Professorship of Data Engineering
School of Computation, Information and Technology
Technical University of Munich

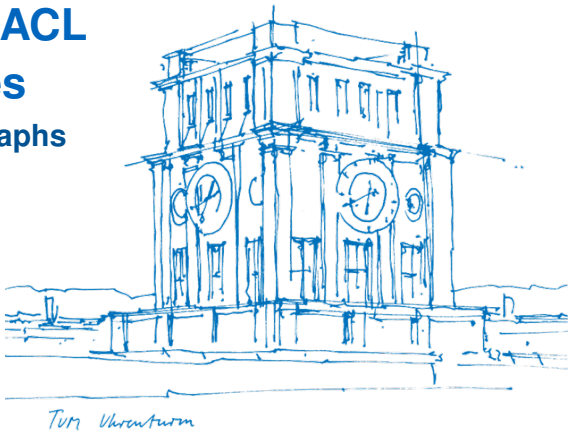
Effects of Entailment in SHACL Validation of Closed Shapes

Workshop on Quality of Knowledge Graphs (QKG) @ ESWC 2026

Zenon Zacouris **Jin Ke** **Maribel Acosta**

Technical University of Munich, Heilbronn

May 11th, 2026



Outline

Background

Closed shape

```
:s1 a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:closed true ;
  sh:property [sh:path :firstName] ;
  sh:property [sh:path :lastName] .
```

- A closed shape only allows explicitly listed properties.

Entailment

```
:Albert :firstName "Albert" .
:Einstein :lastName "Einstein" .
:Albert rdf:type :Person .

:lastName owl:EquivalentProperty :surname .
```

- Entailment derives logically implied triples from an ontology.

  $\Longrightarrow$:Einstein :surname "Einstein" .

Introduction

Closed-shape Constraint

Closed shapes enforce a strict whitelist of properties on a node.

e.g., a person can only be described by a first name and last name.

Entailment

Entailment derives logically implied triples.

e.g., each last name is also a surname.

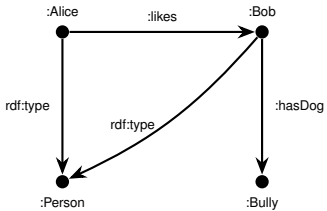
Conflict

Entailment may derive properties that are not permitted by a closed shape.

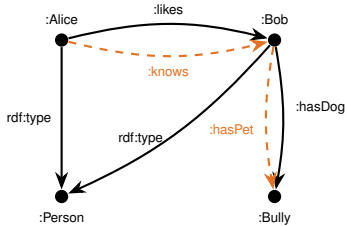
$\Longrightarrow$ previously conformant nodes can become non-conformant.

Motivating Example

Data graph



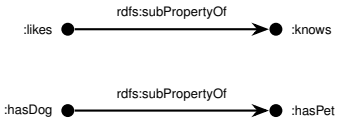
After entailment



Closed shape

```
:s1 a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:closed true ;
  sh:property [sh:path :knows]
      ;
  sh:property [sh:path :hasDog]
      ;
  sh:ignoredProperties (rdf:
      type) .
```

Ontology



Dashed orange edges are inferred triples.

	Without entailment	With entailment
:Alice	×	×
:Bob	✓	×

Effects of Entailment on Closed Shape Validation

Closed-shape validation in the presence of entailment is not well-defined.

Three possible cases

1. **Spurious violation:** a new predicate is entailed for a node based on one of its already permitted predicates
2. **Sound violation:** a new predicate is entailed for a node without being justified by one of its permitted predicates
3. **No effect:** entailment introduces no relevant new predicate for the node

Our goal: remove spurious violations without affecting sound ones.

OWL-LD Rules and Their Effects

Rule family	Effect	Reason
Subproperty	spurious	Adds a new predicate for the same subject.
Equiv. property	spurious	Adds an equivalent predicate for the same subject.
Domain	spurious	Adds rdf:type for the same subject.
Range	sound	Adds rdf:type to the object node, not the subject.
Inverse / symmetric	sound	Adds predicates to the object-side node.
Functional	sound	Adds owl:sameAs, not a relevant new closed-shape predicate.
Equality	sound	Propagates predicates through owl:sameAs.
Transitive	no effect	Adds triples but no new predicate.
Subclass	no effect	Adds rdf:type triples from existing rdf:type.
Other	no effect	Does not introduce relevant new predicates for a node.

