)	a170 Asy(temporallyOverlaps)	a174 Tra(equals)

4.2.7. World changes and situations: (a175)–(a179)

(S1') and (S2') seem inexpressible in *SR_{OLQ}*, since the right-hand side of a RIA can only have a role name (see (f7) and (f8)). (S3) and (S4) are guaranteed by (a119) and (a108), respectively. On (S5), the necessary condition for being a Fact is captured by (a110), while the sufficient condition is by (a175). On (S6), it is impossible to define a role as the composition of other roles; a role can only *include* a role composition. Therefore, we capture the sufficient condition for directlyCauses in (a176), but the necessary condition seems inexpressible. Only the sufficient conditions of (S7) seem expressible in *SR_{OLQ}*, as in (a27) and (a177); the necessary condition seems inexpressible due to the same reason as for (S1').

f7 triggers \sqsubseteq obtainsIn \circ beginPoint ⁻	f8 bringsAbout \sqsubseteq endPoint \circ obtainsIn ⁻	
a175 Situation $\sqcap (\geq 1 \text{ obtainsIn.TimePoint}) \sqsubseteq$ Fact		(S5)+
a176 bringsAbout \circ triggers \sqsubseteq directlyCauses		(S6)+
a177 Tra(causes)		(S7)+

From the strict linear ordering of TimePoints and (T6'), it follows that the beginPoint of an Event must strictly precedes its endPoint. From (S1'), (S2') and (S6) it follows that directlyCauses is asymmetric (a178). More strongly, it also follows that a chain of directlyCauses is acyclic. Since causes should be the transitive closure of directlyCauses, causes is asymmetric (a179), and transitive (a177).

a178 Asy(directlyCauses)	a179 Asy(causes)
---------------------------------	-------------------------

4.2.8. Processual roles: (a180)–(a181)

We provide both a necessary condition (R2) and a sufficient condition (R1) for an Object to plays a ProcessualRole. (R1) is translated to (a180) and (a181). Unfortunately, (R2) does not seem expressible in *SR_{OLQ}*—see (f9)—as *SR_{OLQ}* forbids both union of roles (and role compositions) and non-role-names on the right-hand side of a RIA.

a180 dependsOn ⁻ \circ instantiates \circ inducedBy ⁻ \sqsubseteq plays	(R1) –
a181 dependsOn ⁻ \circ hasPart ⁻ \circ instantiates \circ inducedBy ⁻ \sqsubseteq plays	(R1) –
f9 plays \sqsubseteq (dependsOn ⁻ \circ instantiates \circ inducedBy ⁻) \sqcup (dependsOn ⁻ \circ hasPart ⁻ \circ instantiates \circ inducedBy ⁻)	(R2)

¹⁶We do not use the role assertion “Ref(equals)” because its semantics in *SR_{OLQ}* is stronger than the usual definition of reflexivity, as it enforces the domain of the role name equals to be the whole domain of quantification, i.e., “ \top .” Strictly, the role assertion “Ref(*R*)” reads “the role name *R* contains the identity relation.”

4.2.9. Qualities and quality structures

Since it seems that first-order variables are needed to express (Q1), this axiom appears inexpressible in *SRIOQ*. Moreover, discarding the uniqueness constraint in (Q2), it could be captured by (f10), which is not a *SRIOQ* formula because *SRIOQ* forbids role compositions on the right-hand side of a RIA. It seems that the best one can do is to impose a cardinality constraint on the codomain of `qualeOf` in order to ensure that a `Quality` has at least one `Qual` (a114). Similarly to (Q2), (Q3) appears inexpressible in *SRIOQ*, see (f11).

f10 `instantiates` \circ `associatedWith` \sqsubseteq `qualeOf` \circ `memberOf` (Q2) –

f11 `qualeOf` \sqsubseteq `instantiates` \circ `associatedWith` \circ `memberOf` (Q3)

4.2.10. The creation, termination and change of objects: (a182)–(a185)

(C1) can be partially characterized by (a182). It seems inexpressible in its totality because there are neither role negations nor role disjunctions in *SRIOQ*. Similarly to (C1), (C2) and (C3) appear to be inexpressible in their totality in *SRIOQ*; see (a183) for (C2), and (a184) and (a185) for (C3).

a182 `createdBy` \circ `bringsAbout` \sqsubseteq `presentIn` (C1) –

a183 `terminatedBy` \circ `triggers` \sqsubseteq `presentIn` (C2) –

a184 `changedBy` \circ `bringsAbout` \sqsubseteq `presentIn` (C3) –

a185 `changedBy` \circ `triggers` \sqsubseteq `presentIn` (C3) –

5. UFO-B *SRIOQ* theories: the maximally structured subsets of \mathcal{T}

In this section, we determine all the maximally structured subsets of \mathcal{T} . Let us introduce, in Section 5.1, the two structural restrictions of *SRIOQ*, viz., regularity and simplicity.

5.1. The structural restrictions of *SRIOQ*

It is known that allowing an unrestricted use of roles in an `RBox` can easily lead to undecidability of reasoning (Horrocks et al., 2006, p. 58). For that reason, *SRIOQ* imposes some additional syntactic constraints—the structural restrictions known as *simplicity* and *regularity*—in the construction of axioms in a `TBox`.

Definition 8 (Simplicity). *We simplify the notion of simplicity presented in (Horrocks et al., 2006, p. 59) in order to qualify only role names in \mathbf{R} as simple, instead of qualifying also inverse roles.¹⁷ There are two forms of direct sources of non-simplicity for a role name R : (i) RIAs having a role composition τ on the left-hand side, like $\lceil \tau \sqsubseteq R \rceil$; and (ii) the role assertions $\lceil \text{Tra}(R) \rceil$ and $\lceil \text{Tra}(R^-) \rceil$. There is one form of indirect source of non-simplicity for a role name R : RIAs like $\lceil S \sqsubseteq R \rceil$ and $\lceil S^- \sqsubseteq R \rceil$, where S is a non-simple role name. Let “ \sqsubseteq^* ” denote the reflexive transitive closure of the inclusion relation \sqsubseteq over roles (i.e., excluding role compositions). A role name R is simple w.r.t. a `TBox` iff for any role name S such that $S \sqsubseteq^* R$ or $S^- \sqsubseteq^* R$, the `TBox` does not contain $\lceil \text{Tra}(S) \rceil$ or $\lceil \text{Tra}(S^-) \rceil$, and there is no role composition τ such that the `TBox` contains $\lceil \tau \sqsubseteq S \rceil$.*

¹⁷From the original definition of simplicity in (Horrocks et al., 2006, p. 59), the only case in which an inverse role R^- is non-simple is when the related role name R is non-simple. Therefore, the simplicity rules for role names can be rewritten in order to take into account also the cases where inverse roles would be considered non-simple.

The syntactic rule called “simplicity” forbids non-simple role names and inverses of non-simple role names to be used in concepts of the form $\lceil \exists R.Self \rceil$ and $\lceil \leq n R.C \rceil$, and role assertions of the form $\lceil Irr(R) \rceil$, $\lceil Asy(R) \rceil$, and $\lceil Dis(R, S) \rceil$.

Regularity enforces the set of RIAs to “induce” a constrained strict partial order on the set of roles, preventing a role hierarchy from containing cyclic dependencies that could lead to undecidability (Horrocks et al., 2006, p. 58).

Definition 9 (Regularity). A strict partial order \prec on the set of roles is called regular iff, for all roles R and S , the additional constraint of $\lceil S \prec R \leftrightarrow S^- \prec R^- \rceil$ holds. A RIA $\lceil \tau \sqsubseteq R^- \rceil$ is \prec -regular iff R is a role name and S_1, \dots, S_n are roles different from R and R^- such that:

- (1) $\tau = \lceil R \circ R^- \rceil$, or
- (2) $\tau = \lceil R^- \rceil$, or
- (3) $\tau = \lceil S_1 \circ \dots \circ S_n^- \rceil$ and $S_i \prec R$, for all $1 \leq i \leq n$, or
- (4) $\tau = \lceil R \circ S_1 \circ \dots \circ S_n^- \rceil$ and $S_i \prec R$, for all $1 \leq i \leq n$, or
- (5) $\tau = \lceil S_1 \circ \dots \circ S_n \circ R^- \rceil$ and $S_i \prec R$, for all $1 \leq i \leq n$.

A set of RIAs is regular iff there exists a regular order \prec such that each RIA in the set is \prec -regular. The syntactic rule called “regularity” ensures that the set of all RIAs in a TBox is regular.¹⁸

In the following, whenever we speak of a $SR\mathcal{OIQ}$ TBox, we will implicitly assume that it satisfies the two syntactic restrictions of simplicity and regularity, unless explicitly stated otherwise.

5.2. Incompatibilities between $SR\mathcal{OIQ}$ axioms

In order to track the incompatibilities between axioms in $\mathcal{T} = \{(a1), \dots, (a185)\}$ that lead to infringing the $SR\mathcal{OIQ}$ syntactic rules of *regularity* or *simplicity*, we build a very simple meta-level theory \mathcal{I} in propositional logic. By means of this meta-level theory, we define the problem of finding subsets of \mathcal{T} that satisfy regularity and simplicity as a SAT problem, which is decidable and can be addressed by many SAT solvers (Biere et al., 2009).

5.2.1. Incompatibilities due to regularity

We use the proposition “ $\mathbb{I}(i)$ ” to mean that the axiom (ai) is included in the TBox (i.e., “ \mathbb{I} ” serves as a “proposition builder”, whereas “ i ” behaves like an index in the proposition). Moreover, relationships like “ $r \prec s$ ” can also be seen as propositions. In order to capture the incompatibility between axioms due to regularity, we add a conditional statement for each RIA that, when included in the TBox, implies an ordering constraint. For instance, given an axiom i and roles R and S , the fact that the inclusion of

¹⁸In our implementation, we simplified the notion of regularity in the following way. Let us define a function that extracts the “base role name” of a role: given that the set of roles has only role names and inverses of role names and that $\lceil R^{- -} \rceil$ is not allowed in $SR\mathcal{OIQ}$, we define $base(R)$ for any role name R as $\lceil base(R) \triangleq R^- \rceil$ and $\lceil base(R^-) \triangleq R \rceil$. Since the right-hand side of a RIA can only have role names, from (1)-(5) there is no constraint on the regular order \prec enforcing a role to precede the inverse of a role name, i.e., no constraint of the form $S \prec R^-$. Therefore, instead of qualifying a strict partial order \prec as regular by means of “ $S \prec R \leftrightarrow S^- \prec R^-$,” one can restrict the domain of \prec to role names (instead of roles) and modify rules (3)-(5) by replacing “ $S_i \prec R$ ” with “ $base(S_i) \prec R$.” In this way, the size of the domain of \prec is halved, what could improve the performance of techniques for solving partial order constraints such as the one proposed in Codish et al. (2006).

axiom i in the TBox requires both relationships $R \prec S$ and $R^- \prec S$ to hold is captured by the propositional formula “ $\mathbb{I}(i) \rightarrow (R \prec S) \wedge (R^- \prec S)$.” The interested reader can find these conditions in Appendix A.

