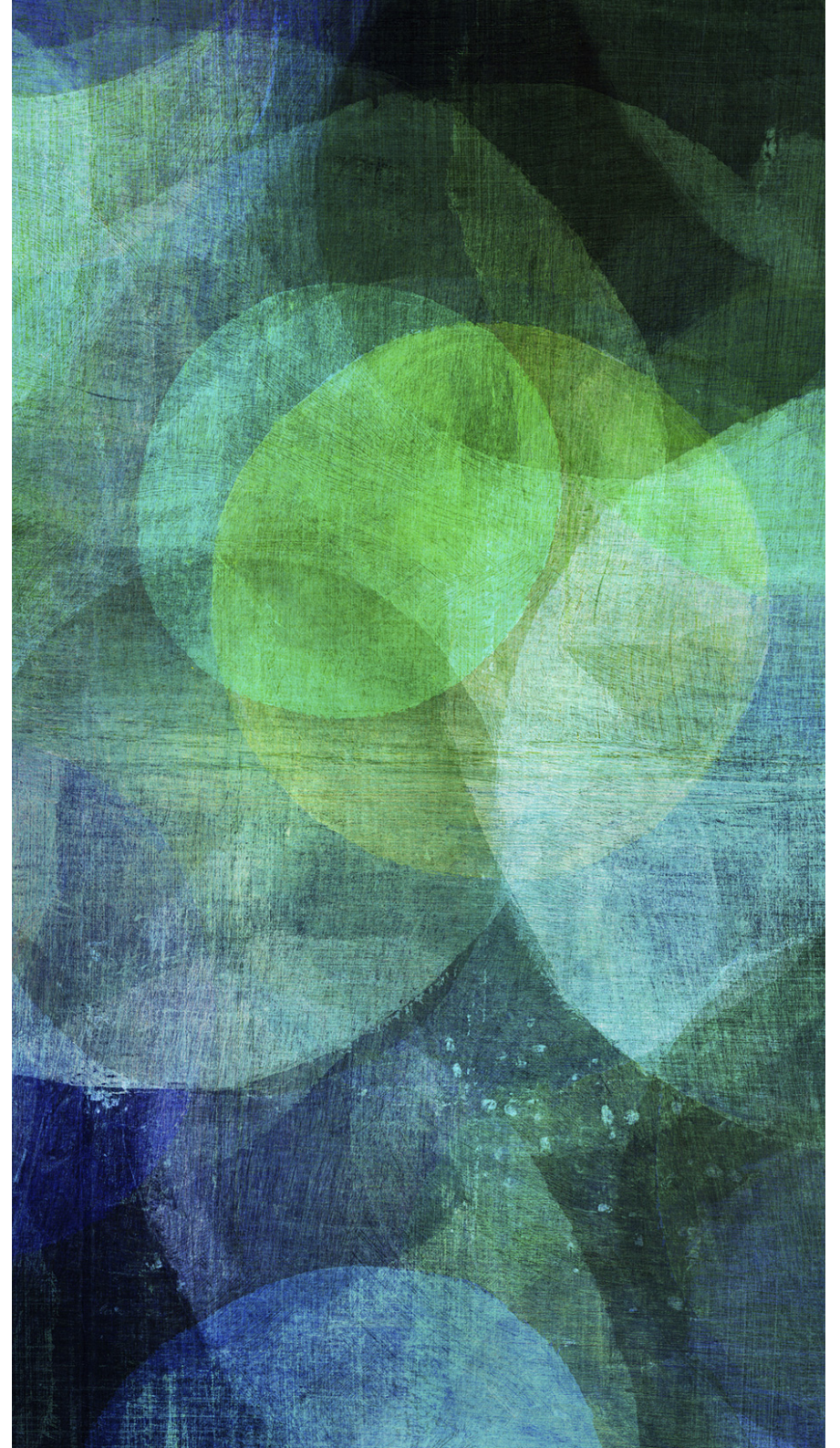


CONSULTAS SOBRE EL RDF

SPARQL



ACCESO A LOS DATOS RDF

- El enfoque HTTP/RESTful es muy bueno para construir aplicaciones simples
- Pero a medida que los datos crecen ,la complejidad en las relación también crece.
 - más posibilidades de consultas complejas son necesarias!
- ¿Cómo puedo hacer consultas sobre datos RDF?
 - por ejemplo, ¿Cómo puedo obtener información de DBpedia?

CONSULTANDO LOS GRAFOS RDF

- En la práctica se requieren realizar consultas un poco más complejas, por ejemplo:
 - “dame los pares de recursos (a,b) para los cuales existe un x tal que (x parent a) y (b brother x)” (esto retorna los tíos)
 - estas reglas pueden tornarse complejas
- Este es el objetivo de SPARQL(Query Language for RDF)

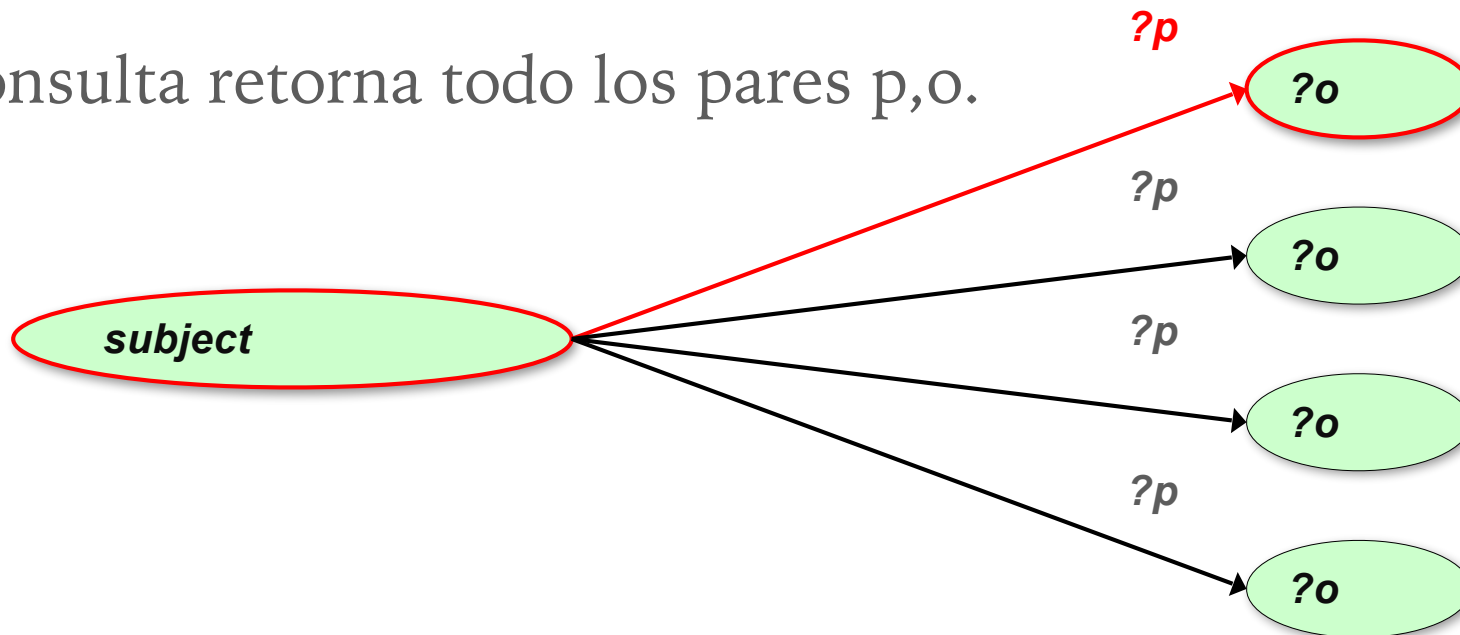
GENERAL: PATRONES QUE SE CUMPLEN EN EL GRAFO

- La idea fundamental es: usar patrones en el grafo
 - los patrones contienen símbolos que no están ligados
 - para ligar los símbolos, se seleccionan subgrafos RDF
 - si existe una selección posible, la consulta retorna los recursos que no están ligados

OUR PYTHON EXAMPLE IN SPARQL

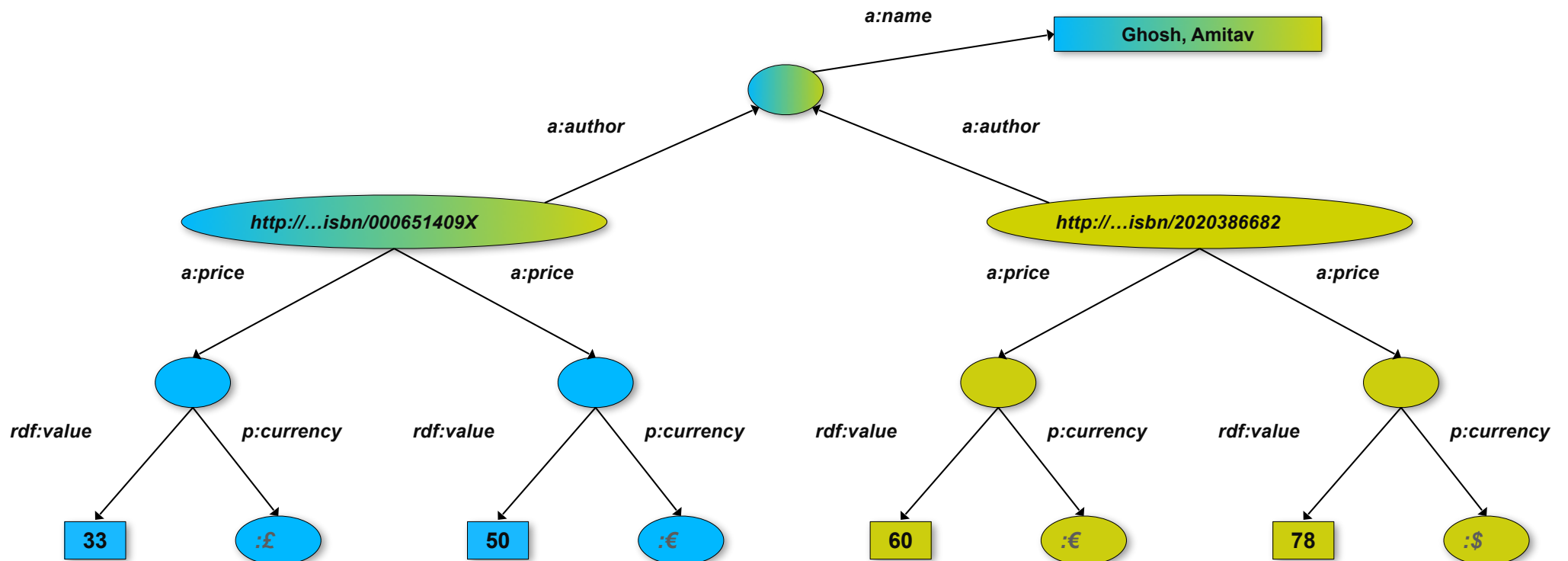
```
SELECT ?p ?o  
WHERE {subject ?p ?o}
```

- Las tripletas en la sentencia WHERE definen el patrón del grafo con *?p* y *?o* como símbolos no ligados.
- La consulta retorna todo los pares p,o.