Only rules that introduce new predicates for the same focus node can create closed-shape issues.

Outline

Related Work

Research Gap

Existing work rewrites shapes, reduces violations, or checks if constraints are preserved under entailment rules. None addresses how entailment introduces new predicates that break closed shapes.

Outline

ReC-SHACL

Key intuition behind ReC-SHACL

If a predicate is a logical consequence of the permissible properties of a closed shape, then it should also be treated as permissible.

Let Q be the original set of permissible properties of a closed shape.

ReC-SHACL computes an extended set Q^+ such that:

$$Q \subseteq Q^+$$

where Q^+ contains predicates that can be entailed from Q under ontology O and entailment regime R.

ReC-SHACL is implemented in 5 steps.

Step 1: Extract Permissible Properties

Closed shape

```
:s1 a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:closed true ;
  sh:property [sh:path :knows] ;
  sh:property [sh:path :hasDog]  ;
  sh:ignoredProperties (rdf:type) .
```

$$Q = \{\texttt{:knows},\ \texttt{:hasDog}, \texttt{rdf:type}\}$$

Step 2: Construct the Witness Graph

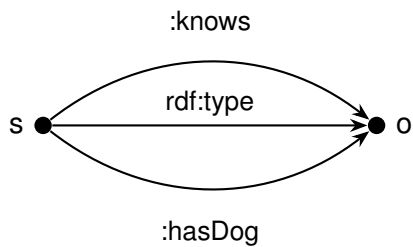
Input

$Q = \{\texttt{:knows},\ \texttt{:hasDog},\ \texttt{rdf:type}\}$

$$G_Q := \bigcup_{q \in Q \setminus \{\texttt{owl:sameAs}\}} \{(s, q, o)\}$$

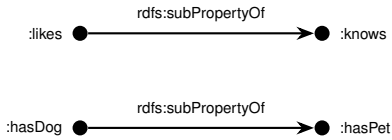
Witness graph

$$G_Q = \{(s, \texttt{:knows}, o),\ (s, \texttt{:hasDog}, o),\ (s, \texttt{rdf:type}, o)\}$$

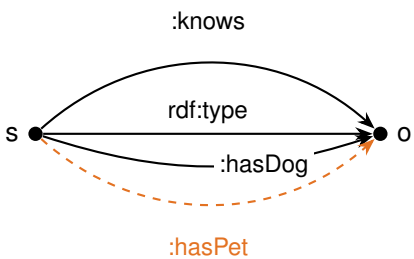


Step 3: Compute Closure under Ontology O

Ontology O

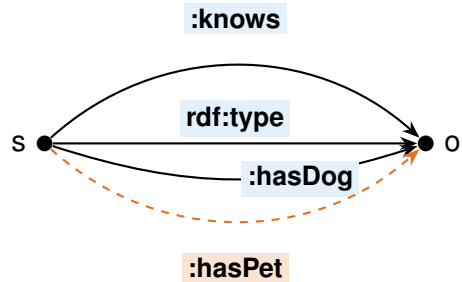


Closure of G_Q



Dashed orange edge is inferred.

Step 4: Collect Outgoing Predicates from s



$$Q^+ = \{\texttt{:knows},\ \texttt{:hasDog},\ \texttt{rdf:type},\ \texttt{:hasPet}\}$$

Step 5: Rewrite the Closed Shape

Extend permissible properties from Q to Q^+

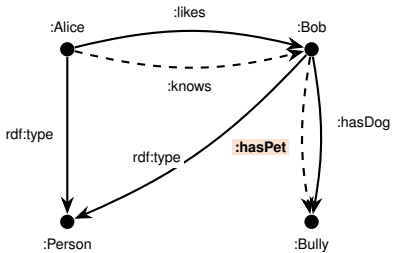
```
:s1 a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:closed true ;
  sh:property [sh:path :knows] ;
  sh:property [sh:path :hasDog] ;
  sh:property [sh:path :hasPet] ;
  sh:ignoredProperties (rdf:type) .
```

$$Q^+ \setminus Q = \{\texttt{:hasPet}\}$$

Result: `:hasPet` is no longer a closed-shape violation.

Revisited Example

Entailed data graph



Dashed edges are inferred triples.

Rewritten closed shape

```
:s1 a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:closed true ;
  sh:property [sh:path :knows] ;
  sh:property [sh:path :hasDog] ;
  sh:property [sh:path :hasPet]  ;
  sh:ignoredProperties (rdf:type) .
```

	Without entailment	With entailment
:Alice	×	×
:Bob	✓	✓

Outline

Experimental Setup

Data graph statistics

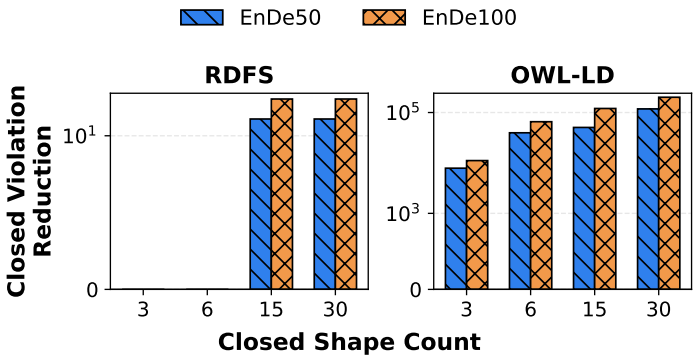
Data Graph	Triples	Subjects	Predicates	Objects
DBpedia (DB50)	534,696	134,303	2,888	140,855
DBpedia (DB100)	912,108	208,836	3,463	229,587

Shapes graph statistics

Shapes Graph	Node Shapes	Property Shapes	Targeted Classes	Targeted Props	Closed	Constraints
C_3	30	611	30	255	3	707
C_6	30	611	30	255	6	707
C_{15}	30	611	30	255	15	707
C_{30}	30	611	30	255	30	707

Shape variants differ only in the number of closed node shapes.

Reduction in Spurious Violations



↑↑ Higher is better

Figure 1 Spurious violations are resolved for both RDFS and OWL-LD, but more significantly for OWL-LD.

Runtime Analysis

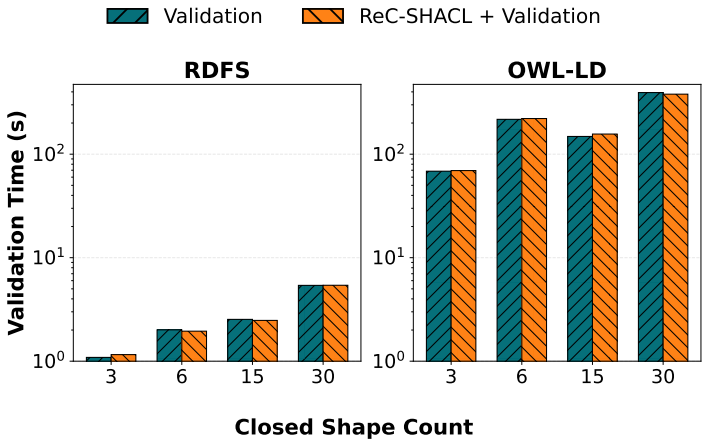


Figure 2 Total execution time.

↓↓ **Lower is better**

Runtime Analysis

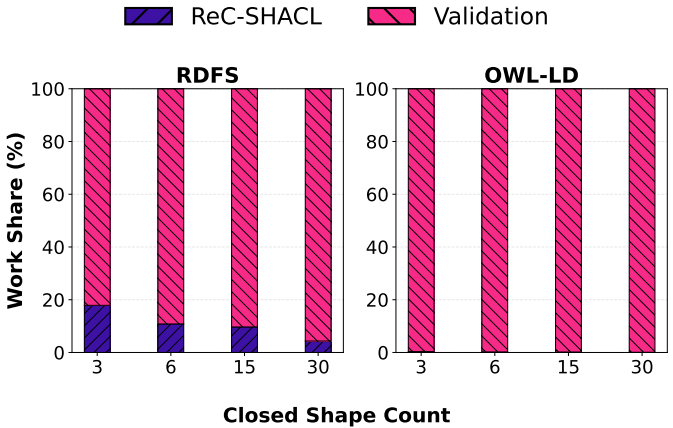


Figure 3 ReC-SHACL runtime has a negligible impact on the total validation runtime

Outline

Conclusion

- Closed-shape validation under entailment is not well-defined.
- Entailment can cause spurious, sound, or no changes in the validation outcome.
- ReC-SHACL extends permissible properties using shape rewriting.
- The resulting shapes graph can be used with any off-the-shelf validator.
- Our approach removes spurious closed-shape violations with negligible runtime overhead.

Thank you