STQC and Ternary Relationships in DL-based Languages

Writer: Peihong (Karl) Xie

Final Project Collaborator: Jieming Yu

Description logic (DL) is a decidable fragment of First-order Logic (FOL) that provides a logical and semantic foundation for many computational languages used by technologies relevant to databases and ontologies, such as RDF, SPARQL, and OWL. These languages can be collectively referred to as "DL-based languages" for the sake of simplicity. In a DL-based language, only unary relationships like "Fa" or binary relationships like "aRb" can be explicitly expressed in a straightforward way. For example, an unary relationship can be expressed as a RDF triple with the form of "ex:a rdf:type ex:F", and a binary relationship can also expressed as a RDF triple with the form of "ex:a ex:R ex:b". Then, a question naturally arises: How can a ternary or n-ary relationship be expressed in a DL-based language? This paper will focus on DL-based expressions of ternary relationships, especially time-indexed relationships such as "x is part of y at time t".

1 Reification and Its Process-based Variant

1.1 Standard Solution: Reification

In the community of Semantic Web Technology, a standard solution to this problem is called "reification" (Noy & Rector 2006). Roughly speaking, the basic insight behind reification is that for each temporalized ternary relationship R(x,y,t), we will introduce a new computational artifact class Q, and then utilizes three new binary relationships R1, R2, and R3 to assert the relationship between Q and each relata of R, respectively. (The same strategy, after necessary change, can also be applied to express any n-ary relationship.)

For example, suppose that we want to assert that Socrates taught Plato in 400BC, and this assertion clearly expresses a ternary relationship between Socrates, Plato, and the year of 400BC. Now given the strategy of reification, such a relationship can be expressed as a set of RDF triples if we introduce a computational artifact class "ex:Proposition" and take the assertion that Socrates taught Plato in 400BC as an instance of ex:proposition, say, ex:p1,:

```
ex:p1 a ex:Proposition ;
    ex:hasSubject ex:Socrates ;
    ex:hasObject ex:Plato ;
    ex:atYear ex:400BC .
```

Reification, though convenient and syntactically straightforward, is not satisfactory for three main reasons. First, the introduced computational artifact class is *ad hoc* and arbitrary. Second, new introduced connecting relationships R1, R2, and R3 lack necessary semantic support from a target ontology. Third, the special significance of time is not sufficiently represented in reification.

1.2 A Variant of Reification: The Process-based Strategy

A less semantically *ad hoc* strategy is to transform to a target ternary relationship to a corresponding *process* class. This process-based strategy has a similar syntactic treatment to the original version of reification, but gives a semantically stricter interpretation of the introduced computational artifact class. For example, to express in a DL-based language a ternary relationship asserted in the statement that *a* was caused by *b* at time *t*, we can transform the object property *wasCausedBy* to a process class, *Causation*, and then assert the following set of triples instead:

```
ex:process1 a ex:Causation ;
    ex:hasInput ex:b ;
    ex:hasOutput ex:a ;
    ex:OccupiesTemporalRegion ex:t .
```

Although the process-based variant of reification is appealing and semantically cheap, it is unable to be applied to all ternary relationships. It is because not every ternary relationship has a corresponding process class. For example, it is semantically counter-intuitive to claim that the temporalized mereological relationship, partOf(a,b,t), corresponds to some process class like *Parthood*. Parthood is generally taken to be a relationship rather than a process. In light of this, we have to find a more general strategy of expressing ternary relationships in a DL-based language. Here a solution called "the strategy of temporally qualified continuants" ("STQC" for short) catches our attention.

2 The Semantics of STQC

2.1 Ontological Background: Basic Formal Ontology

The strategy of TQC is developed against the background of Basic Formal Ontology (BFO). BFC is a top-level ontology that includes dozens of classes and their hierarchy enhanced by human-readable definitions and logical axioms. In principle, nearly all scientific domains have the potential to be subsumed under the BFO class hierarchy. Specifically speaking, each thing in the world is either a continuant or occurrent according to BFO. On the one hand, a continuant wholly exists at every temporal region where it exists. That is, a continuant has no temporal part, though it has spatial parts. Typical examples of continuants are books, rocks and cats. On the other hand, an occurrent unfolds itself along the temporal region it occupies. So it has not only spatial parts but also temporal parts. Typical examples of occurrents are the second world war, metabolism and cell division.

