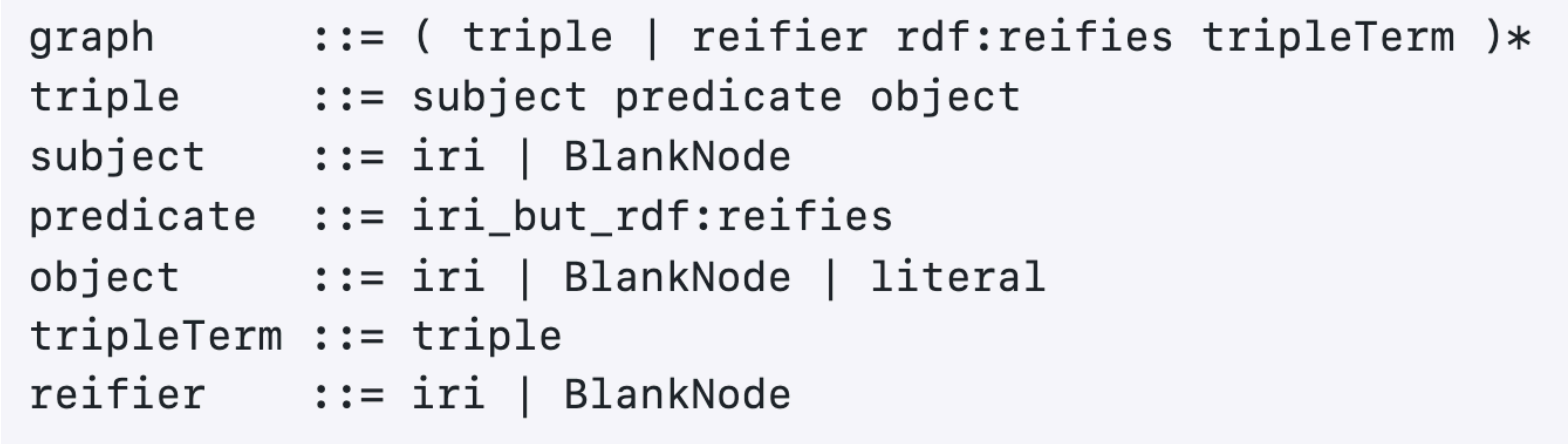
RDF-star profiles

what can I live with?

Enrico Franconi 25 April 2024

Profile 1 (many-to-many)

Syntax:



```
graph      ::= ( triple | reifier rdf:reifies tripleTerm )*
triple     ::= subject predicate object
subject    ::= iri | BlankNode
predicate  ::= iri_but_rdf:reifies
object     ::= iri | BlankNode | literal
tripleTerm ::= triple
reifier    ::= iri | BlankNode
```

Simple entailment

- same definition as RDF-1.1
- based on graph matching
- extension of RDF 1.1 matching for both entailment and SPARQL

Semantics:

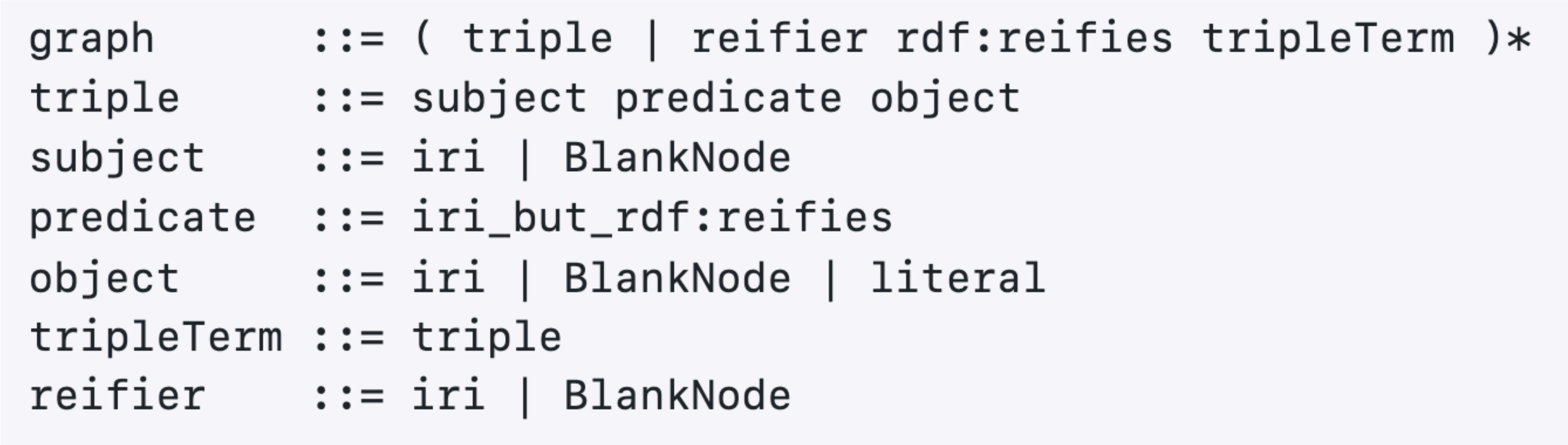
simple one-line extension of standard RDF 1.1 model-theoretic semantics:
`https://github.com/w3c/rdf-star-wg/wiki/RDF%E2%80%90star-semantics:-option-3`