Codish et al. (2006) show how to encode, in propositional logic, constraints for a partial order over a set S by using $|S| \cdot \lceil \log_2 |S| \rceil$ propositional variables. In our case, since there are 66 roles, we need 462 propositional variables for our regularity constraints, which is impracticable for our computational resources (the size of the search space of the SAT problem is 2^{462}). Therefore, we manually dealt with regularity by showing two incompatibilities, and showing two regular orders of roles that reflect the inclusion of one or the other of the two choices of axioms, while obeying regularity for all the other axioms. More specifically, the axiom (a22) is incompatible with both (a180) and (a181), which is captured by the meta axiom (m1). Since we found two strict partial orders satisfying regularity, one for $\mathcal{T} \setminus \{(a180), (a181)\}$, and another for $\mathcal{T} \setminus \{(a22)\}$, it is guaranteed that there are no more incompatibilities due to regularity other than those excluded by (m1). The interested reader can find the respective partial orders in Figure 5b and Figure 5c in Appendix A.

$$\mathbf{m1} \quad \mathbb{I}(22) \rightarrow \neg(\mathbb{I}(180) \vee \mathbb{I}(181))$$

5.2.2. Incompatibilities due to simplicity

Given a role name R , we capture the fact that R is non-simple by using the “proposition builder” symbol “ $\bar{\mathbb{S}}$ ” to build propositions of the form “ $\bar{\mathbb{S}}(R)$.” We are then able to capture constraints such as “the inclusion of axiom i makes the role name R non-simple,” e.g., by the propositional formula “ $\mathbb{I}(i) \rightarrow \bar{\mathbb{S}}(R)$,” and “the non-simplicity of role name R forbids the inclusion of axiom i ,” e.g., by “ $\bar{\mathbb{S}}(R) \rightarrow \neg\mathbb{I}(i)$.”

Since our set \mathcal{T} of *SR_{OLQ}* axioms is finite—there are 185 axioms—and the set \mathbf{R} of role names is finite—there are 33 role names—then the set of proposition symbols built by means of \mathbb{I} and $\bar{\mathbb{S}}$ is also finite: $185 + 33 = 218$ proposition symbols (also called propositional variables).

Unfortunately, our computational resources could not handle solving a SAT problem with search space at the magnitude of 2^{218} ($> 10^{65}$). We address this issue by deriving incompatibility meta-axioms from the two types of constraints discussed above. By the transitivity of the logical implication operator on pairs of propositional formulae like “ $\mathbb{I}(i) \rightarrow \bar{\mathbb{S}}(r)$ ” and “ $\bar{\mathbb{S}}(r) \rightarrow \neg\mathbb{I}(j)$,” we derive a meta-axiom “ $\mathbb{I}(i) \rightarrow \neg\mathbb{I}(j)$.” In other words, we create a propositional meta-axiom for each *SR_{OLQ}* axiom that, when included in the TBox, forbids the inclusion of another axiom in the TBox (by making a role non-simple).

Indirect sources of non-simplicity, like “ $(\mathbb{I}(i) \wedge \bar{\mathbb{S}}(r)) \rightarrow \bar{\mathbb{S}}(s)$,” are more difficult to handle. In the special case where the set of role names made non-simple by direct sources of non-simplicity is disjoint from the set of role names made non-simple by indirect sources of non-simplicity (i.e., the set of role names appearing in the antecedents of the simplicity meta-axioms is disjoint from the set of role names appearing in the consequents), one can replace $\bar{\mathbb{S}}(r)$ by a disjunction of its direct sources of non-simplicity: “ $(\mathbb{I}(i) \wedge (\mathbb{I}(j) \vee \dots \vee \mathbb{I}(k))) \rightarrow \bar{\mathbb{S}}(s)$.” In our tool, we first check such disjointness before applying this reduction.

Since our propositional meta-level theory \mathcal{I} could be simplified by this technique, \mathcal{I} does not include propositions about the simplicity of a role or about partial order constraints. \mathcal{I} only concerns incompatibilities between axioms, so that only axioms like “ $\mathbb{I}(i) \rightarrow \neg\mathbb{I}(j)$ ” are included. With this simplification, \mathcal{I} requires only 40 propositional variables. Also, since now we use only one “proposition builder”—“ \mathbb{I} ”—, we may substitute the notation “ $\mathbb{I}(i)$ ” with “ i ” when the context precludes ambiguity (e.g., in Figure 3 in Section 5.3).

In what follows, we list the meta-axioms that capture incompatibilities due to simplicity. The interested reader can find the full derivations in Appendix B.

$$\mathbf{m2} \quad (\mathbb{I}(132) \vee (\mathbb{I}(128) \wedge (\mathbb{I}(130) \vee \mathbb{I}(131)))) \rightarrow \neg\mathbb{I}(129)$$

- m3** $\mathbb{I}(134) \rightarrow \neg\mathbb{I}(119)$
m4 $\mathbb{I}(135) \rightarrow \neg\mathbb{I}(106)$
m5 $(\mathbb{I}(160) \vee \mathbb{I}(159)) \rightarrow \neg\mathbb{I}(158)$
m6 $(\mathbb{I}(161) \vee \mathbb{I}(164)) \rightarrow \neg\mathbb{I}(163)$
m7 $\mathbb{I}(162) \rightarrow \neg\mathbb{I}(169)$
m8 $\mathbb{I}(176) \rightarrow \neg\mathbb{I}(178)$
m9 $(\mathbb{I}(180) \vee \mathbb{I}(181)) \rightarrow \neg\mathbb{I}(115)$
m10 $\mathbb{I}(128) \rightarrow \neg(\mathbb{I}(107) \vee \mathbb{I}(127))$
m11 $(\mathbb{I}(177) \vee (\mathbb{I}(27) \wedge \mathbb{I}(176))) \rightarrow \neg\mathbb{I}(179)$
m12 $\mathbb{I}(166) \rightarrow \neg\mathbb{I}(165)$
m13 $\mathbb{I}(168) \rightarrow \neg\mathbb{I}(167)$
m14 $\mathbb{I}(174) \rightarrow \neg\mathbb{I}(172)$
m15 $(\mathbb{I}(135) \wedge \mathbb{I}(140)) \rightarrow \neg(\mathbb{I}(104) \vee \mathbb{I}(141) \vee \mathbb{I}(142))$
m16 $(\mathbb{I}(22) \wedge (\mathbb{I}(180) \vee \mathbb{I}(181))) \rightarrow \neg(\mathbb{I}(97) \vee \mathbb{I}(111))$

5.3. Finding the structured subsets of \mathcal{T}

Let $\mathcal{I} = \{(m1), \dots, (m16)\}$ be the set of axioms of our meta-level propositional theory about the *SROTQ* structural constraints on $\mathcal{T} = \{(a1), \dots, (a185)\}$. As said before, the proposition “ $\mathbb{I}(i)$ ” means that the axiom (ai) is included in the TBox. Since \mathcal{T} has 185 axioms, there are also 185 corresponding propositions. However, not all of them are required, we only need the 40 propositions mentioned in \mathcal{I} , which we collect in the set $\{\mathbb{I}(22), \mathbb{I}(27), \mathbb{I}(97), \mathbb{I}(104), \mathbb{I}(106), \mathbb{I}(107), \mathbb{I}(111), \mathbb{I}(115), \mathbb{I}(119), \mathbb{I}(127) - \mathbb{I}(132), \mathbb{I}(134), \mathbb{I}(135), \mathbb{I}(140) - \mathbb{I}(142), \mathbb{I}(158) - \mathbb{I}(169), \mathbb{I}(172), \mathbb{I}(174), \mathbb{I}(176) - \mathbb{I}(181)\}$. Let \mathcal{U} be the set of axioms (ai) corresponding to those propositions $\mathbb{I}(i)$. In a sense, \mathcal{U} collects the “unsafe axioms of \mathcal{T} ,” where the axioms not mentioned in \mathcal{I} would be considered “safe,” since together they are known to obey the structural restrictions of regularity and simplicity. This means that the set $\mathcal{T} \setminus \mathcal{U}$ obeys both regularity and simplicity (is structured), being a *SROTQ* theory that partially captures UFO-B.

However, $\mathcal{T} \setminus \mathcal{U}$ does not seem to maximize precision w.r.t. the formalization of UFO-B in FOL. In order to increase precision, we shall find the subsets of \mathcal{U} that are structured, and therefore need not be excluded from \mathcal{T} . Nevertheless, the search space related to such subsets is $2^{|\mathcal{U}|} = 2^{40} = 1\,099\,511\,627\,776$. In practice, each solution of the SAT problem related to \mathcal{I} represents a structured subset of \mathcal{T} .¹⁹

A further inspection reveals that, within \mathcal{I} , there are independent subsets of incompatibility rules. For instance, note that $(m3)$ is about $\mathbb{I}(119)$ and $\mathbb{I}(134)$, and that no other meta-axiom mentions neither $\mathbb{I}(119)$ nor $\mathbb{I}(134)$. This means that no other meta-axiom influences the possible truth-values of $\mathbb{I}(119)$ and $\mathbb{I}(134)$. For example, given a model of $(m2)$ in which $\mathbb{I}(119)$ is true, there must be another model that only differs from the first by $\mathbb{I}(119)$ being false. $\mathbb{I}(119)$ and $(m2)$ are independent, and the same for $\mathbb{I}(134)$.

Figure 3 shows the independent subsets of the incompatibility rules, abstracting the meta-axioms into sets of the propositions that they mention. In Figure 3, \mathcal{I} is partitioned into 11 independent sub-theories \mathcal{I}_i , with $(0 \leq i \leq 10)$. The idea is that instead of monolithically evaluating all the incompatibility rules together, it is possible to individually evaluate each independent subset of incompatibility rules, and then combine the results later. More specifically, a solution for \mathcal{I}_i represents a structured subset of \mathcal{U} , and since the sub-theories \mathcal{I}_i are independent, the consecutive union of one solution per \mathcal{I}_i also represents a structured subset of \mathcal{U} .

¹⁹Solving a SAT problem with 40 proposition symbols is still unfeasible for our computational resources.

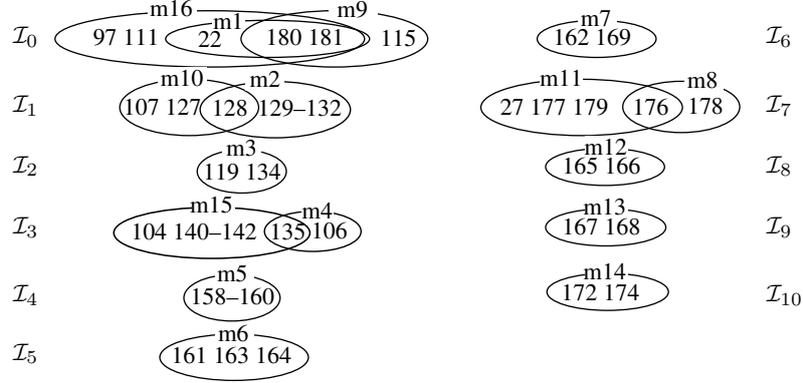


Figure 3. A partition of $\mathcal{I}=\{(m1), \dots, (m16)\}$ into 11 independent sub-theories.