It is not difficult to express in BFO a temporalized ternary relationship $R(x,y,t)$ if it only involves occurrents. In this case, the relationship R can always be technically treated as a binary relationship between the occurrent x's and the occurrent y's respective temporal parts at t, so there is no need to explicitly refer to the particular time t. In other words, the time t seems to "disappear" from the BFO-based treatment just because it is somehow "built into" temporal parts of occurrents x and y.

Nonetheless, the same treatment cannot be directly applied to continuants because continuant has no temporal part unless we adopt a four-dimentionalist conception of continuants. However, a four-dimentionalist conception of continuants will contradict basic assumptions of BFO. As a result, some BFO developers have proposed a new solution called STQC, which is a technical counterpart of a four-dimentionalist ontology without involving any problematic ontological commitment contrary to BFO (Schulz & Hastings 2013, Jansen & Grewe 2014, Grewe et al. 2016, Grewe, Jansen & Smith 2016, and Beverley 2018).

In BFO, each material entity has a unique *history*, which is defined as the sum of all processes taking place in the spatiotemporal region occupied by it (Arp, Smith & Spear 2015: 122). Now according to STQC, the history of a material entity can be divided to different *phases*. For example, the history of philosopher Socrates includes his young phase, middle-age phase, old phase and so on. Each of these phases, as an occurrent part of Socrates' history, corresponds to a particular way Socrates is. Socrates'

young phase corresponds to *young Socrates*, Socrates' middle-age phase corresponds to *middle-age Socrates*, and Socrates' old phase corresponds to *old Socrates*. Advocates of STQC claim that young Socrates, middle-age Socrates and old Socrates are temporally qualified continuants (TQCs) of Socrates.

It is not difficult to find that TQCs of a material entity (as a continuant) play a similar representational role to temporal parts of an occurrent. With STQC, we can use a similar means to deal with temporalized ternary relationship R(x,y,t). If R involves at least one continuant as its relata, then a DL-based model can always refer to some TQC of its continuant relata without explicitly referring to a particular time. However, it is worth noting that TQCs are not on an ontological par with temporal parts. Advocates of STQC take their strategy as a *façon de parler* (way of speaking), but keep neutral on the ontological status of a TQC.

2.2 STQC's Connection to Other Philosophical Views
2.2.1 Kit Fine on Variable Objects and Their Manifestations
Kit Fine (1999)'s view of variable objects can be taken as a philosophical precursor of STQC to some degree. In Fine's view, an entity is either variable or non-variable. A variable entity has different manifestations at different times, while a non-variable entity has always the same manifestation.

It is not difficult to see that the relation of a variable entity to its manifestations is formally isomorphic to the relation of a continuant to its TQCs, though two approaches have different philosophical foundations. So when advocates of STQC talks about temporalized ternary relationships via TQCs, advocates of Fine can talks about them via manifestations instead. As Garbacz & Trypuz (2017) points out, this is what Makolab's R&D Group proposed in the 9th International Conference on Formal Ontology in Information Systems 2016.

Nonetheless, we may have reason to adopt STQC instead of Fine's view. After all, STQC is compliant to a very popular and successful top-level ontology (namely BFO), but Fine's view has not been supported by any top-level ontology yet. This fact means that STQC can be better to improve the interoperability of different domain ontologies, but does not lose any representational benefit, compared with Fine's view.

2.2.2 Bittner and Donnelly on Stage Mereology
Bittner & Donnelly (2004) developed a mereology of persistent entities, and its

treatment of time-dependent parthood for endurants is somewhat similar to STQC. The core notion in this mereology is the notion of a *stage*. Strictly speaking, a stage is an instantaneous time-slice of a perdurant or bfo:occurrent. At every moment an endurant or bfo:continuant *x* exists, there is a stage *y* which is a time-slice of its life or bfo:history *z* (as an occurrent). Moreover, the continuant x is exactly co-located with y. In light of this fact about co-location, Bittner and Donnelly argue that *y* is a stage of *x* in an extended sense.

Given their mereology of stages, we can define temporary, permanent and lifelong parthood relationships without explicitly referring to particular times (Bittner & Donnelly 2004: 286-287):

(1) The continuant x is a *temporary part* of the continuant y iff x has a stage which is part of some stage of y.

