Linguaggi Logiche e Tecnologie per la Gestione Semantica dei testi

RDF+
Ontology languages

• RDF
• RDF(S)
• Ontology Web Language (OWL)
  • OWL lite- OWL DL- OWL Full – OWL2

expressiveness vs. complexity
Ontology Languages

• The expressiveness/complexity depends on the “tools” a language features

• RDF

• RDF (S)

• OWL
RDF-plus (RDF+)

• Augmenting RDF(S) with “a few” OWL constructs

Remark
The semantics of a statement is given by the inference that it can be drawn from it
• In mathematics the inverse of a function $f$, denote as $f^{-1}$ is the function that satisfies the property:

$$f(x) = y \rightarrow f^{-1}(y) = x$$

• In OWL the inverse of a property $P$ is another property $Q$

$$P \text{ owl:inverseOf } Q$$

$$x \ P \ y \rightarrow y \ Q \ x$$
RDF+: owl:InverseOf

sw:Joe swt:hasChild sw:Syd

sw:hasChild owl:inverseOf sw:hasParent

inference

sw:Syd swt:hasParent sw:Joe
Example - owl:InverseOf

• Suppose to have to merge data where the domain and range of two properties are reversed.
Example - **owl:InverseOf**

- As first step we need to **invert** domain and range!
RDF+  owl:SymmetricProperty

• Only concerns one property

\[
\text{P rdf:type owl:SymmetricProperty}
\]

\[
\text{P owl:inverseOf P }
\]
sw:marriedTo rdf:type owl:SymmetricProperty
• In mathematics a relation $R$ is said to be transitive if:

$$R(a, b), R(b, c) \rightarrow R(a, c)$$

$P$ $\text{rdf:type}$ $\text{owl:TransitiveProperty}$
RDF+ : \texttt{owl:TransitiveProperty}

- Relating Parents to Ancestors
- Parents are NOT transitive
  - My parents’ parents are NOT my parents
- Ancestors are transitive
  - My parents’ ancestors are my ancestors too

How to encode this information?

using \texttt{rdfs:subPropertyOf} and \texttt{owl:TransitiveProperty}
RDF+: owl:TransitiveProperty

sw:Joe

sw:Greg

sw:Dom

sw:hasParent

swt:hasParent

sw:hasAncestor

rdfs:subPropertyOf
RDF+ \texttt{owl:equivalentClass}

• When 2 classes have the \texttt{same members} they are defined to be equivalent.
• The use of \texttt{owl:equivalentClass} does not imply class equality.
• Class equality means that the classes have the same intensional meaning (denote the same concept).
RDF+ owl:equivalentClass

A owl:equivalentClass B

r rdf:type A

inference

r rdf:type B

B rdfs:subClassOf A

r rdf:type B

inference

r rdf:type A
RDF+: owl:equivalentClass

sw:Giuseppe rdf:type sw:Analyst

sw:Analyst owl:equivalentClass sw:Researcher

sw:Giuseppe rdf:type sw:Researcher
RDF+ `owl:equivalentProperty`

- When two properties are equivalent:
  - in any triple that uses one property as a predicate, the other property can be substituted.

```
P `owl:equivalentProperty` Q
A P B
```

Diagram:

```
A Q B
inference
```

```
P `owl:equivalentProperty` Q
A P B
```
The construct `owl:sameAs` links an individual to an individual.
The statement indicates that two URI references actually refer to the same thing: the individuals have the same "identity".

\[
A \overset{\text{owl:sameAs}}{\rightarrow} B
\]

sw:Joe \overset{\text{owl:sameAs}}{\rightarrow} ns1:Giuseppe
A functional property can only take one value for any particular individual.

Only one value as object

\[ \text{P rdf:type owl:FunctionalProperty} \]

\[ \text{X P A} \]

\[ \text{X P B} \]

\[ \text{inference} \]

\[ \text{A owl:sameAs B} \]
RDF+ owl:FunctionalProperty

sw:hasSquare

X \(\rightarrow\) Y

X \(\rightarrow\) Z

sw:hasSquare owl:FunctionalProperty

inference

Y \(\rightarrow\) Z

owl:sameAs
• Expresses the opposite of \textit{owl:FunctionalProperty}

• Only one value as subject

$P \text{ rdf:type } \textit{owl:InverseFunctionalProperty}$

\[
\begin{array}{c}
A \ P \ X\\
B \ P \ X
\end{array}
\]

\[
\downarrow \ \text{inference}
\]

$A \ \textit{owl:sameAs} \ B$
RDF+  owl:InverseFunctionalProperty

Y \rightarrow_{sw:hasSquareRoot} X

Z \rightarrow_{sw:hasSquareRoot} X

sw:hasSquare  owl:InverseFunctionalProperty

Y \rightarrow_{owl:sameAs} Z