We solved the SAT problem of each of the 11 sub-theories \mathcal{I}_i of \mathcal{I} , obtaining the respective structured subsets of \mathcal{U} . The conclusion is that, from all the 2^{40} subsets of \mathcal{U} , the number of subsets that are structured is $\prod_{i=0}^{10} |\{m|m \text{ is a model of } \mathcal{I}_i\}| = 28 \cdot 57 \cdot 3 \cdot 41 \cdot 5 \cdot 5 \cdot 3 \cdot 17 \cdot 3 \cdot 3 \cdot 3 = 6\,757\,902\,900$. This technique was also implemented in our tool.

5.4. Finding the maximally structured subsets of \mathcal{T}

In order to maximize precision w.r.t. \mathcal{T} , from all the structured subsets of \mathcal{U} , we focus on the maximally structured subsets, i.e., subsets of \mathcal{U} such that the inclusion of any other axiom from \mathcal{U} generates an unstructured subset of \mathcal{U} due to disobeying the regularity or simplicity structural restrictions. Therefore, from all the models of \mathcal{I} , we are interested in the ones that represent maximally structured subsets of \mathcal{U} .

In practice, the problem of finding a maximally structured subset of \mathcal{U} reduces to finding a model of \mathcal{I} that is maximal w.r.t. the truthness of the $\mathbb{I}(i)$ propositions about including the axiom (a_i) in the TBox—Let us call this property “maximally- \mathbb{I} .” In such models, the change of any truth-value assignment from *false* to *true* generates an interpretation that does not satisfy the meta-theory \mathcal{I} , i.e., which is no model of \mathcal{I} . Since the sub-theories \mathcal{I}_i are independent, the consecutive union of one maximally- \mathbb{I} model per \mathcal{I}_i is a maximally- \mathbb{I} model of \mathcal{I} , which represents a maximally structured subset of \mathcal{U} (our target).

However, a maximally structured subset of \mathcal{U} concerns only the axioms that are in \mathcal{U} , while the sub-theories \mathcal{T}_j of \mathcal{T} that we aim at building deal with the whole $\mathcal{T}=\{(a1), \dots, (a185)\}$. Since \mathcal{T} has more safe axioms than unsafe axioms, instead of informing which axioms from \mathcal{U} a \mathcal{T}_j includes, we present which axioms from \mathcal{U} a \mathcal{T}_j excludes. Also, since in Figure 4 (Section 6.2) we will refer to the sets of excluded axioms by their indexes, we use tuples instead of sets: \mathbb{U}_i is a tuple that orders sets of axioms from \mathcal{U} , where the user must pick exactly one set from \mathbb{U}_i to be excluded from \mathcal{T} when creating a *SR_{OTQ}* theory \mathcal{T}_j .

$$\mathbb{U}_0 = \langle \{22, 115\}, \{180, 181\} \rangle;$$

$$\mathbb{U}_1 = \langle \{107, 127, 129\}, \{107, 127, 130, 131, 132\}, \{128, 129\}, \{128, 132\} \rangle;$$

$$\mathbb{U}_2 = \langle \{119\}, \{134\} \rangle;$$

$$\mathbb{U}_3 = \langle \{104, 106, 141, 142\}, \{106, 140\}, \{135\} \rangle;$$

$$\mathbb{U}_4 = \langle \{158\}, \{159, 160\} \rangle;$$

$$\mathbb{U}_5 = \langle \{161, 164\}, \{163\} \rangle;$$

$$\mathbb{U}_6 = \langle \{162\}, \{169\} \rangle;$$

$$\begin{aligned} \mathbb{U}_7 &= \langle \{27, 177, 178\}, \{176, 177\}, \{176, 179\}, \{178, 179\} \rangle; \\ \mathbb{U}_8 &= \langle \{165\}, \{166\} \rangle; \\ \mathbb{U}_9 &= \langle \{167\}, \{168\} \rangle; \\ \mathbb{U}_{10} &= \langle \{172\}, \{174\} \rangle. \end{aligned}$$

Given the 6 757 902 900 structured subsets of \mathcal{U} , the number of subsets that are maximally structured is $2 \cdot 4 \cdot 2 \cdot 3 \cdot 2 \cdot 2 \cdot 4 \cdot 2 \cdot 2 \cdot 2 = 12\,288$, leading to 12 288 UFO-B *SROTQ* theories \mathcal{T}_j .

5.5. Discussion

Each maximally structured sub-theory \mathcal{T}_j can be defined by choosing exactly one set from each tuple \mathbb{U}_i . We will briefly discuss some consequences of these choices, which inevitably result in precision loss when compared to the reference ontology.

On \mathbb{U}_0 , by removing (a180) and (a181), one allows the existence of a *ProcessualRole* r that is inducedBy a *ParticipationUniversal* u and an *Object* o that is the participant of an *AtomicEvent* p that instantiates u , or that is partOf a *Participation* that instantiates u , and such that o does not plays the *ProcessualRole* r .

On \mathbb{U}_0 , by removing (a22) and (a115), one allows the existence of an *Object* that plays a *ProcessualRole* without instantiating it, and the existence of a *ProcessualRole* played by no *Object*.

On \mathbb{U}_1 , removing (a130) allows a *ComplexEvent* a to have an *Event* b as part without mereologically overlapping b . Similarly, removing (a131) allows an *Event* a to be part of a *ComplexEvent* b without mereologically overlapping b . Removing (a132) loses pretty much the definition of *mereologicallyOverlaps* as it would allow a *ComplexEvent* a to have as part an *Event* b that is part of a *ComplexEvent* c and such that a and c do not mereologicallyOverlaps.

On \mathbb{U}_2 , by removing (a134) one allows the existence of a *Situation* that activates a *Disposition* manifestedBy an *Event* but that does not triggers the *Event*.

On \mathbb{U}_3 , removing (a135) allows an *AtomicEvent* that manifests a *Disposition* that inheresIn an *Object* but such that the *AtomicEvent* does not dependsOn the *Object*. Removing (a141) allows a *Participation* to exclusivelyDependsOn less/more than one *Object*. Removing (a142) allows an *Event* that exclusivelyDependsOn exactly one *Object* to not be a *Participation*.

On \mathbb{U}_5 , by removing (a161) one allows the existence of an *Event* the endPoint of which precedes the beginPoint of another *Event*, but such that the first *Event* does not happen before the second *Event*.

On \mathbb{U}_6 , removing (a162) allows the existence of an *Event* the endPoint of which is the beginPoint of another *Event*, but such that the first *Event* does not meets the second *Event*. Removing (a169) allows an *Event* to meets itself, which should be forbidden since it is a theorem of UFO-B that *Events* must perdure, they must have their beginPoints different from their endPoints. Removing (a169) also allows two *Events* to meets each other, which is forbidden for the same reason just discussed.

On \mathbb{U}_7 , by removing (a176) one allows the existence of an *Event* that bringsAbout a *Situation* that triggers an *Event*, but such that the first *Event* does not directlyCauses the second *Event*.

Going back to the issue of (a55) being entailed by $\{(a27),(a56)\}$ and (a81) being entailed by $\{(a27),(a82)\}$ (Section 4.2), note that $\mathbb{I}(27)$ is in the first set in \mathbb{U}_7 , so the *SROTQ* theories built from this set do not have (a27). Similarly for (a92), which is entailed by $\{(a91),(a130)\}$ or $\{(a65),(a131)\}$, where both $\mathbb{I}(130)$ and $\mathbb{I}(131)$ are in the second set in \mathbb{U}_1 . Finally, (a90) is entailed by $\{(a88),(a140)\}$, where $\mathbb{I}(140)$ is in the second set in \mathbb{U}_3 .

On the other hand, (a89) is entailed by $\{(a88),(a143)\}$, but neither (a88) nor (a143) are excluded in any set of any \mathbb{U}_i . This means that we can safely add an optional tuple \mathbb{U}_{11} to remove (a89) from \mathcal{T} :

$$\mathbb{U}_{11} = \langle \{89\} \rangle.$$

Note that while $\approx 0.614628\%$ of the subsets of \mathcal{U} are structured, only $\approx 0.000001\%$ of them are maximally structured. Given that finding the structured subsets of \mathcal{U} is already a difficult task in terms of time complexity, is it worth to also solve the task of finding the maximally structured subsets?²⁰ In other words, how bad is it to produce sub-theories \mathcal{T}_j of \mathcal{T} that arbitrarily exclude axioms from UFO-B, losing precision without a reasonable computational justification? For instance, just by removing all the 40 unsafe axioms (see \mathcal{U}) from \mathcal{T} , one can be sure that the resultant sub-theory $\mathcal{T} \setminus \mathcal{U}$ conforms to regularity and simplicity (is a structured subset of \mathcal{T}), and so that reasoning on it is decidable. On the other hand, the maximally structured sub-theories remove only 14 to 23 unsafe axioms from \mathcal{T} .

Since UFO-B is infrequently updated, selecting the maximally structured *SRIOQ* formalizations is also an infrequent task, and we believe it is not too much of a burden to do it, aiming at sub-theories that capture as much as possible of the reference formalization of UFO-B.

Finally, a problem that is equivalent to the one we dealt with in this section is the problem of finding all the Maximal Satisfiable Subsets (MSSes) (Liffiton and Sakallah, 2008) of $\mathcal{U} \cup \mathcal{I}$ and then collecting only the ones that model \mathcal{I} . This is because all axioms in \mathcal{I} *must* be satisfied, while the satisfiability of \mathcal{U} should be maximized.

Concerning implementation, we built a (free and open source) Haskell stack package that, given a text file with an ASCII representation of \mathcal{T} (using a syntax that we defined), derives both the regularity and simplicity meta-axioms. Since the SAT problem for regularity is hard to solve computationally by means of the technique presented in Codish et al. (2006), but may be easy to solve by hand (as in our case), we also allow the user to provide pre-computed meta-axioms for regularity in a separate file.²¹ The tool returns the tuples \mathbb{U}_i together with some extra data, e.g., on the amount of propositions used and quantities of structured and maximally structured subsets of \mathcal{U} per \mathcal{I}_i . Regarding computational times, on a machine equipped with Ubuntu 18.10, Intel® Core™ i7-3520M CPU @ 2.90GHz \times 4, and 5.7GiB of RAM, and providing \mathcal{T} and (m1) as input, computing the tuples \mathbb{U}_i took 0.48s of CPU time. The source code is available at <https://osf.io/kmvna/>, as well as the files used/generated for this paper.

6. Validation

Preparing for expounding the validation in Section 6.2, let us first introduce the final part of the syntax of *SRIOQ* in Section 6.1.

6.1. Individuals in *SRIOQ*

Beyond the terminological knowledge, it is often also useful to predicate upon specific individuals (constants) from the universe of discourse. An example in FOL could be “Person(mary).” In *SRIOQ*,

²⁰We thank Frank Loebe (reviewer) for asking for an explanation on the trade-off of guaranteeing structural maximality.

²¹The user may also provide as input all the meta-axioms for regularity and simplicity (our $\{(m1), \dots, (m16)\}$), instead of \mathcal{T} .

this is done by means of an ABox. Let \mathbf{I} be a set of *individual names*, where the sets \mathbf{C} (concept names), \mathbf{R} (role names), and \mathbf{I} (individual names) are mutually disjoint.