(2) The continuant x is a *permanent part* of the continuant y iff all stages of x is part of some stage of y.

(3) The continuant x is a *livelong part* of the continuant y iff all stages of x is permanent part of y and all stages of y have some stage of x as part.

As we will see below, STQC can be seen as an extension of Bittner and Donnelly's mereology. On the one hand, basic elements of their mereology are instantaneous stages while basic elements of STQC are TQCs, which can be instantaneous or non-instantaneous. This fact means that STQC will be somewhat more flexible in dealing with a binary relationship temporalized to a non-instantaneous interval than the mereology of stages, though the latter has also a way to do it. On the other hand, while the mereology of stages covers binary treatments of temporalized mereological relationships, STQC can cover binary treatments of not only temporalized mereological relationships but also other kinds of temporalized ternary relationships.

3 Syntactical Representations of STQC
3.1 A Terminological Shift
Schulz & Hastings (2013) and Jansen & Grewe (2014), as two relatively early versions of STQC, both proposed to introduce the following object property terms to characterize STQC:

x hasMax y iff x is a TQC, y is a continuant, and x is a TQC of y.

x maxOf y iff x is a continuant, y is a TQC, and y is a TQC of x.

x hasTime t iff t is a temporal region, x is a TQC, and t is a phase of x.

x atSomeTime y iff x hasMax • maxOf y (x and y are sibling TQCs in the sense that x and y are both TQCs of the same continuant).

However, this terminology does not show in a straightforward way how a TQC is connected to a (temporally unqualified) continuant. After all, there is no place of TQC in the BFO class hierarchy. So to be BFO-compliant, advocates of STQC has to find an indirect way to connect TQCs to bfo:continuants. An ideal medium is bfo:history because the unique history y of a continuant x can be qualified to a phase z of it, which identifies a unique TQC of x (namely w) that occupies z. However, the above terminology does not specify such an indirect connection.

This might be why Grewe, Jansen & Smith (2016) and Beverley (2018) proposes another terminology instead. In their terminology, *hasMax* is replaced with *TQC_of*, *maxOf* with *hasTQC*, and *hasTime* indirectly with *hasPhase*.[①] Accordingly, "x atSomeTime y" can be defined as "x TQC_of • has_TQC y".

Not every object property term is necessary to run STQC. For example, as Grewe et al. (2016) shows, it is enough for STQC to use only TQC_of and its inverse to run STQC.[②] While this is a possibility, another representationally equivalent possibility is to *use only phaseOf and its inverse to run STQC*. Here we prefer the second way because it shows the significance of bfo:history for STQC in a more straightforward way.

3.2 The Impact of the Graz Release of BFO

The Graz release of BFO (see [3] in references) is a version of BFO published in 2012. It divides many important BFO object properties (namely binary relationships) to two temporalized forms, R_at_some_time and R_at_all_times (here "R" is a placeholder for a BFO object property). Nonetheless, some early versions of STQC didn't reflect this move of the Graz release. For example, Schulz & Hastings (2013) and Jansen & Grewe (2014) still insisted on non-temporalized forms of relationships, and took for default the assumption that (non-time-indexed) R-ness involves an implicit universal quantification over time, and thus indicates *permanent generic relatedness* (PGR). PGR is a temporalized ternary relationship such that at every time, every X is R-related to

some Y but those Ys may be different at different times. This can be formalized in FOL:

(PGR-FOL) $\forall p \forall t \exists q \ (X(p) \wedge Y(q) \rightarrow R(p,q,t))$

So given the above default assumption, STQC can translate the FOL formula into a DL-based language:

(PGR-STQC) $\forall p \exists q \ (X(p) \wedge Y(q) \rightarrow R(p,q))$

By contrast, the Graz release explicitly reveals the quantification over time involved in R-ness. Later versions of BFO, for example, Grewe, Jansen & Smith (2016) and Beverley (2018), integrate the insight of the Graz release into them. According to Grewe, Jansen & Smith (2016), R_at_all_time indicates for default *permanent specific relatedness* (PSR), and R_at_some_time indicates *temporary relatedness* (TR).

