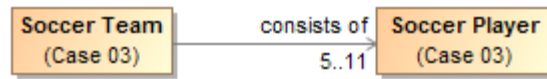


# Universal quantification constraint

Sometimes, in the context of some class, it is necessary to constrain *all* the values of a property to a particular type. This constraint is known as a *universal* quantification or *for-all* constraint (). This kind of constraint is an assertion that only values of the specified type are valid, and that the number of values must be between some minimum and maximum cardinality. In the concept modeling interpretation of UML, introducing a new property or redefining an existing property creates a universal quantification constraint in the context of the owning class. This interpretation is based on {redefines} in UML, which allows adding more specific constraints to an existing property without defining a new property.

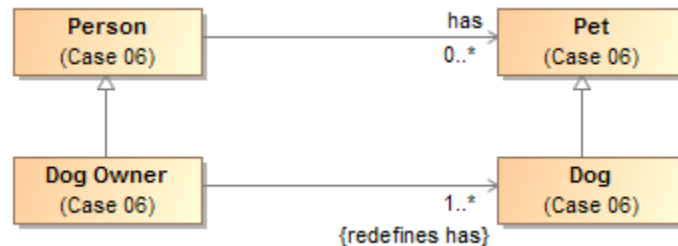
The diagram below shows the introduction of a new property, “consists of”, defining a universal quantification constraint on the property. The constraint states, in the context of Soccer Team and any of its subclasses, that all values of this property must be of the type “Soccer Player”, and that there must be between 5 and 11 values of this property.



The property 'consists of' defining a universal quantification constraint.

The following diagram shows a universal quantification constraint on the property “has” (owned by the class “Person”). It states, in the context of “Dog Owner”, that all values of the property “has” must be of type “Dog”, and that at least one value of this property must exist.

**Note**  
A property that is redefined must have the same name as the redefined property. In this case, leaving the redefined property unnamed is equivalent to naming the property the same as the one being redefined (“has” from the class “Person”).



A universal quantification constraint on the property 'has'.

**Optimized Multiplicity**  
UML multiplicity on a redefining property is now optimized for unqualified cardinality restrictions in OWL.

## Related pages

- [Concept Modeling Semantics](#)
- [Property restrictions](#)