rdfs:states

Thomas Lörtsch

RDF-star WG meeting

TPAC 2024
Anaheim CA, USA
hybrid meeting
23–27 SEPTEMBER 2024

Map Turtle-star annotation syntax to 'rdfs:states'

... or learn to live with another source of frustration in RDF

- "unasserted assertions" use cases
- popular intuitions
- reification too unspecific to meet them all
- Turtle-star syntactic sugar coming to the rescue
- but mappings to N-Triples-star loosing it again
- therefore proposing a second property, `rdfs:states`, to capture those intuitions

See RDF-star WG Github issue #128 for details and discussion

Use cases

"unasserted assertions"

RDF standard reification
RDF* CG

- versioning
- staging
- competing viewpoints, e.g. fact checking, health care, cultural studies

Qualification

- annotating a statement in the graph
- a statement and "its" annotation should be closely connected

Intuitions

A logic-driven intuition:

If it's in the graph, it is true (otherwise it's unknown).
If it's added later, it becomes true.

A lay person's intuition:

The world is not flat: there may be competing 'truths' (maybe not unknown, but unwelcome), "unasserted" may mean "explicitly not endorsed" (independent of other statements).

A Property Graph intuition:

"The edge EXISTS", i.e. there are no "unasserted assertions"".
The statement being annotated is rather understood to be a given.

Reification

Semantically a **very** weak construct with almost no meaning:
- describes a triple, but doesn't assert it
- just refers to a possible instance (wherever, whenever, whatever...)

That is useful to avoid problems in the logic,
but it doesn't properly meet popular intuitions.

Extra clarification would need to be provided (and standardized)
to enable sound integration of data from different sources.

Entailment

Entailment can provide a solid link between a triple and its annotation,
but is only available in the upper layers of RDF.

Syntactic sugar

N-triples-star

```
:r rdf:reifies <<( :s :p :o )>> .
```

Turtle-star

Unasserted syntax

```
<< :s : p :o >> :a :b .
```

Annotation syntax

```
:s :p :o {| :a :b |} .
```

Turtle-star meets the intuitions of both "lay persons" and the Property Graph aspect.
However, what it expresses is not what is stored in the database as N-triples-star.

Mapping

No roundtripping, jeopardizing monotonicity

```
<< :Foo :madeOf :Bar >> :says :Alice .
:Foo :madeOf :Bar {| :says :Bob |} .
```

<== imagine 5 lines of N-triples-star ==>

```
:Foo :madeOf :Bar {| :says :Alice |} ,
                  {| :says:Bob |} .
```

Imagine Alice's intent being to describe but not endorse the annotated statement.
That intent will be completely overridden.
Proper support for "unasserted assertions" shouldn't allow that.

Mapping

No roundtripping, jeopardizing disambiguation

```
<< :Foo :madeOf :Bar >> :becauseOf :TheoryOne .
<< :Foo :madeOf :Bar >> :becauseOf :TheoryTwo .
:Foo :madeOf :Bar {| :becauseOf :TheoryThree |} .
```

<== imagine 7 lines of N-triples-star ==>

```
:Foo :madeOf :Bar {| :becauseOf :TheoryOne |} ,
                  {| :becauseOf :TheoryTwo |} ,
                  {| :becauseOf :TheoryThree |} .
```

Imagine this in the medical domain, discussing different treatments against a new virus.

Proposal

A new property `rdfs:states` to enable explicit disambiguation

Because different syntactic sugars in Turtle-star represent different intuitions
they are mapped to different properties in N-triples-star:

unasserted syntax => `rdf:reifies`

annotation syntax => `rdfs:states`

Just a syntactic macro - no entailment needed -
but properly expressing the intent conveyed by the syntax.

Round tripping now works as expected.

Discussion

No solution ever comes without its own issues...

- Users have to be nudged into using Turtle-star and SPARQL-star.
In N-triples-star there's a risk of "dangling annotations" or a three-valued model.

- Querying for all annotated statements requires a join over `rdf:reifies` and `rdfs:states`. However, most queries can be expected to only address annotations on asserted statements and will not require any extra effort.

- Introducing a second property can also bring new problems, but the correspondence to the syntactic sugar in Turtle-star suggests that it meets requirements and intuitions pretty well.

- Use cases suggest that the ability to express "assertedness" is an important aspect in certain scenarios. Therefore we would need to address this problem anyway. Integrating the solution into the core makes most people's life a lot easier.

p.s.:

the issue can't be postponed until more experience is gathered...

... because what the syntactic sugar in Turtle-star is mapped to is defined now,
and that can't be changed later on.

this is not about propositional attitudes...

... which are a much trickier issue. The question is really very basic:
is the annotation supposed to refer to a statement considered true, or not