Definition 10 (Individual assertion). *Assertions on individuals declare which concepts apply to them and how they relate to each other. Examples are “mary : Person,” “(mary, john) : motherOf,” “(john, mary) : ¬motherOf,” and “mary ≠ john.” More generally, an individual assertion is a formula like $\lceil a : C \rceil$, $\lceil (a, b) : R \rceil$, $\lceil (a, b) : \neg R \rceil$, or $\lceil a \neq b \rceil$, with $a, b \in \mathbf{I}$, R a role, and C a concept.*

Definition 11 (ABox). *An ABox is a finite set of individual assertions.*

ABox consistency for *SR_QIQ* is decidable (Horrocks et al., 2006).

6.2. Validation by means of ABoxes

In order to empirically evaluate the theories \mathcal{T}_j derived in the previous section, we codified the *SR_QIQ* axioms in OWL 2 DL and checked if they are consistent with instances generated from the reference codification of UFO-B in Alloy.

First, we built a full TBox in OWL 2 DL, which is possibly undecidable since the corresponding set of *SR_QIQ* axioms $\mathcal{T} = \{(a1), \dots, (a185)\}$ (presented in Section 4.2) violates both regularity and simplicity. Then, we manually annotated each OWL 2 DL axiom with a label so to have a bijection between the *SR_QIQ* axioms of \mathcal{T} and the axioms of the full TBox in OWL 2 DL. Thereafter, we used a script supported by the OWLAPI²² to automatically generate, from \mathcal{T} , an OWL 2 DL TBox _{j} for each one of the 12 288 *SR_QIQ* theories \mathcal{T}_j (as explained in Section 5.5).

For the generation of instances, we used the Alloy Analyzer tool version 5.0.0.1 with a new Alloy codification that exactly corresponds to the FOL axioms shown in Section 3.²³ These models are “fully-consistent,” in the sense that they have at least one instance for each class and relation. In practice, this means that we added to the Alloy codification a new axiom composed of several conjunctions of existential quantifications, one per class or relation of UFO-B. Using Alloy Analyzer, we generated 11 models of the FOL theory, with scope-size—i.e., the cardinality of the domain of quantification—from 25 to 35, where 25 is the minimum scope-size where the Alloy codification is satisfiable, and a maximum of 35 was chosen due to computational time constraints (generating such an instance took more than 24 hours).²⁴

Each Alloy model was parsed as a collection of individual assertions composing an ABox (using OWLAPI) and joined to each TBox _{j} to form an OWL 2 DL KB. In total, this process produced $12\,288 \times 11 = 135\,168$ KBs. We used the DL reasoner Pellet version 2.4.0 to test the consistency of these KBs, all of which are consistent. For each KB, 5 executions of the consistency test were performed,²⁵ totaling 675 840 executions. We present execution times in Figure 4, in which each $\mathbb{U}_{i,k}$ represents the sum of all the execution times of TBoxes that remove the k -th set of axioms of the tuple \mathbb{U}_i , for each one of the 11 ABoxes but taking the *average* of the 5 executions.

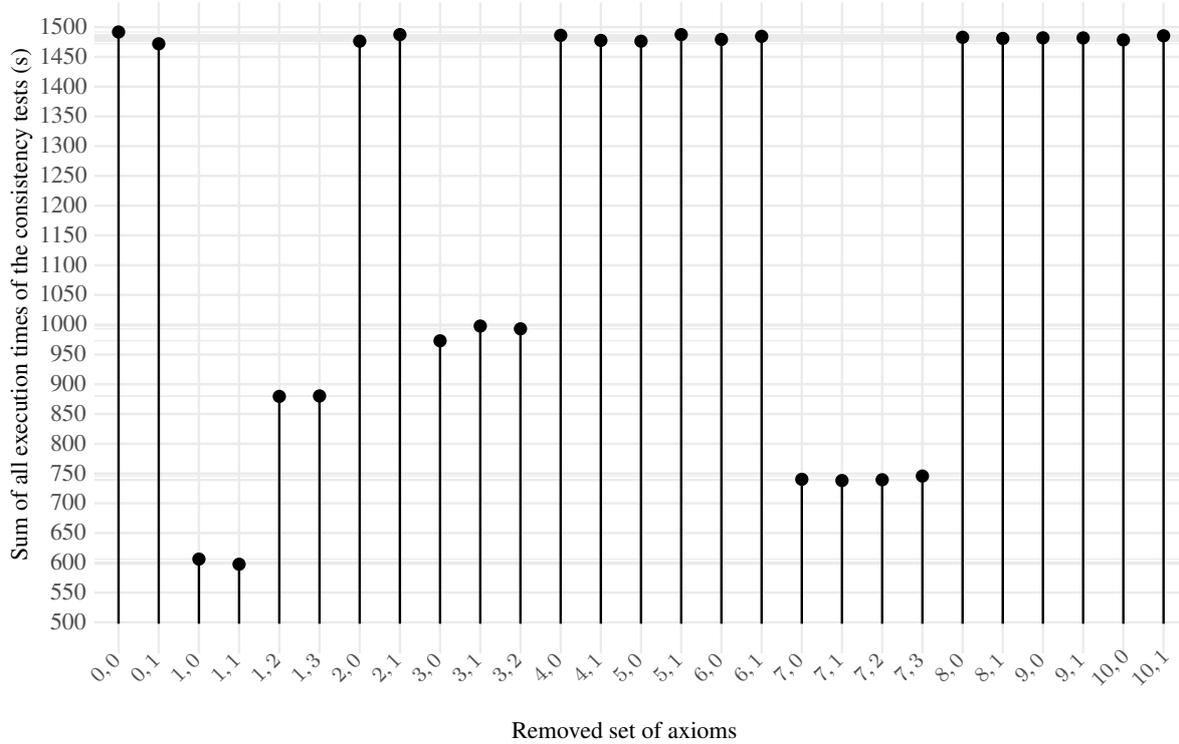
In Figure 4, one can note the difference between the values for $\mathbb{U}_{1,0}$ or $\mathbb{U}_{1,1}$ and the values for $\mathbb{U}_{1,2}$ or $\mathbb{U}_{1,3}$. What $\mathbb{U}_{1,0}$ and $\mathbb{U}_{1,1}$ have in common is the exclusion of $\{107, 127\}$. Interestingly, (a107) is the only

²²<http://owlcs.github.io/owlapi/>.

²³We are not using the original Alloy extended version of UFO-B (Guizzardi et al., 2013).

²⁴Note that, while being *models* in the technical sense, the models generated by the Alloy Analyzer tool concerning the specification of UFO-B in Alloy are different from the models found by the SAT solver for \mathcal{I} .

²⁵On a machine equipped with an Intel Core at 3.30GHz, Ubuntu 15.04, and Sun Java 1.8.

Figure 4. Execution times concerning the sets $\mathbb{U}_{i,k}$.

axiom in \mathcal{T} that states a “ ≥ 2 ” cardinality constraint and the cases in which (a107) is removed have the two shortest averages of execution times. On the other hand, “Asy” is removed in other cases, with worse average of execution times. Therefore, it seems that either “ ≥ 2 ” alone is an “expensive” operator, or the interplay between “ ≥ 2 ” and “Asy” is the culprit. Finally, for all the \mathbb{U}_i , the sum of the values of each possibility k must be the same (≈ 3000 s); this is not visually apparent in Figure 4 because the inferior threshold is 500s instead of zero.

7. Related work

We briefly present some ontologies of events expressible in (decidable) fragments of FOL (Section 7.1), and compare them with UFO-B (Section 7.2).

7.1. Event ontologies

The Event Ontology (EO) (Raimond and Abdallah, 2007a) is probably the lightest and most used ontology of events, supporting a classification based on space (using the WGS84 Positioning Ontology (Brickley, 2006)) and time regions (using the OWL-Time Ontology (Hobbs and Pan, 2017) and Timeline Ontology (Raimond and Abdallah, 2007b)). DOLCE+DnS Ultralite (DUL) is a lightweight foundational ontology, which is a combination and simplification of the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) (Masolo et al., 2003) and the Constructive Descriptions

and Situations (c.DnS) pattern for representing aspects of social reality. The Event-Model-F is a formal model of events based on DUL. It can express events and their mereological, causal and correlative relationships, also allowing the modeler to represent different points of view on the same event. The Linking Open Descriptions of Events (LODE) (Shaw et al., 2009) provides interoperability between Cyc, EO, Event-Model-F, ABC CIDOC CRM, EventsML-G2 and DUL by an analysis based on the “factual” aspects of events, the four Ws: *What, Where, When* and *Who*. The Simple Event Model (SEM) (van Hage et al., 2011) allows the modeler to avoid making domain-specific assumptions. It also allows the representation of view points in the description of an event. Finally, the Process Specification Language (PSL) (Grüninger, 2009) was designed to facilitate the exchange of process information among manufacturing systems. PSL is formalized in Common Logic (CL), but since there is a formalization in OWL (Katsumi and Grüninger, 2015), we include it in this discussion, basing our analysis on this formalization of PSL in OWL.

7.2. Comparison with UFO-B

In our comparison, we follow a strategy based on Shaw et al. (2009), focusing on the following aspects:

- *Agentivity*: Similarly to LODE, EO, DUL and PSL, UFO-B is agnostic concerning agentivity (or, more generally, to teleology). Differently, SEM allows for the representation of subjective points of view by means of `sem:View`. Event-Model-F has also an interpretation pattern that defines an `F:Interpretant`.
- *Time*: UFO-B follows the usual view of events as entities happening over a time interval, which is expressed by means of the relations `ufo-b:beginPoint` and `ufo-b:endPoint` to a class `ufo-b:TimePoint`. Similarly, SEM uses the roles `sem:hasBeginTimeStamp` and `sem:hasEndTimeStamp` with a literal. PSL uses the roles `psl:begins` and `psl:ends` to relate `psl:Activity_Occurrences`, `psl:Objects` and `psl:TimeIntervals` to `psl:TimePoints`. EO and LODE use `TemporalEntity` from OWL-TIME, which allows the representation of intervals, instants, and Allen’s relations between intervals. Note that, due to the syntactic constraints of *SR/OIQ* (which guarantee decidability), each of our different UFO-B formalizations allows the representation of some, but not all, aspects of Allen’s temporal relations. DUL has an object property `dul:hasEventDate` with a codomain `dul:TimeInterval`, this latter being the domain of `dul:hasIntervalDate`, a datatype property that encodes values from `xsd:date`. Note that `dul:TimeInterval` is also used in Event-Model-F to constrain events.
- *Space*: In UFO-B, “all spatial properties of events are defined in terms of the spatial properties of their participants” (Guizzardi et al., 2013, p.331). Spatial properties are then inherited from UFO-A, which formalizes the notions of `ufo-a:Quality`, `ufo-a:Quale`, `ufo-a:QualityUniversal`, `ufo-a:QualityDomain`, and `ufo-a:QualityDimension`. In PSL, both `psl:Activities` and `psl:Activity_Occurrences` (i.e., types and tokens) relate to `psl:Locations` by means of the roles `psl:occurs_at` and `psl:occurred_at`, respectively. DUL (followed by LODE) makes a distinction between a region with spatial boundaries (like `SpatialThing` from the W3C Basic Geo Vocabulary)²⁶ and meaningful places for which spatial boundaries are not mandatory. This distinction is represented by `dul:hasRegion` (and the generic datatype property `dul:hasDataValue`) for the former case (`lode:inSpace` for LODE) and `dul:hasLocation` for the latter case (`lode:atPlace` for LODE, `sem:hasPlace` for SEM). All the instances of

²⁶<https://www.w3.org/2003/01/geo/>.

`sem:Place` are symbolic places, nevertheless their location can be attached with various datatypes (e.g., W3C Geospatial Vocabulary²⁷ and W3C Basic Geo Vocabulary).

- *Participation*: UFO-B considers an event an `ufo-b:Participation` iff it exclusively depends on a single (possibly inanimate) object; this event is an `ufo-b:participationOf` of the object. In PSL, a `psl:Object` `psl:participates` in a `psl:Activity_Ocurrence`. The relation `psl:participates` is specialized into `psl:performed_in`, which holds between `psl:Actors` and `psl:Activity_Ocurrences`. A `psl:Actor` `psl:performs` a `psl:Activity`. Concerning SEM, the codomain `sem:Actor` of the object property `sem:hasActor` is a class of individuals that are not “necessarily sentient, so potentially objects i.e. a thing, animate or inanimate, physical or non-physical.” So, `ufo-b:participationOf` seems equivalent to `sem:hasActor`, `lode:involved`, `dul:hasParticipant`, and `eo:factor`. UFO-B lacks a more specific participation relation between agents and events, such as `lode:involvedAgent`, `dul:involvesAgent`, and `eo:agent`.
- *Causality*: UFO-B captures causality between events by means of `ufo-b:directlyCauses` and `ufo-b:causes` (a sort of transitive closure of `ufo-b:directlyCauses`). Moreover, UFO-B, DUL, Event-Model-F and EO share the view that events and situations are correlated; e.g., for UFO-B, `ufo-b:Situations` `ufo-b:triggers` `ufo-b:Events`, and `ufo-b:Events` `ufo-b:bringsAbout` `ufo-b:Situations`; while for DUL, `dul:Event` `dul:hasPrecondition` (`dul:hasPostcondition`) `dul:Situation`. EO can represent such a link by means of the object property `eo:factor`. Neither LODE nor SEM deal with causality. The fragment of PSL formalized in OWL (Katsumi and Grüninger, 2015) also does not capture causality.
- *Mereology*: UFO-B formalizes an Extensional Mereology (EM) (Casati and Varzi, 1999, p. 40) between events (partitioning events between `ufo-b:AtomicEvents` and `ufo-b:ComplexEvents`), but the *SRIOIQ* translations can only capture some aspects of the `ufo-b:hasPart` relation. Interestingly, DUL distinguishes between mereological and temporal parthood; `dul:hasPart` corresponds to `ufo-b:during`, and `dul:hasConstituent` to `ufo-b:hasPart`. SEM and EO consider a simple theory of parthood on events by means of `sem:hasSubEvent` (assuming no causality) and `eo:sub_event`. Event-Model-F is more concerned with an event composition relation than event parthood. This theory relates an `F:Event` that is `F:Composite` to all its `F:Event` `F:Components` by means of a single `F:EventCompositionSituation` relationship. An `F:EventCompositionSituation` satisfies an `F:CompositionDescription` that defines the structure that “glues” the `F:Components` in an `F:Composite` `F:Event`. PSL formalizes parthood (`psl:p`) and proper parthood (`psl:pp`), making no restrictions on their (co)domain. LODE does not deal with parthood relations among events.
- *Manifestation*: UFO-B explains the nature of atomic events in the following way: an `ufo-b:Situation` `ufo-b:activates` an `ufo-b:Disposition` (which `ufo-b:inheritsIn` an `ufo-b:Object`) which is `ufo-b:manifestedBy` an `ufo-b:AtomicEvent`. As far as we know, none of the previously discussed theories provides an explanation for the occurrence of an event. The closest relation to `ufo-b:manifestedBy` would be an inverse of `eo:product` (“Everything produced by an event”).

Note that UFO-B does not have a concept of datatype properties because it deals with such entities by means of Qualities, Qualess, QualityUniversals and QualityStructures.

²⁷<https://www.w3.org/2005/Incubator/geo/XGR-geo/>.

Table 1 presents the expressivities, the numbers of logical axioms, classes, object properties (OP) and data properties (DP)²⁸ of some foundational event-based ontologies that have been released over the past years.

	<i>Ontology</i>	Expressivity	Axioms	Classes	OP	DP
1	CIDOC CRM	<i>SRI^D</i>	434	81	130	7
2	DUL	<i>SHIN^D</i>	600	76	107	5
3	EM-F	<i>ALCHI_Q</i>	115	38	23	0
4	EO	<i>SROIN^D</i>	300	31	50	30
5	LODE	<i>ALHF</i>	31	13	15	0
6	PSL	<i>SRI_Q</i>	410	27	42	0
7	SEM	<i>ALHI^D</i>	53	17	17	7
8	UFO-B	<i>SRI_Q</i>	186	26	34	0

Table 1
Metrics about some event ontologies.

Note that the OWL 2 DL version of UFO-B has 186 axioms, one more than the 185 axioms of the *SR_{OIQ}* version. This is because we had to reify `hasPart-`. This is also the reason why the OWL 2 DL version of UFO-B has 34 object properties, while the *SR_{OIQ}* version has 33 role names.

The Uniform Resource Locators (URLs) of these ontologies are:

- (1) <http://purl.org/NET/cidoc-crm/core>
- (2) <http://ontologydesignpatterns.org/ont/dul/DUL.owl>
- (3) <http://events.semantic-multimedia.org/ontology/2008/12/15/model.owl>
- (4) <http://motools.sf.net/event/event.n3>
- (5) <http://linkedevents.org/ontology/2010-10-07/rdfxml/>
- (6) http://stl.mie.utoronto.ca/ontologies/process_specification_language/psl_1oc_actors.owl
- (7) <http://semanticweb.cs.vu.nl/2009/11/sem/sem.rdf>
- (8) <http://www.nemo.inf.ufes.br/ufo-b/>

8. Conclusions

The development of suitable foundational theories is an important step towards the definition of precise semantics and sound methodological principles for modeling complex real-world phenomena. From an engineering point of view, there is also a need for decidable representations of these theories that can support automated reasoning. We address the combination of ontological adequacy and decidability by providing OWL 2 DL TBoxes that (partially) represent the axiomatization of UFO-B.

The difficulty in dealing with the trade-off between expressivity and time complexity when writing a theory in *SR_{OIQ}* lies not only in doing tricky combinations with the constructs of *SR_{OIQ}* to

²⁸As shown by the Protégé tool, available at: <https://protege.stanford.edu/>. Note that Protégé claims that the unstructured set \mathcal{T} is expressible in *SRI_Q*, while, as discussed in Section 4.1, the presentation of *SRI_Q* in Horrocks et al. (2005) does not provide the construct for asymmetry “Asy,” which we use, and that cannot be enforced by means of RIAs (Horrocks et al., 2006, p. 59).

write axioms that correspond to the axioms of the original theory—but also in managing an exponential amount of possibilities concerning incompatibilities between axioms. To deal with this issue, we designed a methodology for the translation from a FOL theory to *SR_{OLQ}*. The large number of codification alternatives of a single reference theory in FOL provides us evidence of the difficulty that a modeler faces when using DLs as the single formalism to specify a reference ontology such as UFO-B. Because of that, we defend an approach which clearly separates reference ontology specification (in FOL) from lightweight ontology codification (in *SR_{OLQ}*), while ensuring the systematic derivation of the lightweight ontology. The approach guarantees that precision is only sacrificed when required for particular computational goals, in our case, those goals for which *SR_{OLQ}* was designed in the first place. Given the number of possible combinations of axioms to be included in the lightweight ontology, maximizing precision while obtaining desirable computational properties would be too hard to perform manually.

As future work, we intend to investigate alternative DL mappings for the UFO theories presented here, dealing with different aspects of the trade-off between expressivity and tractability. We plan to investigate both mappings to more expressive languages (e.g., including the use of Semantic Web Rule Language (SWRL) for modeling aspects such as the temporal relations between events that could not be captured here) as well as to less expressive languages (e.g., OWL 2 EL) that can support efficient reasoning over large event data sets. Another important question is whether the axioms that were not included in the *SR_{OLQ}* translation are really not expressible in *SR_{OLQ}* and, in that case, whether *SR_{OLQ}* can be extended with additional features such as role negation (Schmidt and Tishkovsky, 2007) or nominal schemata (Krötzsch et al., 2011) without affecting decidability of the reasoning services.

Another interesting future work we intend to investigate is the formalization of both the FOL and *SR_{OLQ}* versions of UFO-B in The Distributed Ontology, Model and Specification Language (DOL) (Object Management Group, 2018) to upload them in Ontohub²⁹ and check their alignment, as any theorem of any *SR_{OLQ}* version of UFO-B should correspond to a theorem of the FOL version.

The interested reader can find our input files, the SAT solutions, the Haskell source code, our Alloy codification of UFO-B, the generated Alloy models, the OWL 2 DL TBoxes, and the KBs at <https://osf.io/kmvna/>.

Acknowledgements

We thank Frank Loebe and the anonymous reviewer for their thorough and constructive comments, which greatly contributed to improving the final version of this article. We also thank Carlos Mencía for his help in validating the structural maximality of our UFO-B *SR_{OLQ}* theories.

On funding, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. In particular, the first author is currently supported by the postdoctoral grant with process № 83740910/18 regarding the “edital FAPES/CAPES № 10/2018 - Bolsa de Fixação de Doutores (PROFIX).” Moreover, the first two authors were respectively supported by the postdoctoral grants with process № 71024352 and process № 71047522 regarding the “edital FAPES/CAPES № 009/2014 - Bolsa de Fixação de Doutores.” The third author was partly supported by the OCEAN Project (Free University of Bozen-Bolzano) and by the CNPq grant #312158/2015-7. The fifth author was partly supported by the CNPq grants #407235/2017-5 and #312123/2017-5, CAPES grant #23038.028816/2016-41 and FAPES grant #69382549.

²⁹<https://ontohub.org>.