EJEMPLO DE SPARQL

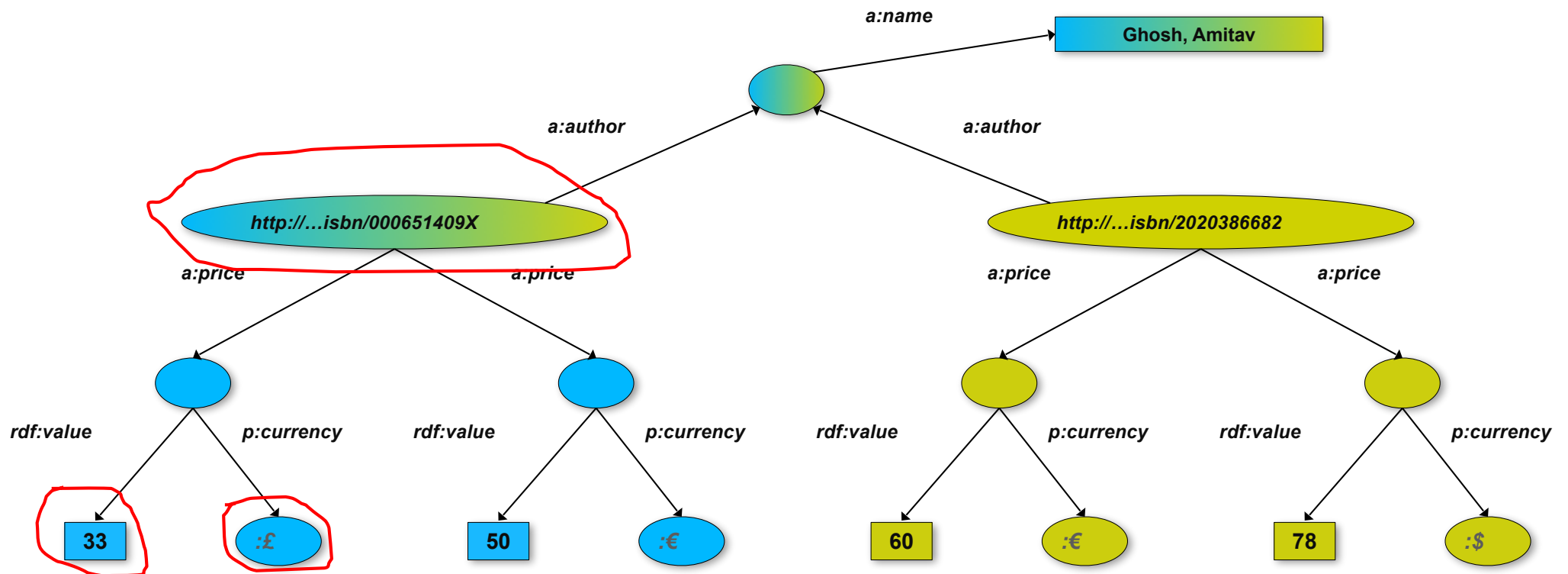
```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```



EJEMPLO DE SPARQL

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

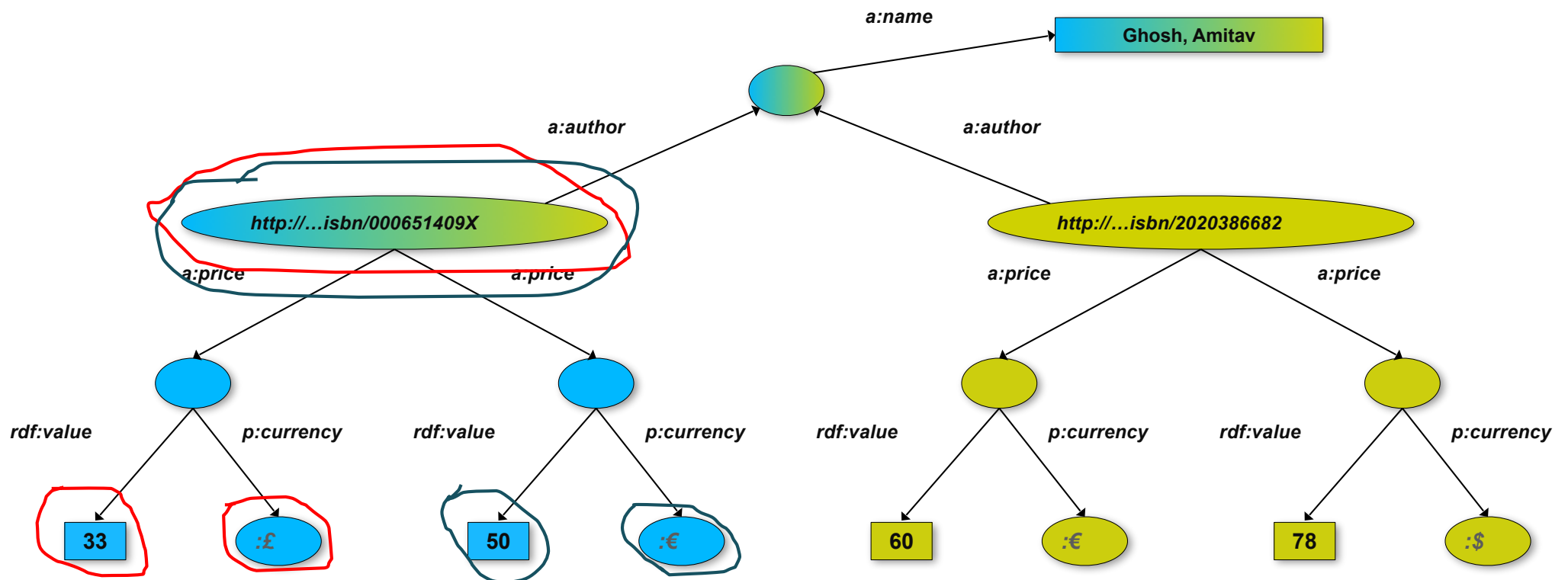
Retorna: [*<...409X>,33,:£*]



EJEMPLO DE SPARQL

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

