FACULTY OF ENGINEERING AND ARCHITECTURE

IDLab
INTERNET & DATA LAB

Notation3 as the rule language for the Semantic Web

Dörthe Arndt, March 2019

GHENT UNIVERSITY

imec

There is a need for rules in the semantic Web

On the Semantic Web mailing list there was recently a discussion about **improving RDF**.

One important point was the **lack of a standard rule language.**

8. **Lack of a standard rules language.** This is a big one. Inference is fundamental to the value proposition of RDF, and almost every application needs to perform some kind of application-specific inference. ("Inference" is used broadly herein to mean any rule or procedure that produces new assertions from existing assertions -- not just conventional inference engines or rules languages.) But paradoxically, we still do not have a *standard* RDF rules language. (See also Sean Palmer's apt observations about N3 rules.[14]) Furthermore, applications often need to perform custom "inferences" (or data transformations) that are not convenient to express in available (non-standard) rules languages, such as RDF data transformations that are needed when merging data from independently developed sources having different data models and vocabularies. And merging independently developed data is the *most* fundamental use case of the Semantic Web.

One possibility for addressing this need might be to embed RDF in a full-fledged programming language, so that complex inference rules can be expressed using the full power and convenience of that programming language. Another possibility might be to provide a convenient, standard way to bind custom inference rules to functions defined in a programming language. A third possibility might be to standardize a sufficiently powerful rules language.

However, see also some excellent cautionary comments from Jesus Barras(Neo4J) and MarkLogic on inference: "No one likes rules engines --> horrible to debug / performance . . . Reasoning with ontology languages quickly gets intractable/undecidable" and "Inference is expensive. When considering it, you should: 1) run it over as small a dataset as possible 2) use only the rules you need 3) consider alternatives."[15]

Connection to RDF

The Semantic Web Rule language needs to have a strong connection to RDF.

Notation3 Logic

Notation3 Logic is an **extension of RDF**.

All RDF triples are also valid in N3

```
:lisa :isDaughterOf :homer.
```

Rules

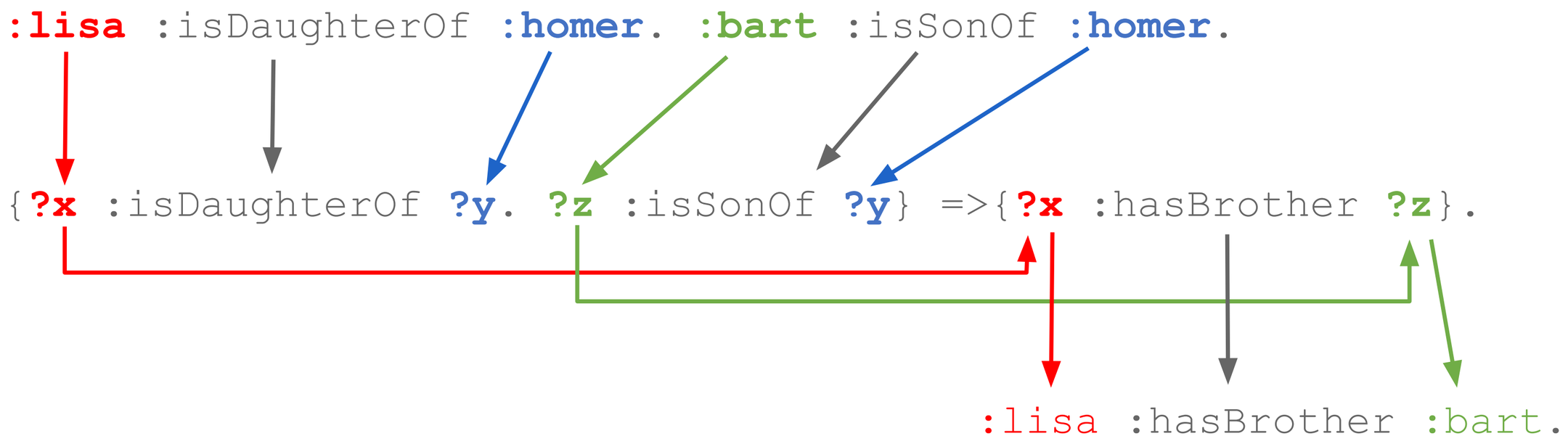
Rules are written using RDF turtle, a graph notation {} and an implication arrow =>:

```
{?x :isDaughterOf ?y. ?z :isSonOf ?y}
                                =>{?x :hasBrother ?z}.
```

“If x is the daughter of y and z is the son of y, then z is the brother of x.”