PSR and TR are two different temporalized ternary relationships from PGR. On the one hand, PSR is such that for all Xs, there is some Y such that: x is R-related to y at all times. Unlike PGR, the instance of Y must be same at all times. The FOL formalization of PSR is:

(PSR-FOL) $\forall p \exists q \forall t \ (X(p) \wedge Y(q) \rightarrow R(x,y,t))$

Given the new default assumption, STQC can translate the FOL formula into a DL-based language:

(PSR-STQC) $\forall p \exists q \ (X(p) \ \wedge Y(q) \rightarrow R\_at\_all\_times(p,q))$

On the other hand, TR is such that for all Xs, there is some Y such that: x is R-related to y at some time. Its FOL formalization is:

(TR-FOL) $\forall p \exists q \exists t \ (X(p) \wedge Y(q) \rightarrow R(x,y,t))$

According to Grewe, Jansen & Smith (2016), this FOL formula might be translated into a DL-based language:

(TR-STQC1) $\forall p \exists q\ (X(p) \wedge Y(q) \rightarrow R\_at\_some\_time(p,q))$

However, Grewe, Jansen & Smith (2016) and Beverley (2018) actually give another DL-based translation involved only R_at_all_times. Beverley (2018) even claims that R_at_some_time is unnecessary for STQC. So another DL-based translation of TR without involving R_at_some_time can be as:

(TR-STQC2) $\forall p \exists ph \exists r \exists pt \exists q\ (X(p) \wedge Y(q) \rightarrow R\_at\_all\_times(pt,q) \wedge$
hasHistory(p,ph) $\wedge$ hasOccurrentPart(ph,r) $\wedge$ phaseOf(r,pt))

A constraint can be introduced to assert that (TR-STQC1) is logically equivalent to (TR-STQC2) if required.

Due to the impact of the Graz release, we can also give a new STQC-based translation of PGR:

(PGR-STQC*) $\forall p \exists ph \forall r \exists pt \exists q\ (X(p) \wedge Y(q) \rightarrow R\_at\_all\_times(pt,q) \wedge$
hasHistory(p,ph) $\wedge$ hasOccurrentPart(ph,r) $\wedge$ phaseOf(r,pt))

For all DL-based formalizations involving TQC in this paper, a convention is adopted here:

(Convention-1) The individual name "*xt*" refers to the TQC of the individual continuant *x* exactly when *x* exists at a temporal region *t*.

(Convention-1) is implicitly suggested by Beverley (2018)'s insight that TQC may be understood as a result of totally parametrizing the binary *existsAt* relation of BFO. For example, the TQC of John, *John2000-2085*, corresponds to *existsAtJohn2000-2085*, which is the total parametrization of existsAt (John, 2000-2085).

3.3 Two STQC-based Schemas

As we have seen, STQC-based representations of the above three temporalized ternary relationships, TR, PSR, and PQR, do not need to explicitly refer to the time parameter or third argument in original relational formulas. This is said to be an advantage of STQC. However, STQC is also supposed to be equipped with a capacity to refer to a

particular time parameter when necessary, especially when we have to represent a particular instance of a temporalized ternary relationship.

In fact, in a very early version of TQC, Schulz & Hastings (2013) has proposed a general schema to do this. According to their proposal, a particular temporalized ternary relationship R(a,b,t) can be translated into a conjunction of five role assertions[3] in a DL-based language (syntactic details have been adjusted according to (Convention-1)):

(STQC-Schema1) R(at,bt) $\wedge$ hasTime(at,t) $\wedge$ hasMax(at,a) $\wedge$ hasTime (bt,t) $\wedge$ hasMax(bt,b)

If R is a temporalized relationship between an instance and a class, for example, rdf:type or instanceOf, then the above conjunction would have the following simpler form because the class *b* as the second relata of R cannot be said to have TQCs:

(STQC-Schema2) R(at,b)$\wedge$hasTime(at,t)$\wedge$hasMax(at,a)

Given the aforementioned terminological shift and the impact of the Graz release, however, we have to make two necessary adjustments:

(1) Replace (STQC-Schema1) and (STQC-Schema2) with logically equivalent schemas which involves phaseOf (and/or its inverse) but neither hasMax nor hasTime.

