Appendices

P.J. McBrien

Imperial College London

Appendix 1: RDF Data Model and OWL Schemas

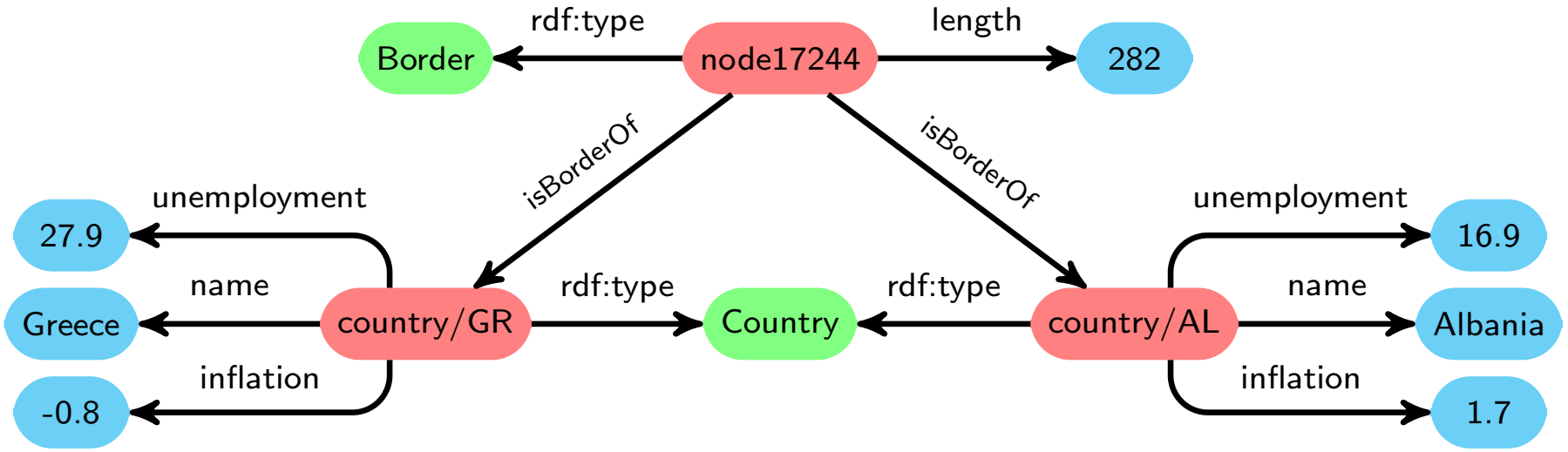
P.J. McBrien

Imperial College London

RDF Graph Model

Triples

Resource Description Framework (RDF):
data (nodes) are connected by **properties** (directed edges)
A **triple** is the combination of two data items linked by a property.