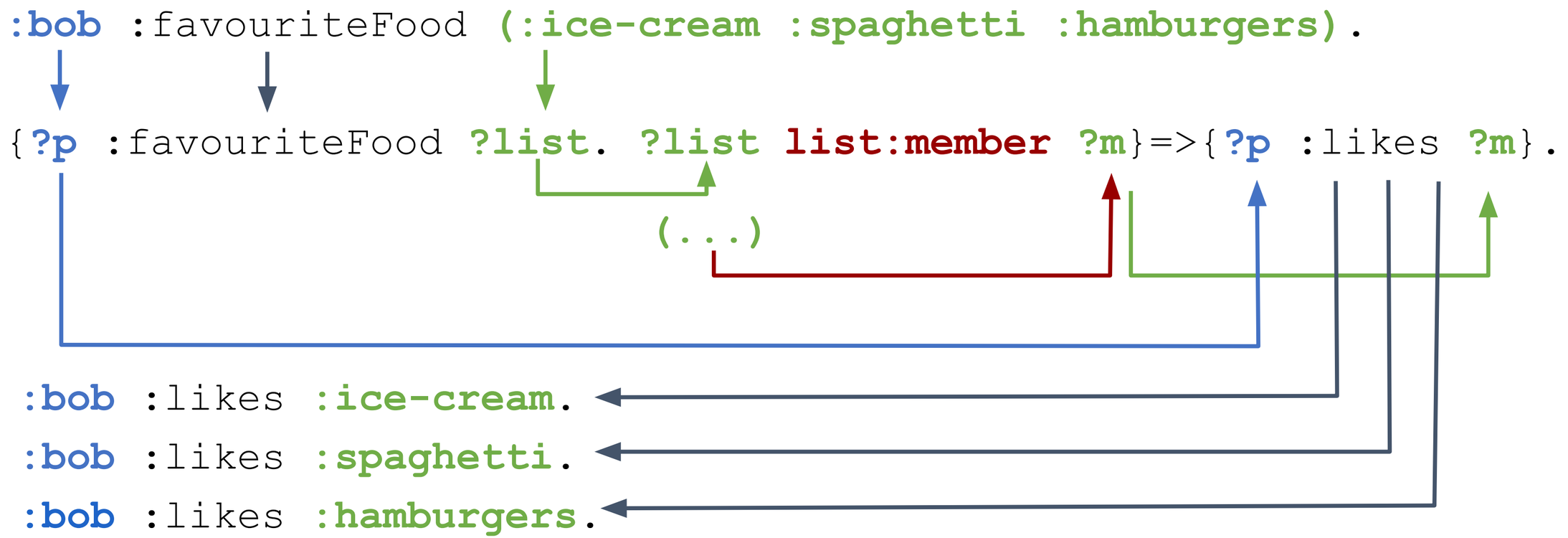
Application of rules

Rules can be applied to RDF triples:



Built-in functions can be used to operate on triples

Example: `list:member`



Built-in functions

Which built-in functions do we need?

Forward vs. backward

In N3 it is allowed to write rules either in a foward way or in a backward way. Reasoners could use that as an indication to either do forward or backward reasoning.

```
Forwards
{?x :isDaughterOf ?y. ?z :isSonOf ?y}=>{?x :hasBrother ?z}.
```

Backwards:

```
{?x :hasBrother ?z}<={?x :isDaughterOf ?y. ?z :isSonOf ?y}.
```

With backward reasoning rules we can do logical programming (e.g. like Prolog).

Citation of graphs

N3 Logic allows the citation of graphs

```
:lisa :says {:bob :likes :ice-cream}.
```

“Lisa says that Bob likes ice-cream.”

Lists are first-class citizens

```
:bob :favouriteFood (:ice-cream :spaghetti :hamburgers).
```

is **different** from

```
:bob :favouriteFood _:b1 rdf:first :ice-cream .
_:b1 rdf:rest _:b2 .

_:b2 rdf:first :spaghetti .
_:b2 rdf:rest _:b3 .

_:b3 rdf:first :hamburgers .
_:b3 rdf:rest rdf:nil .
```

GHENT UNIVERSITY imec

Blank nodes and literals in all positions of a triple

```
“ABC” a :Literal.

:lisa _:x :bart.
```

Both triples are valid N3.


GHENT
UNIVERSITY


imec
IDLab
INTERNET & DATA LAB