References

- Allen, J.F. (1983). Maintaining Knowledge about Temporal Intervals. *Commun. ACM*, 26(11), 832–843. doi:10.1145/182.358434.
- Armstrong, D.M. (1989). *Universals: An Opinionated Introduction*. Focus Series. Westview Press. <https://books.google.com.br/books?id=MJzmAAAAIAAJ>.
- Armstrong, D.M. (2004). *Truth and Truthmakers*. Cambridge University Press.
- Azevedo, C.L.B., Iacob, M., Almeida, J.P.A., van Sinderen, M., Pires, L.F. & Guizzardi, G. (2015). Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate. *Inf. Syst.*, 54, 235–262. doi:10.1016/j.is.2015.04.008.
- Batsakis, S., Petrakis, E.G.M., Tachmazidis, I. & Antoniou, G. (2017). Temporal representation and reasoning in OWL 2. *Semantic Web*, 8(6), 981–1000. doi:10.3233/SW-160248.
- Benevides, A.B., Bourguet, J., Guizzardi, G. & Peñalosa, R. (2017). Representing the UFO-B Foundational Ontology of Events in SROIQ. In *Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, September 21-23, 2017*. http://ceur-ws.org/Vol-2050/FOUST_paper_7.pdf.
- Biere, A., Heule, M., van Maaren, H. & Walsh, T. (Eds.) (2009). *Handbook of Satisfiability*. Frontiers in Artificial Intelligence and Applications (Vol. 185). IOS Press.
- Bourguet, J., Guizzardi, G., Benevides, A.B. & Zamborlini, V. (2017). Empirically Evaluating Three Proposals for Representing Changes in OWL2. In *Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, September 21-23, 2017*. http://ceur-ws.org/Vol-2050/DEW_paper_4.pdf.
- Brickley, D. (2006). W3C Semantic Web Interest Group: Basic Geo (WGS84 lat/long) Vocabulary. Online. Last accessed at 1st July 2019. <https://www.w3.org/2003/01/geo/>.
- Carolo, F. & Burlamaqui, L. (2011). Improving web content management with semantic technologies. In *Semantic Technology Conference (SemTech), San Francisco*.
- Casati, R. & Varzi, A. (2015). Events. In E.N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2015 ed.). Metaphysics Research Lab, Stanford University.
- Casati, R. & Varzi, A.C. (1999). *Parts and places: The structures of spatial representation*. MIT Press.
- Codish, M., Lagoon, V. & Stuckey, P.J. (2006). Solving Partial Order Constraints for LPO Termination. In F. Pfenning (Ed.), *Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Proceedings*. Lecture Notes in Computer Science (Vol. 4098, pp. 4–18). Springer. doi:10.1007/11805618_2.
- Daly, C. (1994). Tropes. In *Proceedings of the Aristotelian Society*. New Series (Vol. 94, pp. 253–261). Wiley. JSTOR. <http://www.jstor.org/stable/4545197>.
- de Souza, É.F., de Almeida Falbo, R. & Vijaykumar, N.L. (2017). ROoST: Reference Ontology on Software Testing. *Applied Ontology*, 12(1), 59–90. doi:10.3233/AO-170177.
- Duarte, B.B., Souza, V.E.S., de Castro Leal, A.L., de Almeida Falbo, R., Guizzardi, G. & Guizzardi, R.S.S. (2016). Towards an Ontology of Requirements at Runtime. In R. Ferrario and W. Kuhn (Eds.), *Formal Ontology in Information Systems - Proceedings of the 9th International Conference, FOIS 2016, Annecy, France, July 6-9, 2016*. Frontiers in Artificial Intelligence and Applications (Vol. 283, pp. 255–268). IOS Press. doi:10.3233/978-1-61499-660-6-255.
- Grüninger, M. (2009). Using the PSL Ontology. In S. Staab and R. Studer (Eds.), *Handbook on Ontologies*. International Handbooks on Information Systems (pp. 423–443). Springer. doi:10.1007/978-3-540-92673-3_19.
- Guizzardi, G. (2005). Ontological foundations for structural conceptual models. PhD Thesis, University of Twente, Enschede, The Netherlands, Enschede.
- Guizzardi, G. (2006). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In O. Vasilecas, J. Eder and A. Caplinskis (Eds.), *Databases and Information Systems IV - Selected Papers from the Seventh International Baltic Conference, DB&IS 2006, July 3-6, 2006, Vilnius, Lithuania*. Frontiers in Artificial Intelligence and Applications (Vol. 155, pp. 18–39). IOS Press. <http://www.booksonline.iospress.nl/Content/View.aspx?pid=5421>.
- Guizzardi, G. & Wagner, G. (2004). A Unified Foundational Ontology and some Applications of it in Business Modeling. In J. Grundspenkis and M. Kirikova (Eds.), *CAiSE'04 Workshops in connection with The 16th Conference on Advanced Information Systems Engineering, Riga, Latvia, 7-11 June, 2004, Knowledge and Model Driven Information Systems Engineering for Networked Organisations, Proceedings, Vol. 3* (pp. 129–143). Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia.
- Guizzardi, G., de Almeida Falbo, R. & Guizzardi, R.S.S. (2008). Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In M. Lencastre, J.F. e Cunha and A. Valecillo (Eds.), *Memórias de la XI Conferencia Iberoamericana de Software Engineering (CIBSE 2008), Recife, Pernambuco, Brasil, February 13-17, 2008* (pp. 127–140).
- Guizzardi, G., Wagner, G., de Almeida Falbo, R., Guizzardi, R.S.S. & Almeida, J.P.A. (2013). Towards Ontological Foundations for the Conceptual Modeling of Events. In W. Ng, V.C. Storey and J. Trujillo (Eds.), *Conceptual Modeling - 32th International Conference, ER 2013, Hong-Kong, China, November 11-13, 2013. Proceedings*. Lecture Notes in Computer Science (Vol. 8217, pp. 327–341). Springer. doi:10.1007/978-3-642-41924-9_27.

- Guizzardi, R.S.S., Li, F., Borgida, A., Guizzardi, G., Horkoff, J. & Mylopoulos, J. (2014). An Ontological Interpretation of Non-Functional Requirements. In P. Garbacz and O. Kutz (Eds.), *Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS 2014, September, 22-25, 2014, Rio de Janeiro, Brazil*. Frontiers in Artificial Intelligence and Applications (Vol. 267, pp. 344–357). IOS Press. doi:10.3233/978-1-61499-438-1-344.
- Hobbs, J.R. & Pan, F. (2017). Time ontology in OWL. Technical report OGC 16-071r2, W3C/OGC. Last accessed at 1st. July 2019. <https://www.w3.org/TR/owl-time/>.
- Horrocks, I., Kutz, O. & Sattler, U. (2005). The Irresistible *SR_{IQ}*. In B.C. Grau, I. Horrocks, B. Parsia and P.F. Patel-Schneider (Eds.), *Proceedings of the OWLED*05 Workshop on OWL: Experiences and Directions, Galway, Ireland, November 11-12, 2005*. CEUR Workshop Proceedings. CEUR-WS.org. <http://ceur-ws.org/Vol-188/sub20.pdf>.
- Horrocks, I., Kutz, O. & Sattler, U. (2006). The Even More Irresistible *SR_{OTQ}*. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006* (pp. 57–67). <http://www.aaai.org/Library/KR/2006/kr06-009.php>.
- Jackson, D. (2012). *Software Abstractions: logic, language, and analysis*. MIT press.
- Jr., P.S.S., Almeida, J.P.A. & Guizzardi, G. (2010). An ontology-based semantic foundation for ARIS EPCs. In S.Y. Shin, S. Ossowski, M. Schumacher, M.J. Palakal and C. Hung (Eds.), *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010* (pp. 124–130). ACM. doi:10.1145/1774088.1774114.
- Katsumi, M. & Grüninger, M. (2015). Using PSL to Extend and Evaluate Event Ontologies. In N. Bassiliades, G. Gottlob, F. Sadri, A. Paschke and D. Roman (Eds.), *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*. Lecture Notes in Computer Science (Vol. 9202, pp. 225–240). Springer. doi:10.1007/978-3-319-21542-6_15.
- Krötzsch, M., Simancik, F. & Horrocks, I. (2012). A Description Logic Primer. *CoRR*, abs/1201.4089. <http://arxiv.org/abs/1201.4089v3>.
- Krötzsch, M., Maier, F., Krisnadhi, A. & Hitzler, P. (2011). A better uncle for OWL: nominal schemas for integrating rules and ontologies. In S. Srinivasan, K. Ramamritham, A. Kumar, M.P. Ravindra, E. Bertino and R. Kumar (Eds.), *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011* (pp. 645–654). ACM. doi:10.1145/1963405.1963496.
- Liffiton, M.H. & Sakallah, K.A. (2008). Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints. *J. Autom. Reasoning*, 40(1), 1–33. doi:10.1007/s10817-007-9084-z.
- Loebe, F. (2003). An analysis of roles: Towards ontology-based modelling. Master's thesis, Computer Science Institute, Department of Mathematics and Computer Science, University of Leipzig, Germany. Onto-Med Report No. 6. <http://www.onto-med.de/publications/2003/loebe-f-2003-a.pdf>.
- Loebe, F. (2007). Abstract vs. social roles - Towards a general theoretical account of roles. *Applied Ontology*, 2(2), 127–158. <http://content.iospress.com/articles/applied-ontology/ao031>.
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N. & Oltramari, A. (2003). WonderWeb Deliverable D18. resreport Del 18, ISTC-CNR. Last accessed at 1st. July 2019. <http://wonderweb.man.ac.uk/deliverables/documents/D18.pdf>.
- Nardi, J.C., de Almeida Falbo, R. & Almeida, J.P.A. (2014). An Ontological Analysis of Service Modeling at ArchiMate's Business Layer. In M. Reichert, S. Rinderle-Ma and G. Grossmann (Eds.), *18th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2014, Ulm, Germany, September 1-5, 2014* (pp. 92–100). IEEE Computer Society. doi:10.1109/EDOC.2014.22.
- Object Management Group (2018). The Distributed Ontology, Modeling, and Specification Language. Technical report formal/18-09-02, OMG Headquarters, 109 Highland Avenue, Needham, MA 02494, USA, Object Management Group. Last accessed at 1st. July 2019. <https://www.omg.org/spec/DOL/1.0/PDF>.
- Pena, R.A.P. (2012). Suporte semântico à publicação de conteúdo jornalístico na Web. mthesis, Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio., Rio de Janeiro. doi:<https://doi.org/10.17771/PUCRio.acad.20235>.
- Raimond, Y. & Abdallah, S. (2007a). The event ontology. Online. Last accessed at 1st. July 2019. <http://motools.sf.net/event/event.html>.
- Raimond, Y. & Abdallah, S. (2007b). The timeline ontology. Online. Last accessed at 1st. July 2019. <http://motools.sf.net/timeline/timeline.html>.
- Schmidt, R.A. & Tishkovsky, D. (2007). Deciding ALBO with Tableau. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, A. Turhan and S. Tessaris (Eds.), *Proceedings of the 2007 International Workshop on Description Logics (DL2007), Brixen-Bressanone, near Bozen-Bolzano, Italy, 8-10 June, 2007*. CEUR Workshop Proceedings. CEUR-WS.org. http://ceur-ws.org/Vol-250/paper_75.pdf.
- Shaw, R., Troncy, R. & Hardman, L. (2009). LOD: Linking Open Descriptions of Events. In A. Gómez-Pérez, Y. Yu and Y. Ding (Eds.), *The Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings*. Lecture Notes in Computer Science (Vol. 5926, pp. 153–167). Springer. doi:10.1007/978-3-642-10871-6_11.
- van Hage, W.R., Malaisé, V., Segers, R., Hollink, L. & Schreiber, G. (2011). Design and use of the Simple Event Model (SEM). *J. Web Semant.*, 9(2), 128–136. doi:10.1016/j.websem.2011.03.003.

- W3C. (2012). *OWL 2 Web Ontology Language Document Overview* (Second edition ed.). Last accessed at 1st. July 2019. <https://www.w3.org/TR/owl2-overview/>.
- Werlang, R. (2015). Ontology-based approach for standard formats integration in reservoir modeling. mathesis, Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação., Porto Alegre. <http://hdl.handle.net/10183/115196>.
- Zamborlini, V. & Guizzardi, G. (2010). On the Representation of Temporally Changing Information in OWL. In *Workshops Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference, EDOCW 2010, Vitória, Brazil, 25-29 October 2010* (pp. 283–292). IEEE Computer Society. doi:10.1109/EDOCW.2010.50.