Turtle/N3 representation of RDF

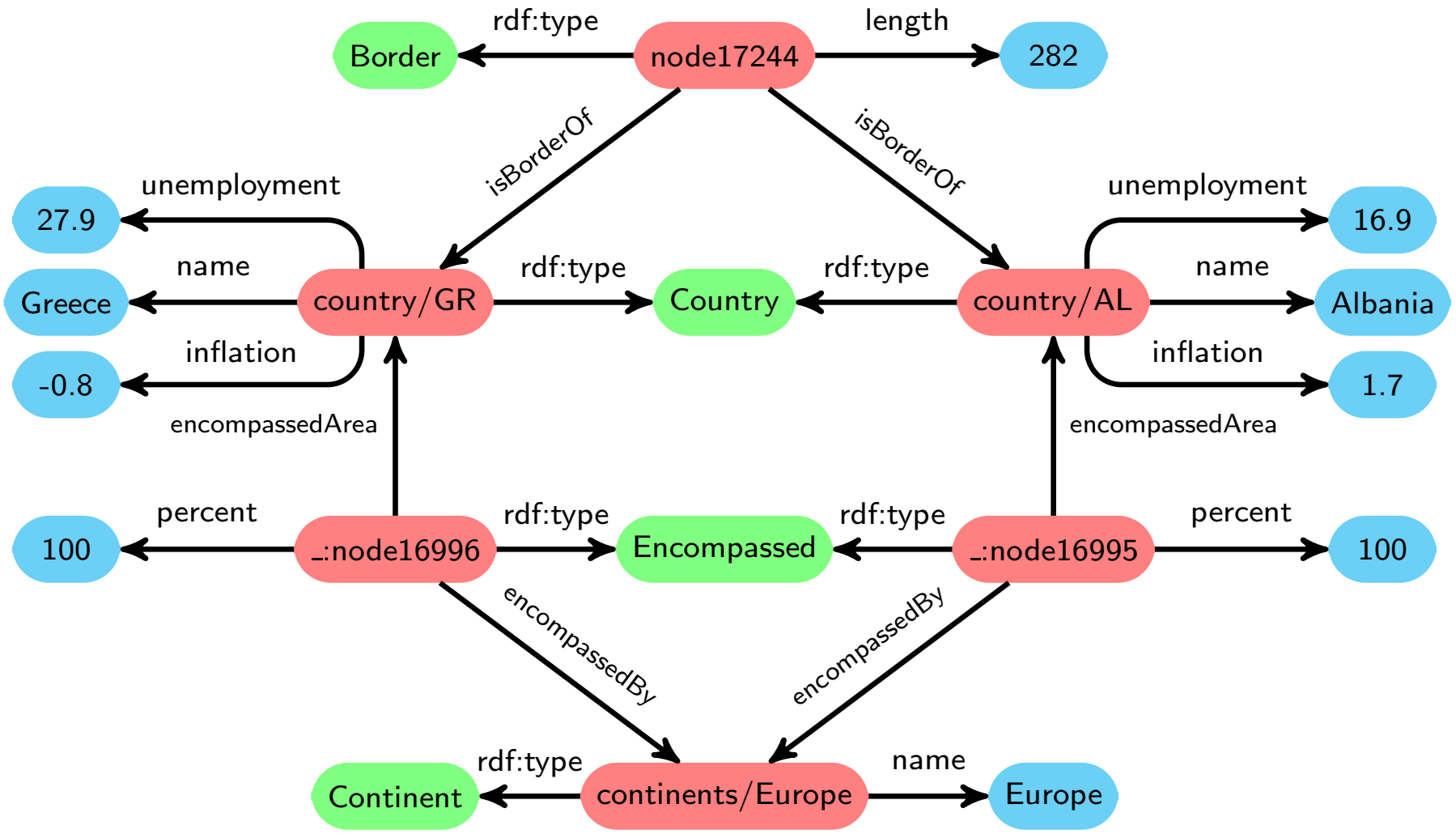
Listing properties of a node

```
<country/AL> rdf:type :Country ;
  :name "Albania" ;
  :inflation 1.7 ;
  :unemployment 16.9 .
```

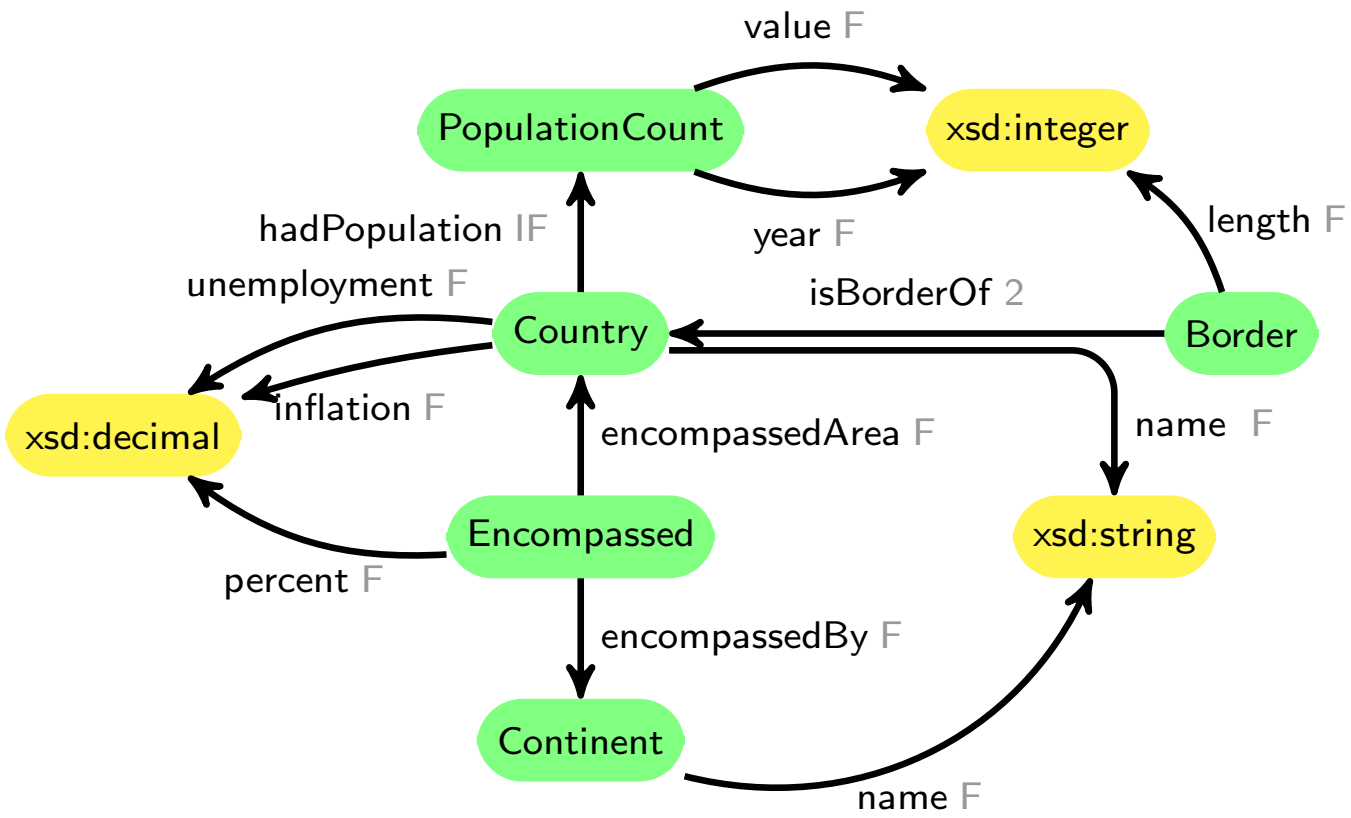
Listing properties of an anonymous node

```
[ rdf:type :Border ;
  :length 282 ;
  :isBorderOf <country/AL> ,
              <country/GR> ] .
```

