

we know that this will never be the case. For instance, the `:isCheaperThan` relation is *asymmetric*, since nobody can be defeated by the person they defeated.⁶

```
:isAdjacentTo rdf:type owl:ObjectProperty ;
              rdf:type owl:SymmetricProperty .
```

```
:isCheaperThan rdf:type owl:ObjectProperty ;
               rdf:type owl:AsymmetricProperty ;
               rdf:type owl:TransitiveProperty .
```

Functional and Inverse-Functional Properties For some properties we know that every individual can always have at most one other individual related via that property. For instance, `:hasNumberOfRooms` is a *functional* property, and the `:hasRoom` property is *inverse-functional*:

```
:hasNumberOfRooms rdf:type owl:DatatypeProperty ;
                  rdf:type owl:FunctionalProperty .
```

```
:hasRoom          rdf:type owl:ObjectProperty ;
                  rdf:type owl:InverseFunctionalProperty .
```

Note that if two apartments a_1 and a_2 are related via `:hasRoom` to the same room r , this is not necessarily inconsistent: the individuals will simply be inferred to be the *same*.

Reflexive and Irreflexive Properties Reflexivity of a property means that every individual is related via that property to itself. For instance, everything `:isPartOf` itself. Irreflexivity, on the other hand, means that no individual is related to itself via that property. Most properties with disjoint domain and range are actually irreflexive. An example is the `:rents` property:

⁶Of course, this is only the case if every pair of persons is only allowed to compete in a single match.