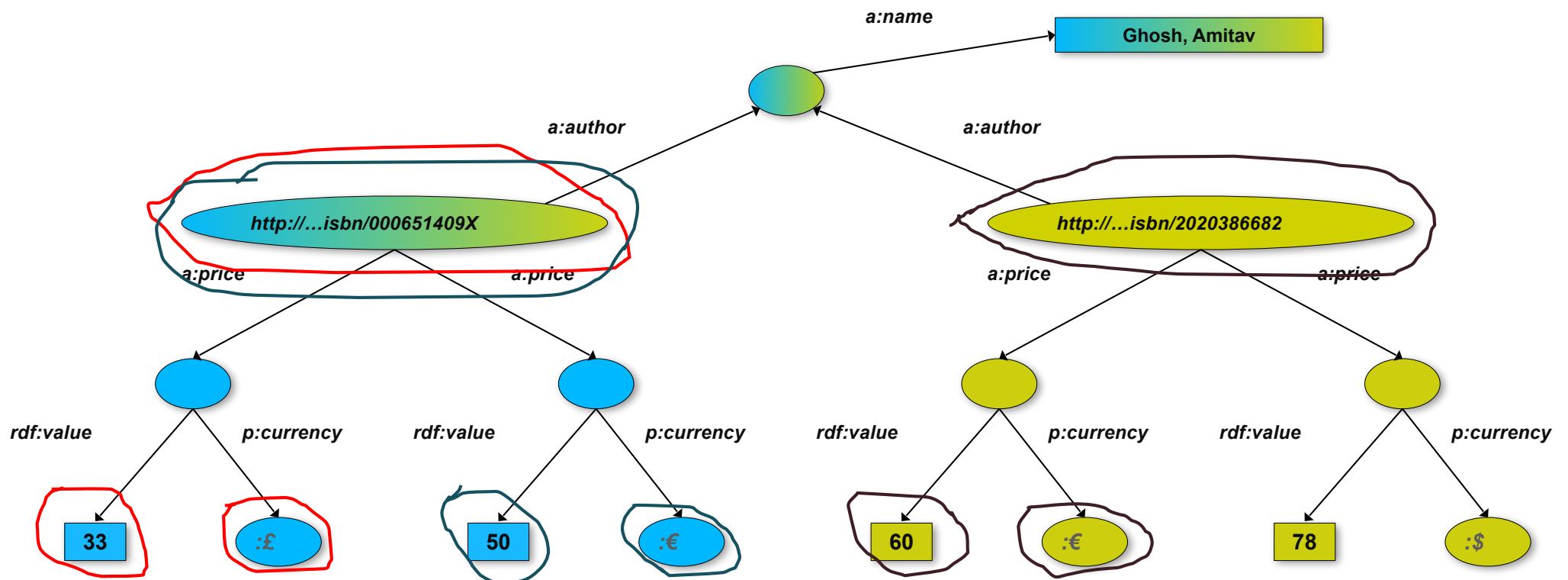
Retorna: [*<...409X>*,33,:£], [*<...409X>*,50,:€]



EJEMPLO DE SPARQL

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

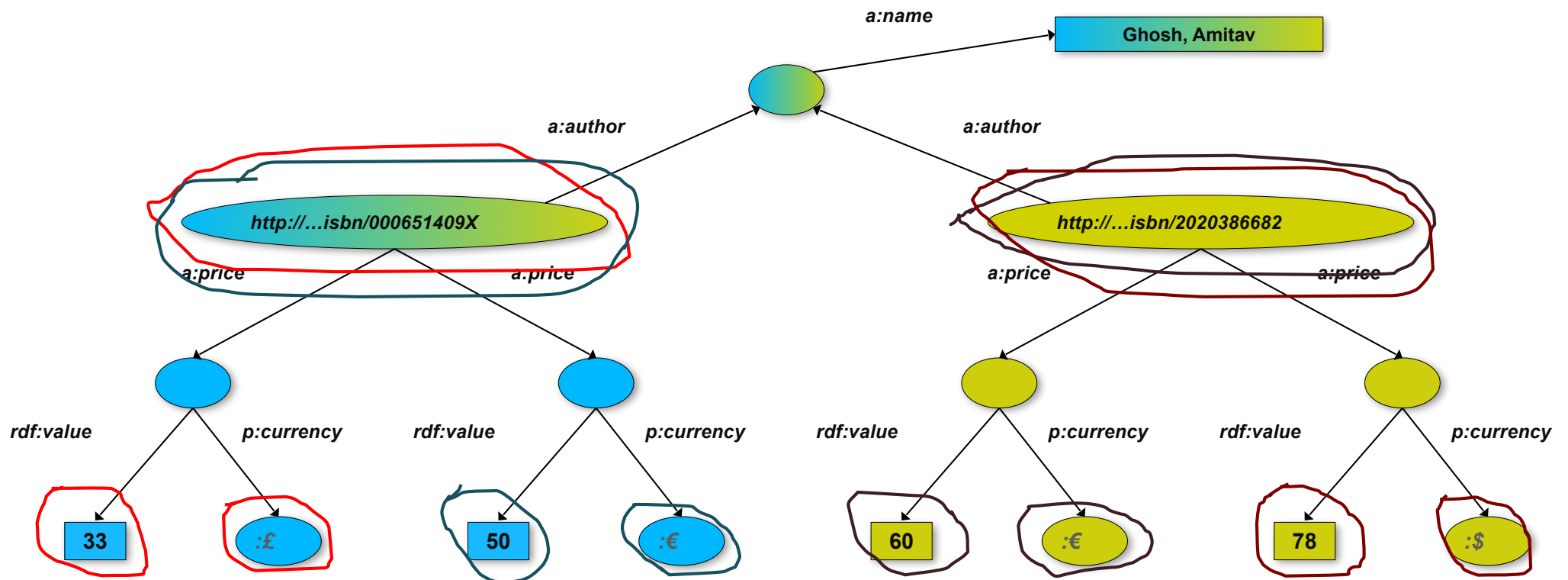
Retorna: [*<...409X>*,33,:£], [*<...409X>*,50,:€],
[<...6682>,60,:€]



EJEMPLO DE SPARQL

```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```

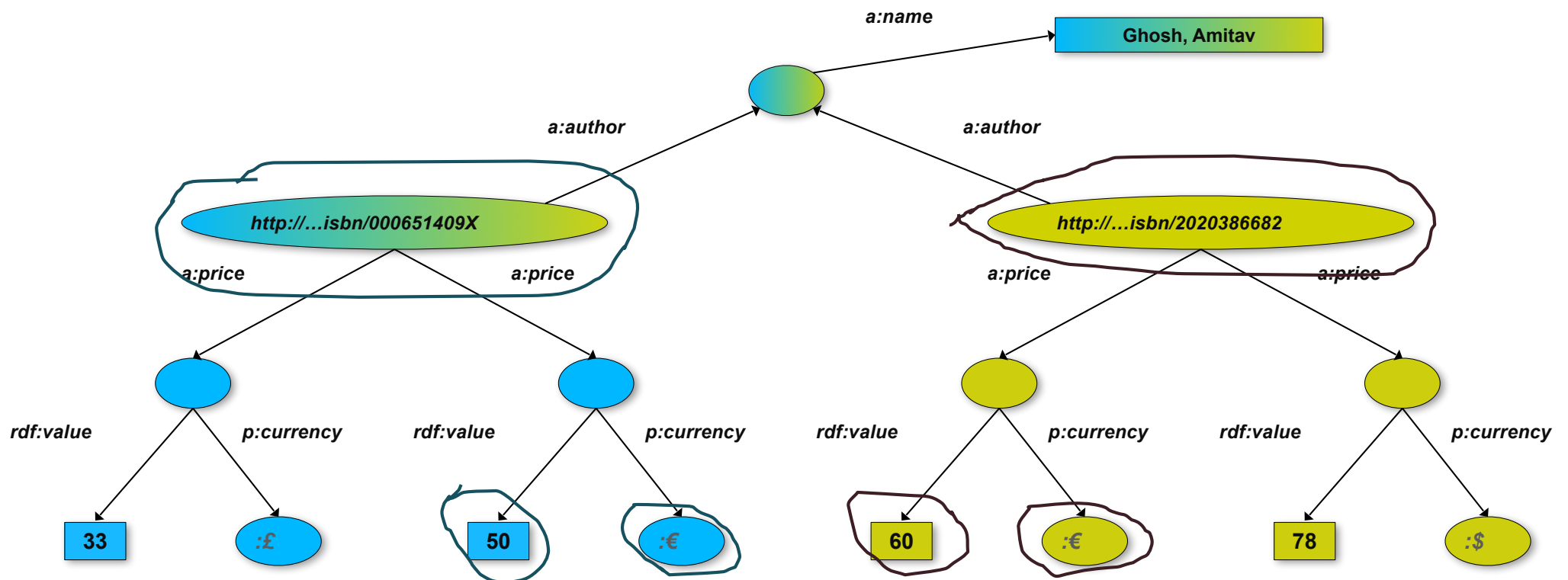
Retorna: [*<...409X>*,33,:£], [*<...409X>*,50,:€],
[*<...6682>*,60,:€], [*<...6682>*,78,:\$]



RESTRICCIONES EN EL PATRÓN

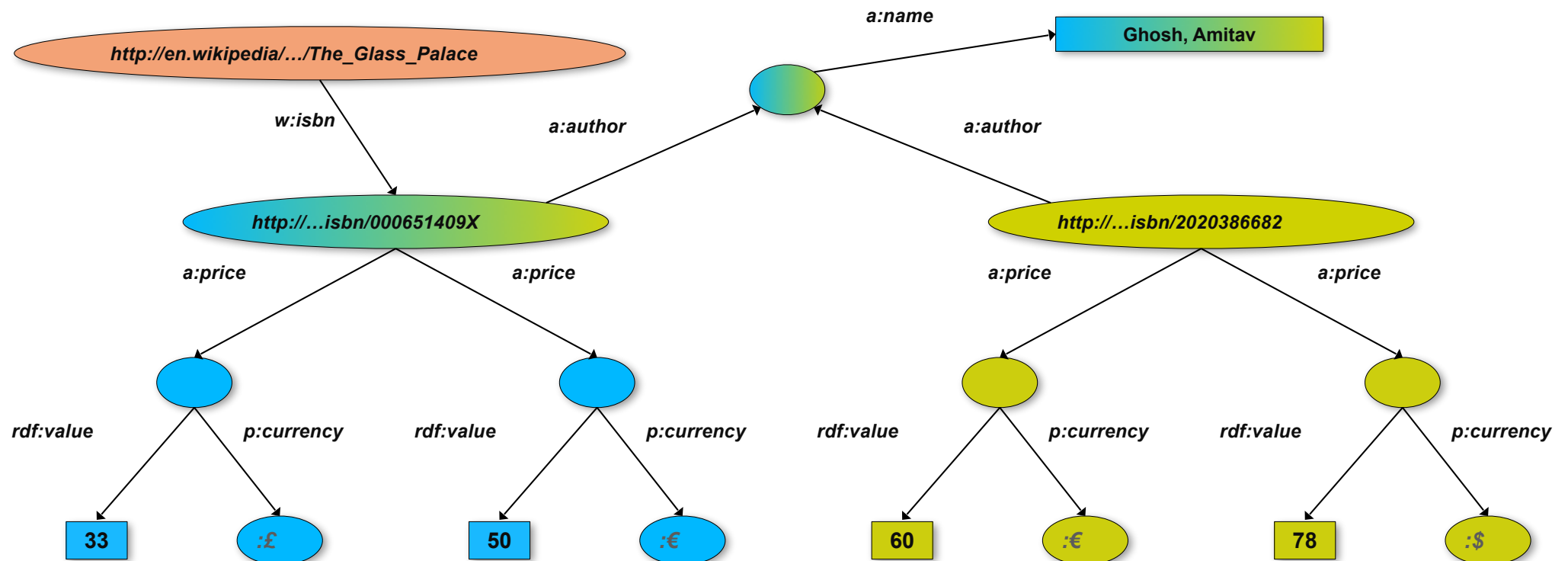
```
SELECT ?isbn ?price ?currency # note: not ?x!  
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.  
        FILTER(?currency == :€) }
```

