4.2.3 Expressivity

Unfortunately, the *expressive power* of RDF and RDFS is very limited in some areas. We often need to provide more precise definitions than what RDF and RDFS allow us to state. If we build ontologies, we may want to be able to reason about:

Class Membership We have seen that RDFS has some simple mechanisms for determining class membership of individual instances using subclass and domain and ranges. However, a more precise description of the conditions under which an instance can be considered to belong to a class would allow for more fine-grained reasoning. For instance, if we have declared that certain property-value pairs are a sufficient condition for membership in a class :A, then if an instance :x satisfies these conditions, we can conclude that :x must be an instance of :A: something is only a tennis match if it involves at least players, rackets, etc.

Classification Similarly, we would like to use the conditions on class membership to infer relations between the classes themselves. For instance, a simple definition of a tennis match can be reused to define badminton matches.

Equivalence and Equality It can be very useful to express *equivalence* between classes. For example, the class :Tortoise shares all its members with the class :Land_Turtle; they are therefore equivalent. Similarly, we would like to be able to state when two instances are the same: the :morning_star and the :evening_star are names for the same planet :venus; these instances are therefore the same. Again, being able to express this directly is nice, but it should also be possible to determine equivalence and equality by applying formal semantics to the description of our classes.

Disjointness and Difference Analogously, sometimes we know that two classes do not share any instances (they are *disjoint*) or that two instances are decidedly not