Appendix A. Regularity constraints

Providing further details in addition to Section 5, the constraints imposed by regularity are:

- (1) $\mathbb{I}(22) \rightarrow (\text{plays} \prec \text{instantiates}) \wedge (\text{plays}^- \prec \text{instantiates})$
- (2) $\mathbb{I}(27) \rightarrow (\text{directlyCauses} \prec \text{causes}) \wedge (\text{directlyCauses}^- \prec \text{causes})$
- (3) $\mathbb{I}(130) \rightarrow (\text{hasPart} \prec \text{mereologicallyOverlaps}) \wedge$
 $(\text{hasPart}^- \prec \text{mereologicallyOverlaps})$
- (4) $\mathbb{I}(131) \rightarrow (\text{hasPart} \prec \text{mereologicallyOverlaps}) \wedge$
 $(\text{hasPart}^- \prec \text{mereologicallyOverlaps})$
- (5) $\mathbb{I}(132) \rightarrow (\text{hasPart} \prec \text{mereologicallyOverlaps}) \wedge$
 $(\text{hasPart}^- \prec \text{mereologicallyOverlaps})$
- (6) $\mathbb{I}(134) \rightarrow (\text{activates} \prec \text{triggers}) \wedge (\text{activates}^- \prec \text{triggers})$
- (7) $\mathbb{I}(134) \rightarrow (\text{manifestedBy} \prec \text{triggers}) \wedge (\text{manifestedBy}^- \prec \text{triggers})$
- (8) $\mathbb{I}(135) \rightarrow (\text{manifestedBy} \prec \text{dependsOn}) \wedge (\text{manifestedBy}^- \prec \text{dependsOn})$
- (9) $\mathbb{I}(135) \rightarrow (\text{inheresIn} \prec \text{dependsOn}) \wedge (\text{inheresIn}^- \prec \text{dependsOn})$
- (10) $\mathbb{I}(137) \rightarrow (\text{manifestedBy} \prec \text{presentIn}) \wedge (\text{manifestedBy}^- \prec \text{presentIn})$
- (11) $\mathbb{I}(137) \rightarrow (\text{triggers} \prec \text{presentIn}) \wedge (\text{triggers}^- \prec \text{presentIn})$
- (12) $\mathbb{I}(138) \rightarrow (\text{manifestedBy} \prec \text{presentIn}) \wedge (\text{manifestedBy}^- \prec \text{presentIn})$
- (13) $\mathbb{I}(138) \rightarrow (\text{bringsAbout} \prec \text{presentIn}) \wedge (\text{bringsAbout}^- \prec \text{presentIn})$
- (14) $\mathbb{I}(139) \rightarrow (\text{inheresIn} \prec \text{presentIn}) \wedge (\text{inheresIn}^- \prec \text{presentIn})$
- (15) $\mathbb{I}(140) \rightarrow (\text{dependsOn} \prec \text{exclusivelyDependsOn}) \wedge$
 $(\text{dependsOn}^- \prec \text{exclusivelyDependsOn})$
- (16) $\mathbb{I}(143) \rightarrow (\text{participationOf} \prec \text{exclusivelyDependsOn}) \wedge$
 $(\text{participationOf}^- \prec \text{exclusivelyDependsOn})$
- (17) $\mathbb{I}(160) \rightarrow (\text{beginPoint} \prec \text{precedes}) \wedge (\text{beginPoint}^- \prec \text{precedes})$
- (18) $\mathbb{I}(160) \rightarrow (\text{endPoint} \prec \text{precedes}) \wedge (\text{endPoint}^- \prec \text{precedes})$
- (19) $\mathbb{I}(161) \rightarrow (\text{endPoint} \prec \text{before}) \wedge (\text{endPoint}^- \prec \text{before})$
- (20) $\mathbb{I}(161) \rightarrow (\text{precedes} \prec \text{before}) \wedge (\text{precedes}^- \prec \text{before})$
- (21) $\mathbb{I}(161) \rightarrow (\text{beginPoint} \prec \text{before}) \wedge (\text{beginPoint}^- \prec \text{before})$
- (22) $\mathbb{I}(162) \rightarrow (\text{endPoint} \prec \text{meets}) \wedge (\text{endPoint}^- \prec \text{meets})$
- (23) $\mathbb{I}(162) \rightarrow (\text{beginPoint} \prec \text{meets}) \wedge (\text{beginPoint}^- \prec \text{meets})$
- (24) $\mathbb{I}(176) \rightarrow (\text{bringsAbout} \prec \text{directlyCauses}) \wedge (\text{bringsAbout}^- \prec \text{directlyCauses})$
- (25) $\mathbb{I}(176) \rightarrow (\text{triggers} \prec \text{directlyCauses}) \wedge (\text{triggers}^- \prec \text{directlyCauses})$
- (26) $\mathbb{I}(180) \rightarrow (\text{dependsOn} \prec \text{plays}) \wedge (\text{dependsOn}^- \prec \text{plays})$
- (27) $\mathbb{I}(180) \rightarrow (\text{instantiates} \prec \text{plays}) \wedge (\text{instantiates}^- \prec \text{plays})$
- (28) $\mathbb{I}(180) \rightarrow (\text{inducedBy} \prec \text{plays}) \wedge (\text{inducedBy}^- \prec \text{plays})$
- (29) $\mathbb{I}(181) \rightarrow (\text{dependsOn} \prec \text{plays}) \wedge (\text{dependsOn}^- \prec \text{plays})$
- (30) $\mathbb{I}(181) \rightarrow (\text{hasPart} \prec \text{plays}) \wedge (\text{hasPart}^- \prec \text{plays})$
- (31) $\mathbb{I}(181) \rightarrow (\text{instantiates} \prec \text{plays}) \wedge (\text{instantiates}^- \prec \text{plays})$
- (32) $\mathbb{I}(181) \rightarrow (\text{inducedBy} \prec \text{plays}) \wedge (\text{inducedBy}^- \prec \text{plays})$
- (33) $\mathbb{I}(182) \rightarrow (\text{createdBy} \prec \text{presentIn}) \wedge (\text{createdBy}^- \prec \text{presentIn})$
- (34) $\mathbb{I}(182) \rightarrow (\text{bringsAbout} \prec \text{presentIn}) \wedge (\text{bringsAbout}^- \prec \text{presentIn})$
- (35) $\mathbb{I}(183) \rightarrow (\text{terminatedBy} \prec \text{presentIn}) \wedge (\text{terminatedBy}^- \prec \text{presentIn})$

- (36) $\mathbb{I}(183) \rightarrow (\text{triggers} \prec \text{presentIn}) \wedge (\text{triggers}^- \prec \text{presentIn})$
- (37) $\mathbb{I}(184) \rightarrow (\text{changedBy} \prec \text{presentIn}) \wedge (\text{changedBy}^- \prec \text{presentIn})$
- (38) $\mathbb{I}(184) \rightarrow (\text{bringsAbout} \prec \text{presentIn}) \wedge (\text{bringsAbout}^- \prec \text{presentIn})$
- (39) $\mathbb{I}(185) \rightarrow (\text{changedBy} \prec \text{presentIn}) \wedge (\text{changedBy}^- \prec \text{presentIn})$
- (40) $\mathbb{I}(185) \rightarrow (\text{triggers} \prec \text{presentIn}) \wedge (\text{triggers}^- \prec \text{presentIn})$

Note that, from the constraint (1), assuming $\mathbb{I}(22)$ entails “plays \prec instantiates.” Moreover, from condition (27), assuming $\mathbb{I}(180)$ entails “instantiates \prec plays.” Similarly, from case (31), assuming $\mathbb{I}(181)$ entails “instantiates \prec plays.” Therefore, (a22) is incompatible with both (a180) and (a181). These incompatibilities are captured by the meta axiom (m1) in Section 5.2.1.

The combination of Figure 5a with Figure 5b shows a strict partial order of roles satisfying the regularity syntactic constraint for $\mathcal{T} \setminus \{(a180), (a181)\}$, while Figure 5a together with Figure 5c shows a strict partial order of roles satisfying the regularity syntactic constraint for $\mathcal{T} \setminus \{(a22)\}$. Therefore, (m1) seems to be the only incompatibility due to regularity.

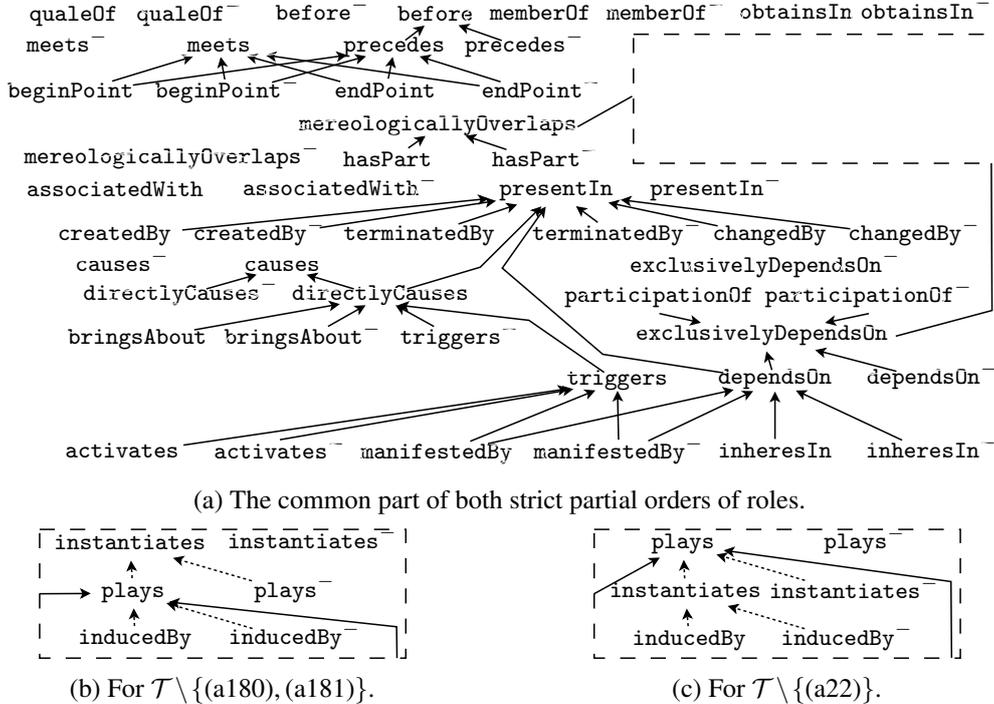


Figure 5. Two strict partial orders of roles satisfying regularity for subsets of \mathcal{T} .