Issues

- can cover exactly **profile 2-transparent** with a simple OWL construct: `owl:FunctionalProperty`

Profile 2-transparent (many-to-one)

Syntax:



```
graph      ::= ( triple | reifier rdf:reifies tripleTerm )*
triple     ::= subject predicate object
subject    ::= iri | BlankNode
predicate  ::= iri_but_rdf:reifies
object     ::= iri | BlankNode | literal
tripleTerm ::= triple
reifier    ::= iri | BlankNode
```

+ additional condition restricting `rdf:reifies` to be semantically functional

Semantics:

simple one-line extension of standard RDF 1.1 model-theoretic semantics:
`https://github.com/w3c/rdf-star-wg/wiki/RDF%E2%80%90star-semantics:-option-3`
+ condition on functionality of `rdf:reifies`

Simple entailment

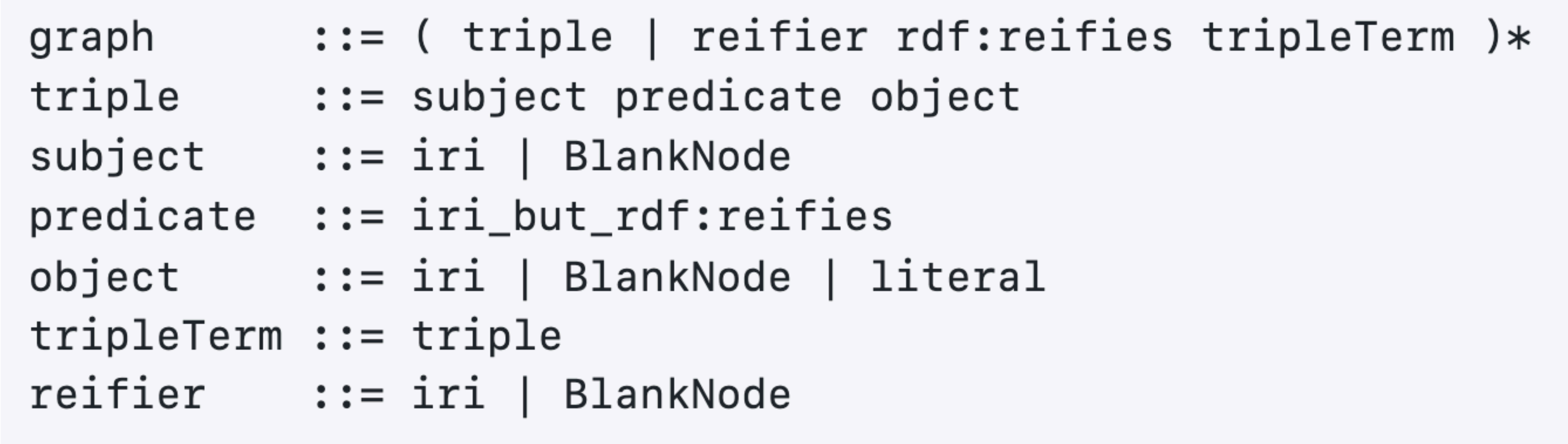
- same definition as RDF-1.1
- not based on graph matching
- not an extension of RDF 1.1 matching for both entailment and SPARQL

