Background on Knowledge Representation and Reasoning (KRR)

1 Introduction

Knowledge Representation and Reasoning (KRR) is a key area in artificial intelligence (AI) that focuses on how knowledge can be represented in a structured way that allows machines to interpret, reason about, and draw conclusions from it. The goal is to convert raw data into actionable knowledge through formal methods and reasoning techniques.

2 Resource Description Framework (RDF)

The **Resource Description Framework (RDF)** is a framework used to represent information about resources on the web. RDF structures information in the form of triples, making it machine-readable.

2.1 Basic Concepts of RDF

An RDF statement consists of three components:

- **Subject**: The resource being described.
- **Predicate**: A property of the subject.
- **Object**: The value of that property [1].

Each RDF statement is structured as a triple: (subject, predicate, object).

Example of RDF Triple

In Turtle syntax, a simple RDF triple might look like this:

```
@prefix ex: <http://example.org/> .

ex:Earth ex:isA ex:Planet .
```

This statement asserts that "Earth is a planet."

2.2 URIs and Literals

- **URIs (Uniform Resource Identifiers)** are used to uniquely identify resources, such as web pages, books, or concepts. They provide a way to reference resources globally. - **Literals** are data values that do not have separate identifiers, such as a string or number. For example, a person's name could be a literal value [1].

3 RDF Serialization Formats

RDF data can be serialized in various formats:

- **N-Triples**: A line-based format where each RDF triple is represented on a single line.
- **RDF/XML**: A verbose XML-based serialization.
- **Turtle (Terse RDF Triple Language)**: A human-readable RDF syntax that extends N-Triples.
- **JSON-LD**: A JSON-based format for linked data [1].

4 Blank Nodes and RDF Lists

4.1 Blank Nodes

A **blank node** represents a resource that does not need to be explicitly identified. It is useful for expressing the existence of an entity with specific attributes, without needing to provide a unique identifier for it [1].

4.2 RDF Lists

RDF also supports lists (collections) of resources, using containers or collections:

- **Containers**: Open lists that can be extended (e.g., a list of moons that can grow).
- **Collections**: Closed lists that cannot be extended (e.g., a fixed list of planets).

5 RDF Reification

Reification in RDF allows statements to be made about other statements, enabling the representation of metadata such as trustworthiness or reliability. For example, the statement:

> "Sherlock Holmes supposes that the gardener killed the butler."

This can be represented using RDF reification by creating a new triple that describes this assertion [1].

6 RDF Schema (RDFS)

RDF Schema (RDFS) extends RDF by providing more expressiveness and enables the definition of:

- **Classes**: Categories or types of resources (e.g., `Planet` is a class of celestial bodies).
- **Properties**: Relationships between resources (e.g., `satelliteOf` describes a relationship between two celestial bodies).
- **Domains and Ranges**: Constraints on properties. For example, `satelliteOf` only applies to celestial bodies [2].

6.1 Example of RDFS Class Definition

In RDFS, a class and its instances can be defined as follows:

```
@prefix ex: <http://example.org/> .

ex:Planet rdf:type rdfs:Class .
ex:Earth rdf:type ex:Planet .
```

This declares that `Planet` is a class, and `Earth` is an instance of that class (i.e., Earth is a planet) [2].

7 Hierarchical Relationships

RDFS supports hierarchical relationships through:

- **`rdfs:subClassOf`**: Defines subclasses and superclasses. For example:

  ```
  Planet rdfs:subClassOf CelestialBody
  ```

  This asserts that all planets are a subclass of celestial bodies [2].

- **`rdfs:subPropertyOf`**: Defines subproperties and superproperties. For example:

  ```
  artificialSatelliteOf rdfs:subPropertyOf satelliteOf
  ```

  This asserts that an artificial satellite is a specific type of satellite.

8 Logical Inference with RDF(S)

RDF(S) supports **logical inference**, allowing new facts to be derived from existing facts based on formal semantics. For instance, if we know that:

```
Pluto rdf:type Planet,
Planet rdfs:subClassOf CelestialBody
```

We can automatically infer that Pluto is a celestial body [2].

9 Linked Data

Linked Data is a method for interlinking structured data on the web. It uses URIs to uniquely identify resources and RDF to describe their relationships, enabling data from different sources to be merged and queried together. **SPARQL** is the query language used for retrieving and manipulating linked data [1].

10 Conclusion

Knowledge Representation and Reasoning (KRR) using RDF and RDFS enables machines to understand, reason about, and infer knowledge from structured data. RDF triples provide a simple yet powerful way to express relationships between entities, while RDFS enhances the expressiveness by introducing class hierarchies, property constraints, and logical inferences. These technologies form the foundation of the Semantic Web, allowing machines to process and reason about data automatically.

References

[1] Dr. Amna Basharat, *RDF - RDFS - Week 2 and 3*, KRR Fall 2023, September 2023.

[2] Dr. Amna Basharat, *RDF Schema - Week 2 and 3*, KRR Fall 2023, September 2023.