for example, rdfs:subClassOf. The namespace specifies only that it applies to classes and has a class as a value. The meaning of being a subclass, namely, that all instances of one class are also instances of its superclass, is not expressed anywhere. In fact, it cannot be expressed in an RDF document. If it could, there would be no need for defining RDF Schema.

We provide a formal semantics in the next section. Of course, RDF parsers and other software tools for RDF (including query processors) must be aware of the full semantics.

2.7 An Axiomatic Semantics for RDF and RDF Schema

In this section we formalize the meaning of the modeling primitives of RDF and RDF Schema. Thus we capture the *semantics* of RDF and RDFS.

The formal language we use is *predicate logic*, universally accepted as the foundation of all (symbolic) knowledge representation. Formulas used in this formalization are referred to as *axioms*.

By describing the semantics of RDF and RDFS in a formal language like logic we make the semantics unambiguous and machine-accessible. Also, we provide a basis for reasoning support by automated reasoners manipulating logical formulas.

2.7.1 The Approach

All language primitives in RDF and RDF Schema are represented by constants: $Resource, Class, Property, subClassOf$, and so on. A few predefined predicates are used as a foundation for expressing relationships between the constants.

An auxiliary theory of lists is used. It has function symbols

nil (empty list),

$cons(x, l)$ (adds an element to the front of the list),