

**Cardinality Restrictions** A cardinality restriction constrains the number of values a certain property may have for a class. If we additionally specify the class these values need to belong to, the restriction is said to be *qualified*. For example:

:StudioApartment

```

    rdf:type          owl:Class;
    rdfs:subClassOf [ rdf:type          owl:Restriction;
                      owl:onProperty  :hasRoom ;
                      owl:cardinality "1"^^xsd:integer
                    ] .

```

This specifies that a studio apartment can have exactly one value for the :hasRoom property. We can turn this into a qualified cardinality restriction by stating that the cardinality holds for members of the :LivingRoom, :Kitchen, and :Bedroom classes only (studios do tend to have separate bathrooms):

:StudioApartment

```

    rdf:type          owl:Class;
    rdfs:subClassOf [ rdf:type          owl:Restriction;
                      owl:onProperty  :isPlayedBy ;
                      owl:qualifiedCardinality "1"^^xsd:integer ;
                      owl:onClass [ owl:unionOf (:LivingRoom :Kitchen :Bedroom) ]
                    ] .

```

Note that the qualified restriction still allows for the members of the restricted class to have additional values for the property, provided that these belong to the complement of the qualifier class. A qualified cardinality restriction on owl:Thing is equivalent to a restriction without qualifier. Table 4.2 summarizes the different cardinality restrictions allowed in OWL2.

**Data Range Restrictions and Datatypes** Universal and existential restrictions on datatype properties allow members of a class to have *any* value from the specified