COSC410

The Web Ontology Language: OWL

First, a bit more RDF

- In natural language, we can say "Wallace eats Wensleydale".
- We can also say "Grommit says 'Wallace eats Wensleydale'".
- The second statement does not imply the first: Grommit might be lying or wrong.

Reification in RDF

- :Grommit :says [
  a rdf:statement;
  rdf:subject :Wallace;
  rdf:predicate :eats;
  rdf:object :Wensleydale].
- There is no semantic law or inference procedure that concludes
  * *:Wallace :eats :Wensleydale*

Missing from SPARQL

- If we believe that Grommit is never mistaken about Wallace and never lies, we might want to have an inference rule
- {:Wallace ?r ?o} ⇐

  {:Grommit :says [
  rdf:subject :Wallace;
  rdf:predicate ?r;
  rdf:object ?o]}

Reusable views, please

- view wallace(?r ?o) = {:Wallace ?r ?o}
  ∪ {:Grommit :says [rdf: subject :Wallace;
  rdf: predicate ?r; rdf:object ?o]}
- We can expand this by hand every time we want it, but there are no reusable views in SPARQL.
- Time for a richer language…

"The" Web Ontology Language?

- There are three in OWL 1
- OWL Lite
- OWL DL
- OWL Full
- The OWL2 revision changed all of them, but there are still three.

OWL Lite

- Lets you define a classification hierarchy and simple constraints. It permits simpler tools and polynomial time algorithms.
- You can represent ISA hierarchies and PARTOF hierarchies.

OWL DL

- Aims to be more expressive while remaining completeness (if it's true in the semantics, a program can show it's true) and decidability.
- "DL" stands for "Description Logic".
- Valid inference in OWL Lite ⇒

  valid inference in OWL DL

OWL Full

- Full expressiveness, that is. There are no guarantees that anything is computable. If you use OWL Full you have no right to expect there to be a tool can handle what you've written.
- Valid inference in OWL DL ⇒

  valid inference in OWL Full.

OWL and RDF

- OWL Full extends RDF; OWL DL and OWL Lite build on a restriction of RDF.
- RDF lets you use a resource as a concept (type), rôle (predicate), and individual (resource) all at the same time.
- OWL DL and Lite do not allow a resource to be both a concept and an individual.

To ensure this:

- Declare every concept to be of type owl:Class.
- Every rôle should be an rdf:Property.
- Every individual should be declared to belong to some type, even if that's just owl:Thing.

OWL Lite classes

- There is a class hierarchy.
- owl:Nothing is the empty class.
- owl:Thing is the most general class.
- owl:Class is the type of classes.
- c1 rdfs:subClassOf c2
- c1 owl:equivalentClass c2

Class intersection

- intersectionOf
- obviously related to ⊓ in description logic

OWL Lite properties I

- There are properties.
- rdf:Property is the type of properties.
- p1 rdfs:subPropertyOf p2 — hierarchy.
- rdfs:domain, rdf:range
- p1 equivalentProperty p2

OWL Lite properties 2

- p1 owl:inverseOf p2
- p rdf:type owl:TransitiveProperty
- p rdf:type owl:SymmetricProperty
- p rdfType owl:FunctionalProperty
- p rdf:type owl:InverseFunctionalProperty

OWL Lite properties 3

- You can declare certain restrictions
- p owl:allValuesFrom c — ∀ restriction
- p owl:someValuesFrom c — ∃ restriction
- p owl:minCardinality n (Lite: n is 0 or 1)
- p owl:maxCardinality n (can be 0)
- p owl:cardinality n

OWL Lite individuals

- Individuals are members of classes.
- x1 owl:sameAs x2 — equality
- x1 owl:differentFrom x2 — inequality
- owl:allDifferent
- owl:distinctMembers

OWL DL (hi, ¬)

- Can define a class by enumerating individuals using owl:oneOf
- Can say class do not overlap using owl:disjointWith *(sic.)*
- Arbitrary Boolean combinations of classes using owl:unionOf, owl:complementOf, owl:intersectionOf

Ignoring OWL Full

- A study in 2006 looked at 1275 ontologies on the Web.
- 924 were in OWL Full, not DL or Lite.
- Adding missing type facts left just 61 in Full

Importing, versioning

- An OWL ontology can import another
- OWL ontologies can be versioned using owl:versionInfo, owl:priorVersion and owl:backwardsCompatibleWith
- You can deprecate with owl:DeprecatedClass and owl:DeprecatedProperty.

CWA & UNA

- We saw last week that SPARQL does seem to depend on the Closed World Assumption (otherwise it couldn't do NOT) and the Unique Name Assumption (or it couldn't do ≠)
- But OWL is an honest-to-goodness description logic and does NOT make these assumptions.

Pellet: an OWL engine

- Open source, Java.
- Sound and complete reasoner for OWL DL
- 1.5 work years by people with no prior description logic knowledge and some but not much knowledge of theorem proving.
- Claimed to be good performance.

What Pellet can do

- Check the *consistency* of an ontology
- *Concept satisfiability*
- *Classification* (complete the class hierarchy)
- *Realisation* (find most specific type of indiv.)
- *Entailment* checking (of classes/concepts)
- SPARQL and other query languages