(2) In the replacement course, keep in mind that hasTime cannot be *directly* replaced with hasPhase. It is because they have different ranges. While the range of hasTime is a set of temporal ranges, the range of hasPhase is a set of phases. A phase is not a temporal region, but an occurrent part of some history. So a new BFO-2020 object property namely *occupiesTemporalRegion* has to be introduced to specify the relationship between a phase (as a process) and its corresponding temporal region in our new schemas.

In light of these considerations, (STQC-Schema1) can be replaced with:

(STQC-Schema1*) R(at,bt)$\wedge$hasHistory(a,ah)$\wedge$hasOccurrentPart(ah,r)$\wedge$ phaseOf(r,at)$\wedge$occupiesTemporalRegion(r,t)$\wedge$hasHistory(b,bh)$\wedge$ hasOccurrentPart(bh,s) & phaseOf(s,bt) & occupiesTemporalRegion(s,t)

If R is a temporalized relationship between an instance and a class, (STQC-Schema2)

can be replaced with:

(STQC-Schema2*) R(*at*,*b*)∧hasHistory(*a*,*ah*)∧hasOccurrentPart(*ah*,*r*)∧
phaseOf(*r*,*at*)∧occupiesTemporalRegion(*r*,*t*)

In the above STQC schemas, "R(*at*,*bt*)" or "R(*at*,*b*)" could be taken as the *"main-branch" sub-formula*. If the main-branch sub-formula involves a TQC such as the TQC *at*, then a corresponding *"annotation-given" sub-formula* like "hasHistory(*a*,*ah*) ∧ hasOccurrentPart(*ah*,*r*) ∧ phaseOf(*r*,*at*) ∧ occupiesTemporalRegion(*r*,*t*)" has to be added to assert the connection between the TQC *at*, the continuant *a*, and the temporal region *t*. For example, (STQC-Schema1*) involves two TQCs, so it is finally a conjunction of a main-branch sub-formula and two annotation-given sub-formulas.

With (STQC-Schema1*) and (STQC-Schema2*), all particular instances of temporalized ternary relationships can have a DL-based formula in principle. Moreover, the introduction of occupiesTemporalRegion also equips STQC with a capacity to specify a particular time.

4. STQC and BFO Continuant Mereology
4.1 STQC-based Translations of BFO-FOL Axioms

BFO 2020 have both a FOL version and an OWL version. As we see, OWL is a DL-based language in the sense that it can explicitly express binary relationships at most. Therefore, some logical constraints involving temporalized ternary relationships in the FOL version of BFO cannot be directly applied to the OWL version of BFO. To capture those FOL-based constraints in OWL, STQC can help here. In this section, we will take FOL-based mereological axioms concerning continuants for example.

"BFO 2020 Continuant Mereology Axioms" (see [4] in references) is made up of 31 axioms, many of which can be expressed with R_at_all_times or R_at_some_time. In the end, we find that only six axioms among them have to be expressed via STQC. For the sake of representational simplicity, we adopt a new convention here:

(Convention-2) The variable name "xh" refers to the individual x's unique history.

With two conventions in this paper, the following are those six axioms, their FOL formulas and corresponding STQC-based translations:

[mqp-1] ContinuantPartOf is dissective on third argument, a temporal region.

(FOL-mqp-1) ∀p,q,r,s(continuantPartOf(p,q,r) ∧ temporalPartOf(s,r) → continuantPartOf(p,q,s))

(STQC-mqp-1) ∀p∀q∀r∀s∃ph∃u∃pr∃qh∃w∃qr∃u2∃ps∃w2∃qs
(**continuantPartOfAtAllTimes**(pr,qr) ∧ hasHistory(p,ph) ∧ hasOccurrentPart(ph,u) ∧ phaseOf(u,pr) ∧ occupiesTemporalRegion(u,r) ∧ hasHistory(q,qh) ∧ hasOccurrentPart(qh,w) ∧ phaseOf(w,qr) ∧ occupiesTemporalRegion(w,r) ∧ temporalPartOf(s,r) → **continuantPartOfAtAllTimes**(ps,qs) ∧ hasHistory(p,ph) ∧ hasOccurrentPart(ph,u2) ∧ phaseOf(u2,ps) ∧ occupiesTemporalRegion(u2,s) ∧ hasHistory(q,qh) ∧ hasOccurrentPart(qh,w2) ∧ phaseOf(w2,qs) ∧ occupiesTemporalRegion(w2,s))

[vjv-1] ProperContinuantPartOf is dissective on third argument, a temporal region.