Appendix B. Simplicity constraints

First step. We capture the (direct and indirect) sources of non-simplicity:

1. A role being non-simple due to subsuming a role composition (direct source of non-simplicity):

- $\mathbb{I}(132) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$
- $\mathbb{I}(134) \rightarrow \bar{\mathbb{S}}(\text{triggers})$
- $\mathbb{I}(135) \rightarrow \bar{\mathbb{S}}(\text{dependsOn})$
- $\mathbb{I}(160) \rightarrow \bar{\mathbb{S}}(\text{precedes})$
- $\mathbb{I}(161) \rightarrow \bar{\mathbb{S}}(\text{before})$
- $\mathbb{I}(162) \rightarrow \bar{\mathbb{S}}(\text{meets})$
- $\mathbb{I}(176) \rightarrow \bar{\mathbb{S}}(\text{directlyCauses})$
- $(\mathbb{I}(137) \vee \mathbb{I}(138) \vee \mathbb{I}(139) \vee \mathbb{I}(182) \vee \mathbb{I}(183) \vee \mathbb{I}(184) \vee \mathbb{I}(185)) \rightarrow \bar{\mathbb{S}}(\text{presentIn})$
- $(\mathbb{I}(180) \vee \mathbb{I}(181)) \rightarrow \bar{\mathbb{S}}(\text{plays})$

2. A role R being non-simple due to the role assertion $\lceil \text{Tra}(R) \rceil$ (direct source of non-simplicity):

- $\mathbb{I}(128) \rightarrow \bar{\mathbb{S}}(\text{hasPart});$
- $\mathbb{I}(159) \rightarrow \bar{\mathbb{S}}(\text{precedes});$
- $\mathbb{I}(164) \rightarrow \bar{\mathbb{S}}(\text{before});$
- $\mathbb{I}(166) \rightarrow \bar{\mathbb{S}}(\text{starts});$
- $\mathbb{I}(168) \rightarrow \bar{\mathbb{S}}(\text{during});$
- $\mathbb{I}(174) \rightarrow \bar{\mathbb{S}}(\text{equals});$
- $\mathbb{I}(177) \rightarrow \bar{\mathbb{S}}(\text{causes});$

3. A role being non-simple due to subsuming non-simple roles (indirect source of non-simplicity):

- $(\bar{\mathbb{S}}(\text{plays}) \wedge \mathbb{I}(22)) \rightarrow \bar{\mathbb{S}}(\text{instantiates})$
- $(\bar{\mathbb{S}}(\text{directlyCauses}) \wedge \mathbb{I}(27)) \rightarrow \bar{\mathbb{S}}(\text{causes})$
- $(\bar{\mathbb{S}}(\text{hasPart}) \wedge \mathbb{I}(130)) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$
- $(\bar{\mathbb{S}}(\text{hasPart}) \wedge \mathbb{I}(131)) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$
- $(\bar{\mathbb{S}}(\text{dependsOn}) \wedge \mathbb{I}(140)) \rightarrow \bar{\mathbb{S}}(\text{exclusivelyDependsOn})$
- $(\bar{\mathbb{S}}(\text{participationOf}) \wedge \mathbb{I}(143)) \rightarrow \bar{\mathbb{S}}(\text{exclusivelyDependsOn})$

Note that in the five previous formulae, the set of role names that are in the antecedents (i.e., {plays, directlyCauses, hasPart, dependsOn, and participationOf}) is disjoint from the set of role names that appear in the consequents (i.e., {instantiates, causes, mereologicallyOverlaps, exclusivelyDependsOn}). Therefore, if a role in the first set is made non-simple, it must be by reasons (1), or (2), but not due to (3). This allows us to substitute the non-simplicity of such role names in the first set by disjunctions of the antecedents in (1)–(3) that make them non-simple. Note that, since there is no (direct or indirect) source of non-simplicity for participationOf, it is impossible for “ $\bar{\mathbb{S}}(\text{participationOf}) \wedge \mathbb{I}(143)$ ” to be true, and hence participationOf has no influence on the non-simplicity of exclusivelyDependsOn.

- $(\mathbb{I}(22) \wedge (\mathbb{I}(180) \vee \mathbb{I}(181))) \rightarrow \bar{\mathbb{S}}(\text{instantiates})$
- $(\mathbb{I}(27) \wedge \mathbb{I}(176)) \rightarrow \bar{\mathbb{S}}(\text{causes})$
- $(\mathbb{I}(128) \wedge (\mathbb{I}(130) \vee \mathbb{I}(131))) \rightarrow \bar{\mathbb{S}}(\text{mereologicallyOverlaps})$
- $(\mathbb{I}(135) \wedge \mathbb{I}(140)) \rightarrow \bar{\mathbb{S}}(\text{exclusivelyDependsOn})$

Second step. We show the axioms forbidden by the non-simplicity of roles:

$\bar{\mathbb{S}}(\text{instantiates}) \rightarrow \neg\mathbb{I}(97)$	$\bar{\mathbb{S}}(\text{instantiates}) \rightarrow \neg\mathbb{I}(111)$	$\bar{\mathbb{S}}(\text{exclusivelyDependsOn}) \rightarrow \neg\mathbb{I}(142)$
$\bar{\mathbb{S}}(\text{memberOf}) \rightarrow \neg\mathbb{I}(98)$	$\bar{\mathbb{S}}(\text{memberOf}) \rightarrow \neg\mathbb{I}(112)$	
$\bar{\mathbb{S}}(\text{inducedBy}) \rightarrow \neg\mathbb{I}(99)$	$\bar{\mathbb{S}}(\text{inducedBy}) \rightarrow \neg\mathbb{I}(113)$	$\bar{\mathbb{S}}(\text{precedes}) \rightarrow \neg\mathbb{I}(158)$
$\bar{\mathbb{S}}(\text{associatedWith}) \rightarrow \neg\mathbb{I}(100)$	$\bar{\mathbb{S}}(\text{qualeOf}) \rightarrow \neg\mathbb{I}(114)$	$\bar{\mathbb{S}}(\text{before}) \rightarrow \neg\mathbb{I}(163)$
$\bar{\mathbb{S}}(\text{inheresIn}) \rightarrow \neg\mathbb{I}(101)$	$\bar{\mathbb{S}}(\text{plays}) \rightarrow \neg\mathbb{I}(115)$	$\bar{\mathbb{S}}(\text{starts}) \rightarrow \neg\mathbb{I}(165)$
$\bar{\mathbb{S}}(\text{beginPoint}) \rightarrow \neg\mathbb{I}(102)$	$\bar{\mathbb{S}}(\text{associatedWith}) \rightarrow \neg\mathbb{I}(116)$	$\bar{\mathbb{S}}(\text{during}) \rightarrow \neg\mathbb{I}(167)$
$\bar{\mathbb{S}}(\text{endPoint}) \rightarrow \neg\mathbb{I}(103)$	$\bar{\mathbb{S}}(\text{inheresIn}) \rightarrow \neg\mathbb{I}(117)$	$\bar{\mathbb{S}}(\text{meets}) \rightarrow \neg\mathbb{I}(169)$
$\bar{\mathbb{S}}(\text{exclusivelyDependsOn}) \rightarrow \neg\mathbb{I}(104)$	$\bar{\mathbb{S}}(\text{manifestedBy}) \rightarrow \neg\mathbb{I}(118)$	$\bar{\mathbb{S}}(\text{temporallyOverlaps}) \rightarrow \neg\mathbb{I}(170)$
$\bar{\mathbb{S}}(\text{participationOf}) \rightarrow \neg\mathbb{I}(105)$	$\bar{\mathbb{S}}(\text{triggers}) \rightarrow \neg\mathbb{I}(119)$	$\bar{\mathbb{S}}(\text{finishes}) \rightarrow \neg\mathbb{I}(171)$
$\bar{\mathbb{S}}(\text{dependsOn}) \rightarrow \neg\mathbb{I}(106)$	$\bar{\mathbb{S}}(\text{obtainsIn}) \rightarrow \neg\mathbb{I}(120)$	$\bar{\mathbb{S}}(\text{equals}) \rightarrow \neg\mathbb{I}(172)$
$\bar{\mathbb{S}}(\text{hasPart}) \rightarrow \neg\mathbb{I}(107)$	$\bar{\mathbb{S}}(\text{hasPart}) \rightarrow \neg\mathbb{I}(127)$	$\bar{\mathbb{S}}(\text{obtainsIn}) \rightarrow \neg\mathbb{I}(175)$
$\bar{\mathbb{S}}(\text{bringsAbout}) \rightarrow \neg\mathbb{I}(108)$	$\bar{\mathbb{S}}(\text{mereologicallyOverlaps}) \rightarrow \neg\mathbb{I}(129)$	$\bar{\mathbb{S}}(\text{directlyCauses}) \rightarrow \neg\mathbb{I}(178)$
$\bar{\mathbb{S}}(\text{obtainsIn}) \rightarrow \neg\mathbb{I}(109)$	$\bar{\mathbb{S}}(\text{inheresIn}) \rightarrow \neg\mathbb{I}(136)$	$\bar{\mathbb{S}}(\text{causes}) \rightarrow \neg\mathbb{I}(179)$
$\bar{\mathbb{S}}(\text{obtainsIn}) \rightarrow \neg\mathbb{I}(110)$	$\bar{\mathbb{S}}(\text{exclusivelyDependsOn}) \rightarrow \neg\mathbb{I}(141)$	

Note that, from the list above, it suffices to consider only the axioms forbidden by the non-simplicity of roles *that were mentioned in the previous step*, since other roles would have no other reason to be non-simple than (1)–(3). Therefore, we consider only:

- $\bar{\mathbb{S}}(\text{instantiates}) \rightarrow \neg(\mathbb{I}(97) \vee \mathbb{I}(111))$
- $\bar{\mathbb{S}}(\text{exclusivelyDependsOn}) \rightarrow \neg(\mathbb{I}(104) \vee \mathbb{I}(141) \vee \mathbb{I}(142))$
- $\bar{\mathbb{S}}(\text{dependsOn}) \rightarrow \neg\mathbb{I}(106)$
- $\bar{\mathbb{S}}(\text{hasPart}) \rightarrow \neg(\mathbb{I}(107) \vee \mathbb{I}(127))$
- $\bar{\mathbb{S}}(\text{plays}) \rightarrow \neg\mathbb{I}(115)$
- $\bar{\mathbb{S}}(\text{triggers}) \rightarrow \neg\mathbb{I}(119)$
- $\bar{\mathbb{S}}(\text{mereologicallyOverlaps}) \rightarrow \neg\mathbb{I}(129)$
- $\bar{\mathbb{S}}(\text{precedes}) \rightarrow \neg\mathbb{I}(158)$
- $\bar{\mathbb{S}}(\text{before}) \rightarrow \neg\mathbb{I}(163)$
- $\bar{\mathbb{S}}(\text{starts}) \rightarrow \neg\mathbb{I}(165)$
- $\bar{\mathbb{S}}(\text{during}) \rightarrow \neg\mathbb{I}(167)$
- $\bar{\mathbb{S}}(\text{meets}) \rightarrow \neg\mathbb{I}(169)$
- $\bar{\mathbb{S}}(\text{equals}) \rightarrow \neg\mathbb{I}(172)$
- $\bar{\mathbb{S}}(\text{directlyCauses}) \rightarrow \neg\mathbb{I}(178)$
- $\bar{\mathbb{S}}(\text{causes}) \rightarrow \neg\mathbb{I}(179)$

Third step. We perform a composition of the constraints generated in the first step with the constraints generated at the second step. This composition generates our meta-axioms (m2)–(m16) in Section 5.2.2 about the incompatibilities due to the simplicity rule.