Resource Description Framework (RDF)

A basis for knowledge representation on the Web

- Simple language to capture assertions (as statements)
  - Captures elements of knowledge about a resource
  - Facilitates incremental acquisition of knowledge
  - Supports inferencing to extract and use knowledge
- Consolidates old KR ideas
  - Frames
  - Object-oriented modeling
- Applies URIs to
  - Clarify meanings
  - Handle vocabulary differences
  - Crucial for heterogeneity

Why RDF?

- Whereas XML and JSON
  - Produce a document tree
  - Don't identify the content represented by a document, i.e.,
    - Concepts the document is about
    - Relationships among the concepts
  - Enable multiple representations for the same content
- RDF expresses the content itself in a standard form

Resources and Literals

- RDF captures descriptions of resources
- A resource is an "addressable" object
  - Of which a description can be given
  - Identified via a URI
  - Worth talking about and possible to talk about
- A literal is something simpler
  - A value, e.g., string or integer
  - Cannot be given a description

Statements or Triples

- RDF is based on a simple grammar
  - An RDF document is simply a set of statements also known as triples
- Each statement consists of
  - Subject: a resource
  - Object: a resource or a literal
  - Predicate: a resource
- Comes with RDFS, a vocabulary to create vocabularies

Rendering RDF

- RDF is not about the surface syntax but about the underlying content
- Using the XML serialization of RDF
  - RDF is not tied to XML
  - Standard XML namespace syntax
  - Namespaces defined by the RDF standard
    - Typically abbreviated rdf and rdfs

Example of N-Triples Notation

The basic syntax: Subject-Predicate-Object

```
<http://www.wiley.com/SOC>
  <http://purl.org/dc/elements/1.1/title>
  "Service-Oriented Computing" .
<http://www.wiley.com/SOC>
  <http://purl.org/dc/elements/1.1/creator>
  "Munindar" .
<http://www.wiley.com/SOC>
  <http://purl.org/dc/elements/1.1/creator>
  "Michael" .
<http://www.wiley.com/SOC>
  <http://purl.org/dc/elements/1.1/publisher>
  "Wiley" .
```

Example in XML

Using the Dublin Core vocabulary

```
<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
   <rdf:Description rdf:about="http://www.wiley.com/SOC">
    <dc:title>Service-Oriented Computing</dc:title>
    <dc:creator>Munindar</dc:creator>
    <dc:creator>Michael</dc:creator>
    <dc:publisher>Wiley</dc:publisher>
   </rdf:Description>
</rdf:RDF>
```

- rdf:Description gathers statements about one subject
- Distinguish rdf:ID from rdf:about

Exercise

Reproduce previous example in JSON Linked Data syntax

Exercise

- Graphs represent binary relationships naturally
  - The vendor ships SKU-99
- Express a three-party relationship
  - The vendor ships SKU-99 quickly
  - Hint: think of gerunds from natural language grammar

Multiparty Relationships

- An edge has two terminals, so limited to binary relationships
- To represent a multiparty relationship, introduce a resource corresponding to the relationship itself
  - That's what a gerund does in NL
  - Analogous to an association entity
  - Include edges originating or targeting this resource

RDF Schema

In essence, an object-oriented type system built on top of RDF

- Defines
  rdfs:Class, rdfs:subClassOf, rdfs:Resource, rdfs:Literal,
  rdfs:Property, rdfs:subPropertyOf, rdfs:range, rdfs:domain,
  rdfs:label, rdfs:comment, rdfs:seeAlso
- Applications of RDF Schema
  - Definining custom vocabularies
  - Discussed in conjunction with OWL, which greatly enhances the above

RDF Schema versus XML Schema

Both help define custom vocabularies

- An XML Schema document gives us syntactic details
- An RDF Schema document gives us a way to capture part of the meaning through a standard vocabulary (rdfs)
- An OWL document (next topic) captures richer meaning

Collections

- Function as containers
  - rdf:Bag
  - rdf:Sequence
  - rdf:Alt (choice)
- Accompanied by properties to extract elements
  - Schematically represented as rdf:_1, rdf:_2, and so on
  - That is, the properties _1, _2, . . . are defined in the rdf namespace
- Collections are applied within OWL
  - Not otherwise emphasized in this course

Reification Motivation

- Express a quotation
  - Alice says the vendor ships SKU-99
- Hint(?): In RDF, we can only talk about resources
  - And literals, but literals are where a graph ends (no out edges)

Reification of Statements

- *Reify:* to make referenceable, essential for quoting statements to
  - Agree or disagree with them
  - Assert modalities: possible, desirable, ...
- Make a statement into a resource; then talk about it
  - rdf:Statement is a class
  - the given statement's rdf:type is rdf:Statement
  - rdf:Statement defines important properties: rdf:subject, rdf:object, and rdf:predicate

Reification Exercise

Produce a model using RDF and RDF Schema of the following assertions.