(FOL-vjv-1) ∀p,q,r,s(properContinuantPartOf(p,q,r) ∧ temporalPartOf(s,r) → properContinuantPartOf(p,q,s))

(STQC-vjv-1) ∀p∀q∀r∀s∃ph∃u∃pr∃qh∃w∃qr∃u2∃ps∃w2∃qs
(**properContinuantPartOfAtAllTimes(pr,qr)** ∧ hasHistory(p,ph) ∧ hasOccurrentPart(ph,u) ∧ phaseOf(u,pr) ∧ occupiesTemporalRegion(u,r) ∧ hasHistory(q,qh) ∧ hasOccurrentPart(qh,w) ∧ phaseOf(w,qr) ∧ occupiesTemporalRegion(w,r) ∧ temporalPartOf(s,r) → **properContinuantPartOfAtAllTimes(ps,qs)** ∧ hasHistory(p,ph) ∧ hasOccurrentPart(ph,u2) ∧ phaseOf(u2,ps) ∧ occupiesTemporalRegion(u2,s) ∧ hasHistory(q,qh) ∧ hasOccurrentPart(qh,w2) ∧ phaseOf(w2,qs) ∧ occupiesTemporalRegion(w2,s))

[plp-1] Continuant part of is transitive at a time.

(FOL-plp-1) ∀a,b,c,t,t2(continuantPartOf(a,b,t) ∧ continuantPartOf(b,c,t2) ∧ temporalPartOf(t,t2) → continuantPartOf(a,c,t))

(STQC-plp-1) ∀a∀b∀c∀t∀t2∃ah∃u∃at∃bh∃w∃bt∃w2∃bt2∃ch∃z2∃ct2∃z∃ct (**continuantPartOfAtAllTimes**(at,bt) ∧ hasHistory(a,ah) ∧ hasOccurrentPart(ah,u) ∧ phaseOf(u,at) ∧ occupiesTemporalRegion(u,t) ∧ hasHistory(b,bh) ∧ hasOccurrentPart(bh,w) ∧ phaseOf(w,bt) ∧ occupiesTemporalRegion(w,t) ∧ **continuantPartOfAtAllTimes**(bt2,ct2) ∧ hasHistory(b,bh) ∧ hasOccurrentPart(bh,w2) ∧ phaseOf(w2,bt2) ∧ occupiesTemporalRegion(w2,t2) ∧ hasHistory(c,ch) ∧ hasOccurrentPart(ch,z2) ∧ phaseOf(z2,ct2) ∧ occupiesTemporalRegion(z2,t2) ∧ temporalPartOf(t,t2) → **continuantPartOfAtAllTimes**(at,ct) ∧ hasHistory(c,ch) ∧ hasOccurrentPart(ch,z) ∧ phaseOf(z,ct) ∧ occupiesTemporalRegion(z,t))

[xpg-1] ProperContinuantPartOf is transitive at a time.

(FOL-xpg-1) ∀a,b,c,t,t2(properContinuantPartOf(a,b,t) ∧ properContinuantPartOf(b,c,t2) ∧ temporalPartOf(t,t2) → properContinuantPartOf(a,c,t))

(STQC-xpg-1) ∀a∀b∀c∀t∀t2∃ah∃u∃at∃bh∃w∃bt∃w2∃bt2∃ch∃z2∃ct2∃z∃ct (**properContinuantPartOfAtAllTimes**(at,bt) ∧ hasHistory(a,ah) ∧ hasOccurrentPart(ah,u) ∧ phaseOf(u,at) ∧ occupiesTemporalRegion(u,t) ∧ hasHistory(b,bh) ∧ hasOccurrentPart(bh,w) ∧ phaseOf(w,bt) ∧ occupiesTemporalRegion(w,t) ∧ **properContinuantPartOfAtAllTimes**(bt2,ct2) ∧ hasHistory(b,bh) ∧ hasOccurrentPart(bh,w2) ∧ phaseOf(w2,bt2) ∧ occupiesTemporalRegion(w2,t2) ∧ hasHistory(c,ch) ∧ hasOccurrentPart(ch,z2) ∧ phaseOf(z2,ct2) ∧ occupiesTemporalRegion(z2,t2) ∧ temporalPartOf(t,t2) → **properContinuantPartOfAtAllTimes**(at,ct) ∧ hasHistory(c,ch) ∧ hasOccurrentPart(ch,z) ∧ phaseOf(z,ct) ∧ occupiesTemporalRegion(z,t))

[bdd-1] ContinuantPartOf is time indexed and has domain: continuant and range: continuant.