Returns: [<...409X>,50,:€], [<...6682>,60,:€]



PATRONES OPCIONALES

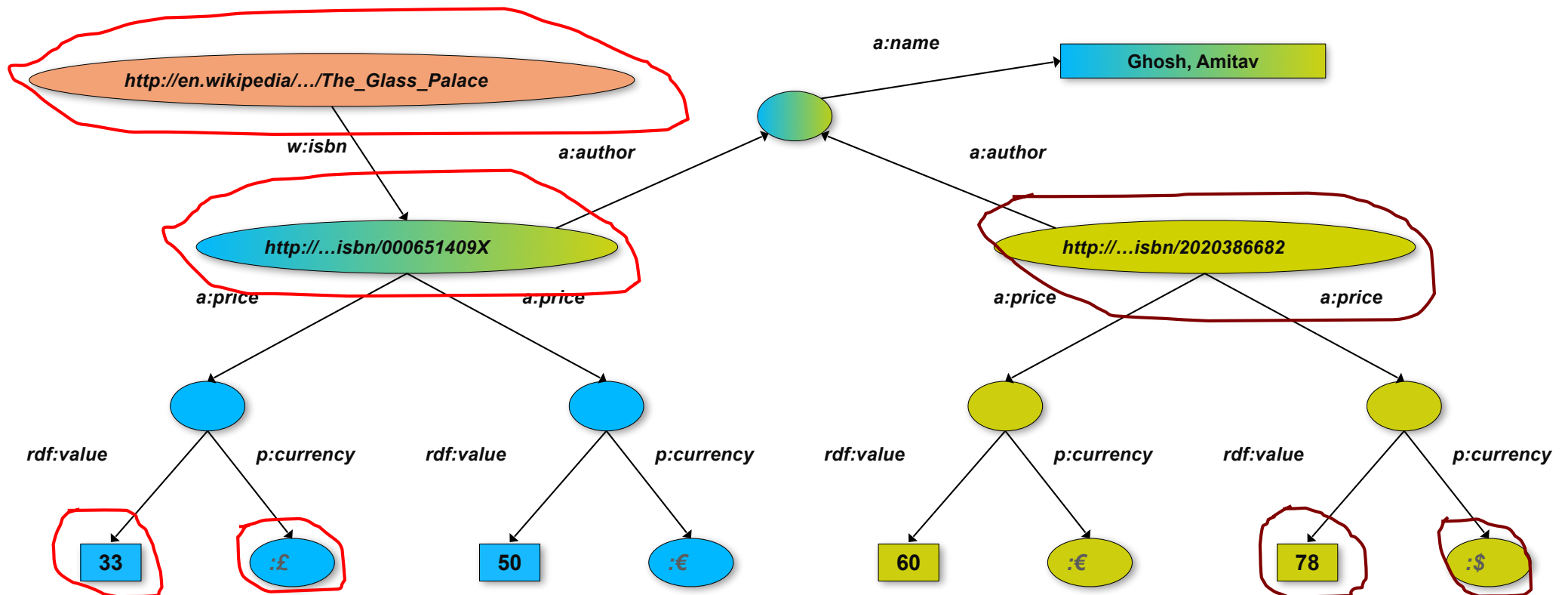
```
SELECT ?isbn ?price ?currency ?wiki
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.
        OPTIONAL ?wiki w:isbn ?isbn. }
```



PATRONES OPCIONALES

```
SELECT ?isbn ?price ?currency ?wiki
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.
        OPTIONAL ?wiki w:isbn ?isbn. }
```

Retorna: [[<..09X>,33,:£,<...Palace>], ... , [<..6682>,78,:\$,]]



CONSTRUIR UN GRAFO NUEVO

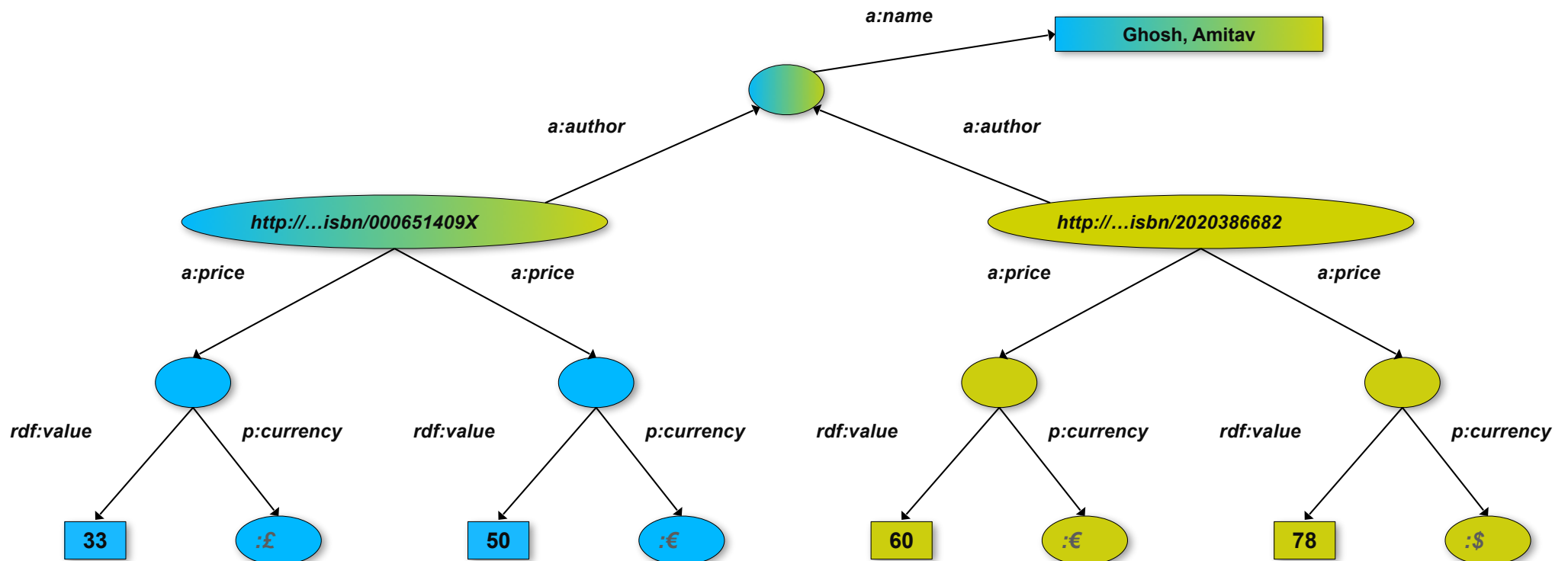
```
CONSTRUCT { ?isbn b:price ?price.
```

```
    ?isbn a:author ?y. ?y a:name ?name . }
```

```
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.
```

```
    ?isbn a:author ?y. ?y a:name ?name .
```

```
    FILTER(?currency == :€) }
```



CONSTRUIR UN GRAFO NUEVO

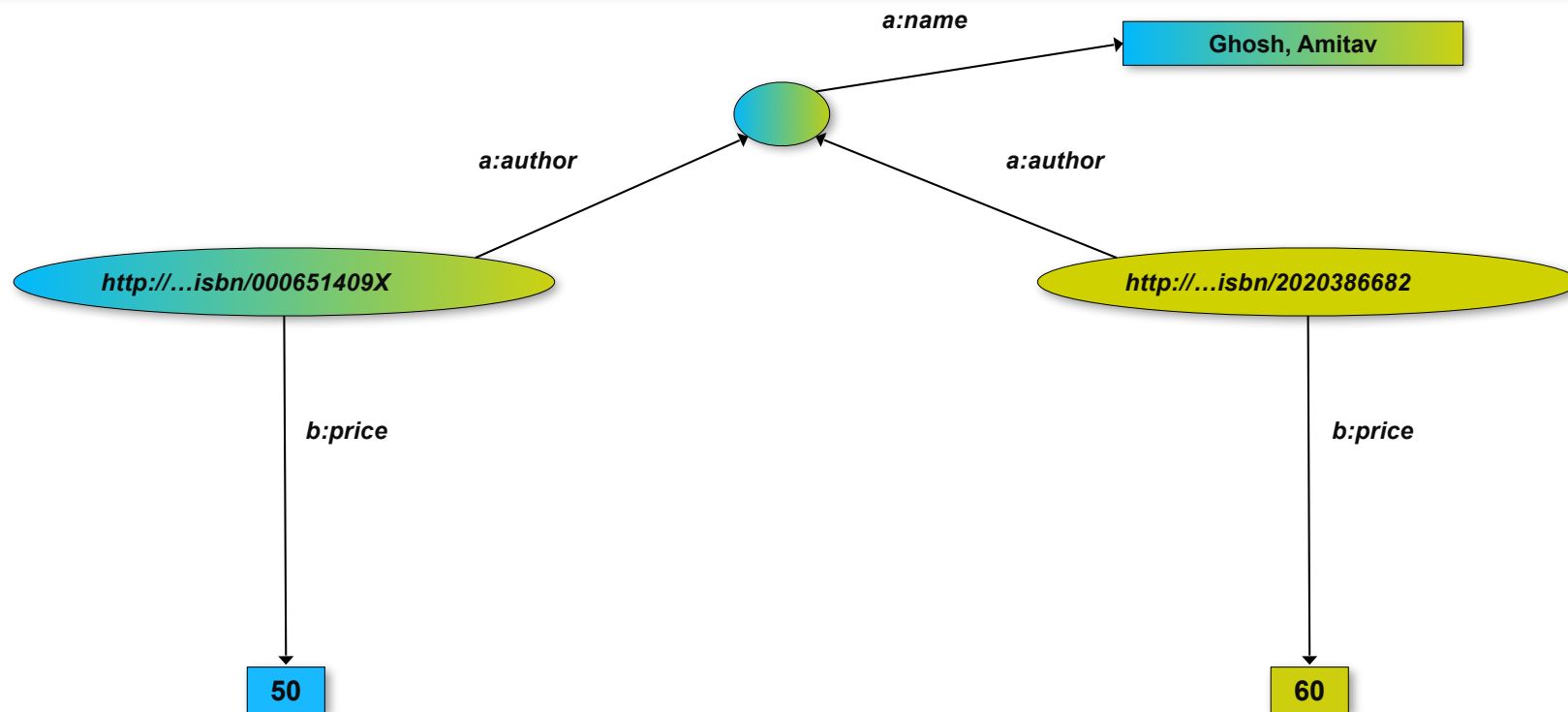
```
CONSTRUCT { ?isbn b:price ?price.
```

```
    ?isbn a:author ?y. ?y a:name ?name . }
```

```
WHERE { ?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.
```

```
    ?isbn a:author ?y. ?y a:name ?name .
```

```
    FILTER(?currency == :€) }
```



