

Representing Prolog Knowledge as Ontology

This is an example about how to represent ontological information in Prolog. The example uses rules similar to the one for the health insurance example.

Classes are Unary Predicates

Classes can be represented on Prolog as unary predicates. For example:

```
person(mary)
```

can be used to represent that mary is a person. A representation as an ontology triple could be

```
uw:mary rdf:type uw:person
```

(uw is the used as namespace)

Be aware that in Prolog there is no schema. In RDFS one could represent explicitly that uw:person is a class using

```
uw:person rdf:type rdfs:Class
```

Properties are binary relations

Properties can be represented on Prolog as binary predicates. For example:

```
age(mary, 60) .
```

```
residence(mary, italy) .
```

can be used to represent that mary has age 60 and has residence Italy, which has the following representation as triples:

```
uw:mary uw:age "60"
```

```
uw:mary uw:residence uw:italy
```

Age is a data property and residence is an objective property. The following triples represents the schema information that age and residence are properties.

```
uw:residence rdf:type rdf:Property
```

```
uw:age rdf:type rdf:Property
```

Rules

Rules in Prolog and in SWRL look very similar. In Prolog the head is on the left while in SWRL the head is on the right

```
Prolog: risk(P,high) :- disease(Person, heartdisease)
```

SWRL: `uw:disease(?p, uw:heartdisease) -> uw:risk(?p, "high")`

Example Knowledge Base

Here you see a Prolog knowledge base and its correspondence in RDFS.

Prolog

```
person(mary) .  
age(mary, 60) .  
residence(mary, italy) .  
disease(mary, heartdisease)  
  
risk(P, high) :- disease(Person, heartdisease)
```

RDFS

```
PREFIX uw: <http://knut.hinkelmann.ch/underwriting#>
```

Facts

```
uw:mary rdf:type uw:person  
uw:mary uw:age "60"  
uw:mary uw:residence uw:italy  
uw:mary uw:disease uw:heartdisease
```

Schema

```
uw:heartdisease rdf:type uw:disease  
uw:italy rdf:type uw:country
```

Declaring the classes and Properties

```
uw:person rdf:type rdfs:Class  
uw:country rdf:type rdfs:Class  
uw:residence rdf:type rdf:Property  
uw:age rdf:type rdf:Property  
uw:risk rdf:type rdf:Property  
uw:disease rdf:type rdf:Property
```

