the inverse property: :isRentedBy has :Person as range and :Apartment as domain. In OWL2 DL, only object properties can have an inverse.

Equivalent Properties Properties can also be defined as equivalent. That is, every two individuals related via a property will always be related via its equivalent, and vice versa. Equivalence is a convenient mechanism for *mapping* elements of different ontologies to each other. For instance:

```
:isPartOf  rdf:type                owl:ObjectProperty ;
           owl:equivalentProperty dbpedia:partOf .
```

Disjoint Properties For some properties we know that no two individuals related via one property can be related via the other: the sets of pairs of individuals for which the properties can hold are *disjoint*. Examples are the :rents and :owns properties:

```
:rents rdf:type            owl:ObjectProperty ;
       rdfs:domain          :Person ;
       rdfs:range           :Apartment ;
       owl:disjointProperty :owns .
```

Clearly, you cannot rent something you own. Note that under the direct semantics of OWL2 DL, the owl:ObjectProperty and owl:DatatypeProperty are disjoint as well.

Property Chains A more complex feature of OWL2 is the ability to define *chains* of properties. Sometimes it is useful to specify shortcuts along the graph of properties relating various individuals. For instance, if we know that :Paul :rents the :BaronWayApartment, and that the :BaronWayApartment :isPartOf the :BaronWayBuilding, for which the dbpedia:location is dbpedia:Amsterdam, we know that :Paul must have a :livesIn relation with :Amsterdam. In OWL2 we can specify this using a property chain axiom: