Ontology reasoning in Python

PyPy project
Anders Lehmann, DFKI june 2006

Main Ideas

- Enable reasoning on OWL from Python
- Map OWL triples to a constraint program
- Use the constraint solver in PyPy
  - Will work on many platforms
  - Will work in many environments
- Use the PyPy JIT-compiler to create fast programs

Constraint Programming

- Variables
- Domains
  - A set of possible values for a Variable
- Constraints
  - Rules that can be used to narrow the domains

```
    x = [1, 2, 3, 4]
    y = [1, 2, 3, 4]

    constraint = "x + y > 6"
Result
    x = [3, 4]
    y = [3, 4]
```

Mapping OWL triples to constraints

- Classes maps to domains
  - (:car rdf:type owl:Class) -> car = ClassDomain()
- Individuals maps to elements in domains
  - (:Audi rdf:type :car) ->
    - Audi = Individual(); car.add(Audi)
- Properties maps to special domains
- The builtin predicates maps to constraints
  - We have to build the constraint from the triples

Mapping OWL triples to constraints

- Some constraints checks assertions
- Special domains needed for reason about schemas
-

Example

_:a rdf:type owl:Restriction .
_:a owl:maxCardinality "2"^^xsd:nonNegativeInteger .
_:a owl:onProperty first:prop .
first:sb1 rdf:type _:a .
first:sb1 first:prop first:ob1 .
first:sb1 first:prop first:ob2 .
first:sb1 first:prop first:ob3 .
first:ob1 owl:differentFrom first:ob2 .
first:ob1 owl:differentFrom first:ob3 .
first:ob2 owl:differentFrom first:ob3 .
first:prop rdf:type owl:ObjectProperty .

prop = ((sb1,ob1), (sb1,ob2), (sb1,ob3))
cons = “ len(prop_for_sb1) <= 2” -> ConsistencyFailure

Results

- The target is OWL-DL
  - OWL-Full is possible, but not useful ?
- Reasoning about Individuals
- Reasoning about schema
  - Not finished – will be in june
- A Logic Object Space in PyPy
  - Will be finished in june

Future Work

- SPARQL
  - Query language – due in August
- Python to OWL
  - Using the PyPy translation tool to produce OWL
  - Not a part of the EU project
- OWL to Python
  - Not a part of the EU project
- CRUD for OWL in Python
  - Either through SPARQL or the above