Example RDF Graph Model



OWL Schemas



Web Ontology Language (OWL)

Allows more schema information to be specified

- Properties can be **functional**, **inverse functional**, or have **cardinality**
- Class properties link to classes
- Data properties link a class to a datatype.

Appendix 2: SPARQL Query Language for RDF Data

P.J. McBrien

Imperial College London

SPARQL

```
SELECT ?country ?inflation ?unemployment
WHERE {
  ?c :name ?country ;
     :inflation ?inflation ;
     :unemployment ?unemployment
}
ORDER BY ?country
```

SPARQL query

Basis of SPARQL query is a SELECT . . . WHERE statement

- WHERE contains a Turtle pattern; with variables prefixed by ?
- SELECT lists variables

Better to restrict the schema of the nodes matched using rdf : type

SPARQL

```
SELECT ?country ?inflation ?unemployment
WHERE {
  ?c rdf:type :Country;
     :name ?country ;
     :inflation ?inflation ;
     :unemployment ?unemployment
}
ORDER BY ?country
```

SPARQL query

Basis of SPARQL query is a SELECT . . . WHERE statement

- WHERE contains a Turtle pattern; with variables prefixed by ?
- SELECT lists variables

Better to restrict the schema of the nodes matched using rdf : type

SPARQL: Querying more than one class

```
SELECT  ?country ?continent ?pc
WHERE {
    ?e rdf:type :Encompassed ;
       :encompassedArea ?c ;
       :encompassedBy ?con ;
       :percent ?pc .
    ?c rdf:type :Country ;
       :name ?country .
    ?con rdf:type :Continent ;
       :name ?continent .
}
ORDER BY ?country ?continent
```

SPARQL: Use FILTER to filter down rows.

```
SELECT  ?country ?continent ?pc
WHERE {
    ?e rdf:type :Encompassed ;
       :encompassedArea ?c ;
       :encompassedBy ?con ;
       :percent ?pc .
    ?c rdf:type :Country ;
       :name ?country ;
       :area ?totalarea .
    ?con rdf:type :Continent ;
       :name ?continent .
    FILTER ( ?country = 'Turkey' )
}
ORDER BY ?country ?continent
```

SPARQL predicates to restrict rows returned should go in FILTER

The SELECT part of a SPARQL query cannot contain calculations.
Must create a new variable using BIND as part of WHERE

SPARQL: Use BIND to perform calculations.

```
SELECT  ?country ?continent ?pc ?area
WHERE {
    ?e rdf:type :Encompassed ;
       :encompassedArea ?c ;
       :encompassedBy ?con ;
       :percent ?pc .
    ?c rdf:type :Country ;
       :name ?country ;
       :area ?totalarea .
    ?con rdf:type :Continent ;
       :name ?continent .
    BIND ( ?pc * ?totalarea / 100.0 AS ?area )
}
ORDER BY ?country ?continent
```

SPARQL Calculations do not go in SELECT

The SELECT part of a SPARQL query cannot contain calculations.
Must create a new variable using BIND as part of WHERE

SPARQL: Care when properties are not functional

```
SELECT ?country ?length ?neighbour
WHERE {
  ?c rdf:type :Country;
     :name ?country ;
     :inflation ?inflation ;
     :unemployment ?unemployment .
  ?b rdf:type :Border;
     :isBorderOf ?c ;
     :isBorderOf ?n ;
     :length ?length .
  ?n rdf:type :Country;
     :name ?neighbour .
  FILTER (?c = ?n) .
}
ORDER BY ?country DESC(?length)
```