(FOL-bdd-1) $\forall$a,b,t(continuantPartOf(a,b,t) $\rightarrow$ instanceOf(a,continuant,t) $\wedge$ instanceOf(b,continuant,t) $\wedge$ instanceOf(t,temporalRegion,t))

(STQC-bdd-1) $\forall$a$\forall$b$\forall$t$\exists$ah$\exists$u$\exists$at$\exists$bh$\exists$w$\exists$bt
(**continuantPartOfAtAllTimes**(at,bt) $\wedge$ hasHistory(a,ah) $\wedge$ hasOccurrentPart(ah,u) $\wedge$ phaseOf(u,at) $\wedge$ occupiesTemporalRegion(u,t) $\wedge$ hasHistory(b,bh) $\wedge$ hasOccurrentPart(bh,w) $\wedge$ phaseOf(w,bt) $\wedge$ occupiesTemporalRegion(w,t) $\rightarrow$ **instanceOf**(at,continuant) $\wedge$ **instanceOf**(bt,continuant) $\wedge$ **instanceOf**(t,temporalRegion))

[kte-1] ProperContinuantPartOf is time indexed and has domain: continuant and range: continuant.

(FOL-kte-1) $\forall$a,b,t(properContinuantPartOf(a,b,t) $\rightarrow$ instanceOf(a,continuant,t) $\wedge$ instanceOf(b,continuant,t) $\wedge$ instanceOf(t,temporalRegion,t))

(STQC-kte-1) $\forall$a$\forall$b$\forall$t$\exists$ah$\exists$u$\exists$at$\exists$bh$\exists$w$\exists$bt
(**properContinuantPartOfAtAllTimes**(at,bt) $\wedge$ hasHistory(a,ah) $\wedge$ hasOccurrentPart(ah,u) $\wedge$ phaseOf(u,at) $\wedge$ occupiesTemporalRegion(u,t) $\wedge$ hasHistory(b,bh) $\wedge$ hasOccurrentPart(bh,w) $\wedge$ phaseOf(w,bt) $\wedge$ occupiesTemporalRegion(w,t) $\rightarrow$ **instanceOf**(at,continuant) $\wedge$ **instanceOf**(bt,continuant) $\wedge$ **instanceOf**(t,temporalRegion))

4.2 Usability Optimization of STQC

A natural worry with STQC is, it seems to require computational artifacts such as TQCs and phases, and relevant object properties such as TQC_of and phaseOf (and/or their inverses), both of which were not included in the representational structure of BFO before, so it will introduce a great complexity to BFO. However, many users of BFO are not experts on the matter of time, so BFO's interoperability will be seriously limited if its structure has to be supplemented with STQC-related terms. In light of this, it is better to find a way to improve STQC's usability without changing BFO's structure.

4.2.1 STQC in Tawny OWL

Grewe, Jansen & Smith (2016) once proposed to use Tawny OWL to optimize STQC's usability. Roughly speaking, Tawny OWL is a programming environment for ontology building. It is built in a modern Lisp dialect called "Clojure" and can introduce an arbitrary linguistic extension, so this is partly why it can automatically generate those STQC-based expressions involving non-BFO terms.

Finally, Grewe, Jansen & Smith (2016) implemented 13 test cases regarding STQC and got mostly successful validation results with one exception.

4.2.2 STQC in SHACL

John Beverley recently proposed another possible optimization plan regarding STQC, which is based on Shapes Constraint Language (SHACL for short, see [11] in references). SHACL can be used to provide sets of RDF triples called "SHACL shapes" to validate RDF graphs and ontologies, so it has been increasingly popular after being approved by W3C.