Issues

- serious implementation change
- unwanted entailments and SPARQL answers
- bad for merge and data integration
- does not cover all use cases

Profile 2-opaque (many-to-one)

Syntax:



```
graph      ::= ( triple | reifier rdf:reifies tripleTerm )*
triple     ::= subject predicate object
subject    ::= iri | BlankNode
predicate  ::= iri_but_rdf:reifies
object     ::= iri | BlankNode | literal
tripleTerm ::= triple
reifier    ::= iri | BlankNode
```

+ additional condition restricting `rdf:reifies` to be semantically functional

Simple entailment

- same definition as RDF-1.1
- not based on graph matching
- not an extension of RDF 1.1 matching for both entailment and SPARQL

Semantics:

simple one-line extension of standard RDF 1.1 model-theoretic semantics:
`https://github.com/w3c/rdf-star-wg/wiki/RDF%E2%80%90star-semantics:-option-3`

+ condition on functionality of `rdf:reifies`

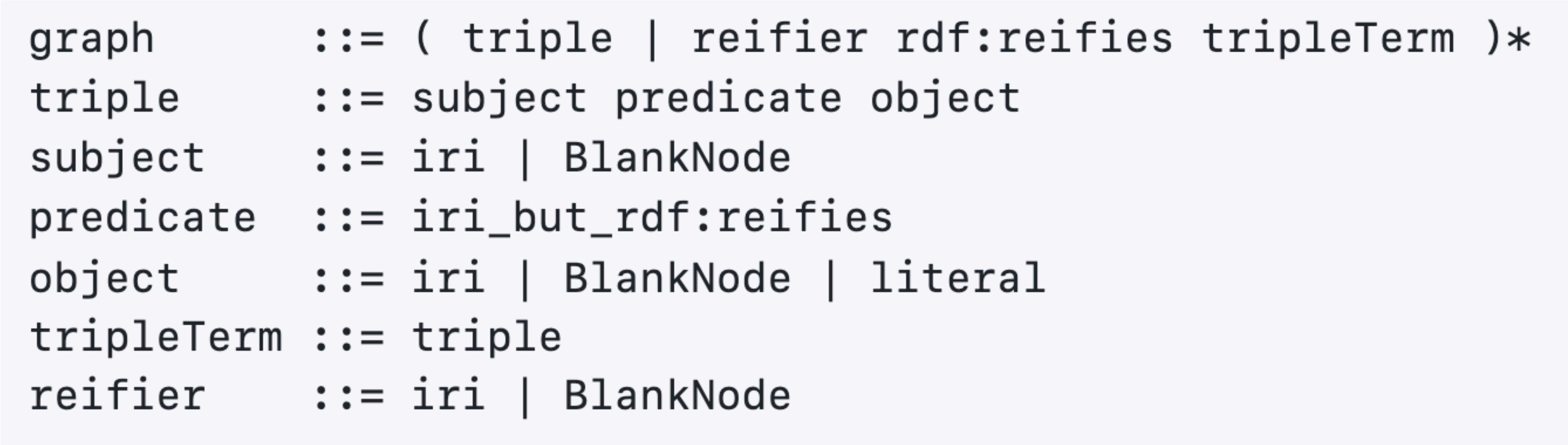
+ A `tripleTerm` is interpreted as a canonical literal

Issues

- serious implementation change
- does not cover all use cases

Profile 2-joke (many-to-one)

Syntax:



```
graph       ::= ( triple | reifier rdf:reifies tripleTerm )*
triple      ::= subject predicate object
subject     ::= iri | BlankNode
predicate   ::= iri_but_rdf:reifies
object      ::= iri | BlankNode | literal
tripleTerm ::= triple
reifier     ::= iri | BlankNode
```

+ additional condition restricting `rdf:reifies` to be semantically (?) functional

Semantics:

(coincides with profile 1)

Simple entailment

- coincides with profile 1

Issues

- what's the difference with profile 1?