- (a) Statement (b) is false.
- (b) Statement (a) is true.

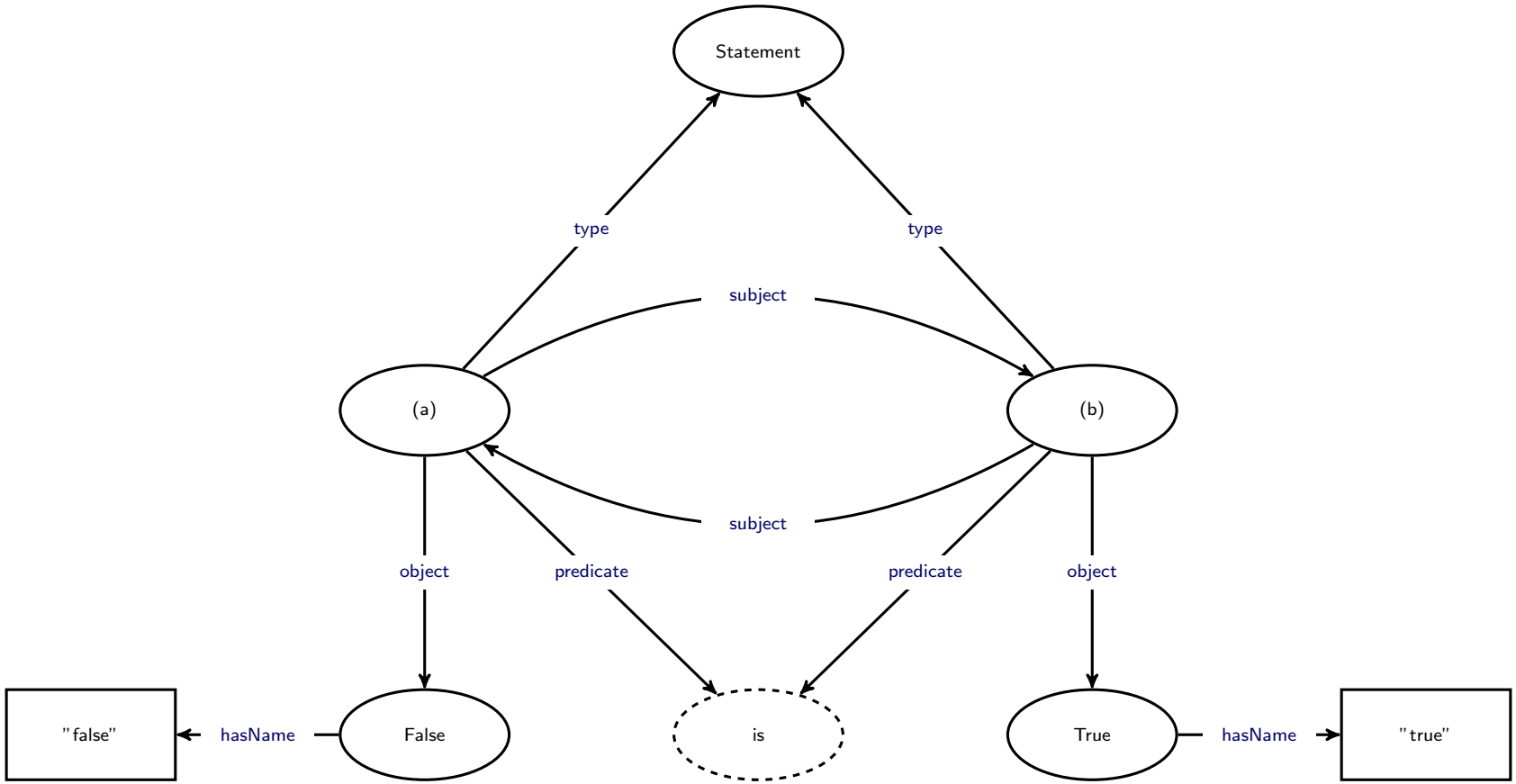
Express your solution as a graph with suitable annotations.

- Notation
  - Resources: solid ellipses
  - Properties (hence, also resources): dashed ellipses
  - Literals: rectangles
- Definitions
  - Two resources named ⌜true⌝ and ⌜false⌝
  - Property: ⌜is⌝

Reification Exercise Solution

Problem-specific constructs: (a), (b), True, False, hasName is

Generic: everything else



RDF Summary

- RDF captures deeper structure than XML
  - RDF captures graphs in general
  - Meaning depends on the graph, not the document that represents a graph
- RDF is based on an simple linguistic representation: subject, predicate, object
  - But webified via URIs
- RDF comes prepackaged with RDF Schema
  - In essence, an object-oriented type system: a vocabulary to create new vocabularies, such as
    - Friend of a Friend (FOAF)
    - Dublin Core
    - Mozilla extensions
  - Provides a basis for OWL (next topic)