

in OWL2 Full, we could impose a cardinality constraint on the class of all classes, essentially limiting the number of classes that can be described in any ontology.

The advantage of OWL2 Full is that it is mapped to an *RDF-based semantics*. It is therefore both structurally and semantically fully upward-compatible with RDF: any legal RDF document is also a legal OWL2 Full document, and any valid RDF Schema inference is also a valid OWL2 Full conclusion. The disadvantage of OWL2 Full is that the language has become so powerful as to be undecidable, dashing any hope of complete (or efficient) reasoning support.

#### 4.3.1.2 OWL2 DL: Direct Semantics

In order to regain computational efficiency, the second sublanguage OWL2 DL is mapped onto a description logic (DL). Description logics are a subset of predicate logic for which efficient reasoning support is possible. OWL2 DL restricts the way in which the primitives of OWL2, RDF, and RDFS may be used. Some of these restrictions are:

- OWL2 DL does not allow the application of OWL2's primitives to each other.
- Secondly, OWL2 DL can only define classes of non-literal resources. All OWL2 DL classes are instances of `owl:Class` rather than `rdfs:Class`.
- Thirdly, OWL2 DL strictly separates properties for which the range includes non-literal resources from those that relate to literal values. All OWL2 DL properties are instances of either `owl:ObjectProperty` or `owl:DatatypeProperty` but not both.
- Finally, in OWL2 DL a resource cannot be a class, property, or instance at the same time. They may share the same name (this is called “punning”) but will always be treated as distinct things by the underlying logic.

The above restrictions ensure that the language maintains its direct correspondence to a well-understood description logic. Figure 4.1 shows the subclass relationships