According to Beverley's proposal, we can use SHACL to design certain shapes that correspond to FOL constraints not captured by the BFO-OWL ontology, for example, the above six mereological axioms (with their STQC-based translations). In this way, STQC does need to occur in BFO itself so no oddity will be added to BFO. Rather, it is merely represented at the level of data validation.

5 Conclusion

Currently, the majority of papers in the literature on STQC primarily focus on comparing it to other strategies for expressing temporalized ternary relationships in OWL, as well as discussing its potential in distinguishing between TR, PSR, and PGR. However, there is a noticeable gap in the literature when it comes to addressing the evolution of STQC over time. This paper aims to fill this gap by tracking the changes in different versions of STQC and extending its capabilities to handle ternary relationship instances and important FOL axioms. In doing so, this paper plays a role of version control mechanism in the development of STQC and also points towards a future research direction, specifically, how to implement STQC for data validation.

References

[1] Arp, R., Smith, B., & Spear, A. D. (2015). *Building Ontologies with Basic Formal Ontology*. Cambridge, MA: MIT Press.

[2] Beverley, J. (2018). Binary Relations in OWL: Generic and Specific (blog). URL = <https://johnbeverley.com/blogic/2018/6/13/binary-relations-in-owl-generic-and-specific>.

[3] BFO 2 OWL Group. BFO 2 OWL Preview Release. URL = < http://purl.obolibrary.org/obo/bfo/2012-11-15-bugfix/bfo.owl>.

[4] BFO 2 OWL Group. BFO 2020 Continuant Mereology Axioms. URL = <https://github.com/BFO-ontology/BFO-2020/blob/master/21838-2/pdf/continuant-mereology.pdf>.

[5] Bittner, T. & Donnelly, M. (2004). The Mereology of Stages and Persistent Entities. In Lopez de Mantaras, R. & Saitta, L. (ed.), *Proceedings of the European Conference of Artificial Intelligence (ECAI04)*, Baltimore: IOS Press: 283-287.

[6] Fine, K. (1999). Things and Their parts. *Midwest Studies in Philosophy* 23(1): 61-74.

[7] Garbacz, P., & Trypuz, R. (2017). Representation of tensed relations in OWL: A survey of philosophically-motivated patterns. In *Metadata and Semantic Research: 11th International Conference, MTSR 2017, Tallinn, Estonia, November 28–December 1, 2017, Proceedings 11*: 62-73. Springer International Publishing.

[8] Grewe, N., Hastings, J., Jansen, L., Ruttenberg, A., Mungall, C., & Schulz, S. (2016). Expressing Time-dependent Relations through Temporal Qualifications. URL = < https://studylib.net/doc/6849748/1---basic-formal-ontology--bfo->.

[9] Grewe, N., Jansen, L., & Smith, B. (2016). Permanent generic relatedness and silent change. In *Formal Ontology and Information Systems*. CEUR, Vol. 1060:1-5.

[10] Jansen, L. and Grewe, N. (2014). Butterflies and Embryos: The Ontology of Temporally Qualified Continuants. In Herre, H., et al., editors. *Ontologies and Data in Life Sciences* (ODLS 2014): Freiburg, Germany, Oct 7–8; 2014. p. E1–5.

[11] Knublauch, H. & Kontokostas, D. (2017). *Shapes Constraint Language (SHACL)*. W3C Recommendation, World Wide Web Consortium (20 July 2017), URL = < https://www.w3.org/TR/shacl/>.

[12] Noy, N. & Rector, A. (2006). *Defining N-ary Relations on the Semantic Web*. W3C Working Group Note, World Wide Web Consortium (April 2006), URL = <http://www.w3.org/TR/swbp-n-aryRelations/>.

[13] Schulz, S. and Hastings, J. (2013). *Temporally qualified continuants for BFO 2 OWL: A Bottom-up View* (power point slides).

---

① In fact, hasTime and hasPhase, though intended to play similar roles in STQC, are not wholly

equivalent. This is why we have to introduce another BFO term OccupiesTemporalRegion to re-establish the connection to time (see below).

② Grewe et al. (2016) used *hasTemporalQualification* instead of *hasTQC*, and *temporalQualificationOf* instead of *TQC_of*, but this different is merely terminological.

③ In description logic, a binary relation is called a "role", so an expression of a binary relation is called a "role assertion".