OTRAS FUNCIONALIDADES

- Limitar el numero de resultados, remover duplicados, ordenarlos, etc
- Especificar diferentes data sources (usando URI-s) dentro de la query
- Construir un grafo combinando un patrón separado y los resultados de la query
- Usar datatypes y tags del lenguaje cuando haya matching con un patrón.
- Agregación de los resultados (min, max, average, etc.)
- Path expressions (algo similar a regex)

SPARQL EN LA PRÁCTICA

- SPARQL usualmente se utiliza en la Web
 - un request http es enviado a una SPARQL endpoint
 - el resultado es el resultado del SELECT, CONSTRUCT, etc
- Documentos separados definen el protocolo y el formato de los resultados
 - Protocolo SPARQL Protocol para RDF con HTTP y/o bindings con SOAP
 - Los resultados del SPARQL pueden presentarse en varios formatos como XML o JSON
- Los datasets grandes por lo general ofrecen “SPARQL endpoints”

REMOTE QUERY/REPLY EXAMPLE

```
GET /qps?&query=SELECT+:+...+WHERE:+... HTTP/1.1
```

```
User-Agent: my-sparql-client/0.0
```

```
Host: my.example
```

```
HTTP/1.1 200 OK
```

```
Server: my-sparql-server/0.0
```

```
Content-Type: application/sparql-results+xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
```

```
  <head>
```

```
    <variable name="a"/>
```

```
    ...
```

```
  </head>
```

```
  <results>
```

```
    <result ordered="false" distinct="false">
```

```
      <binding name="a"><uri>http:...</uri></binding>
```

```
      ...
```

```
    </result>
```

```
    <result> ... </result>
```

```
  </results>
```

```
</sparql>
```

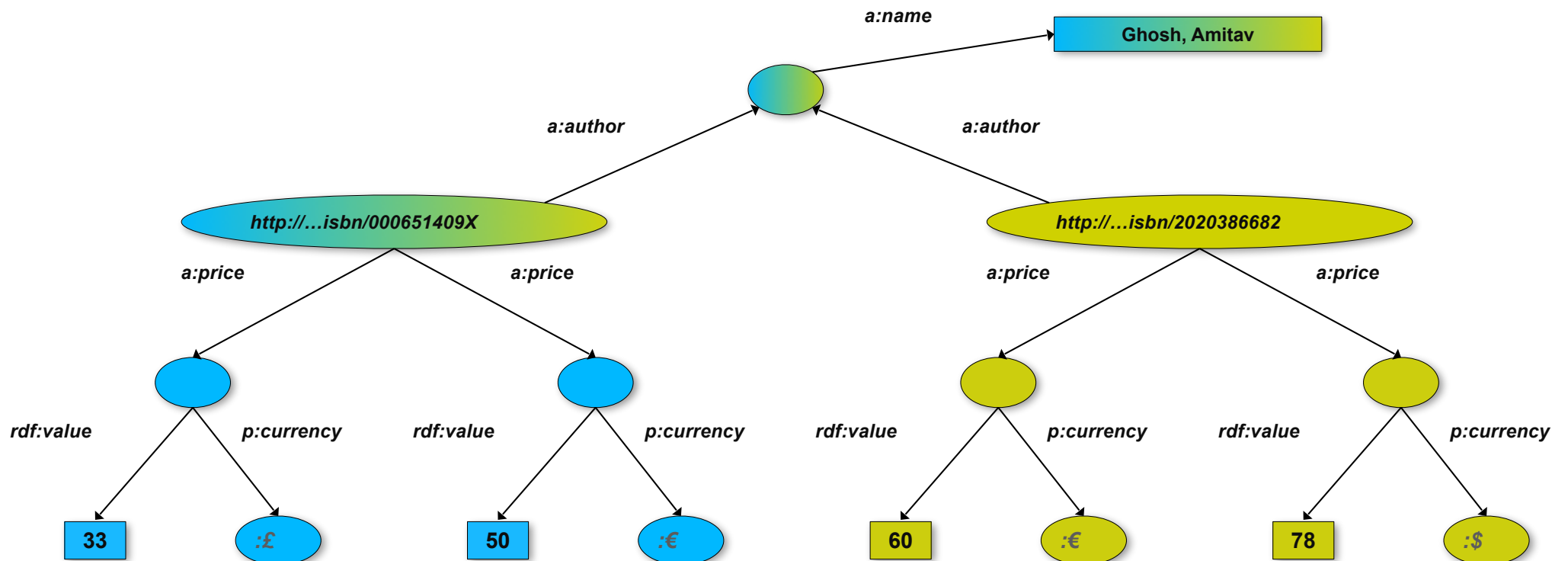
SPARQL 1.1 UPDATE

- SPARQL CONSTRUCT retorna un grafo nuevo
 - los datos originales no son modificados
- Update! para modificar el dataset original

UPDATE: INSERT

```
INSERT {?isbn rdf:type frbr:Work}
```

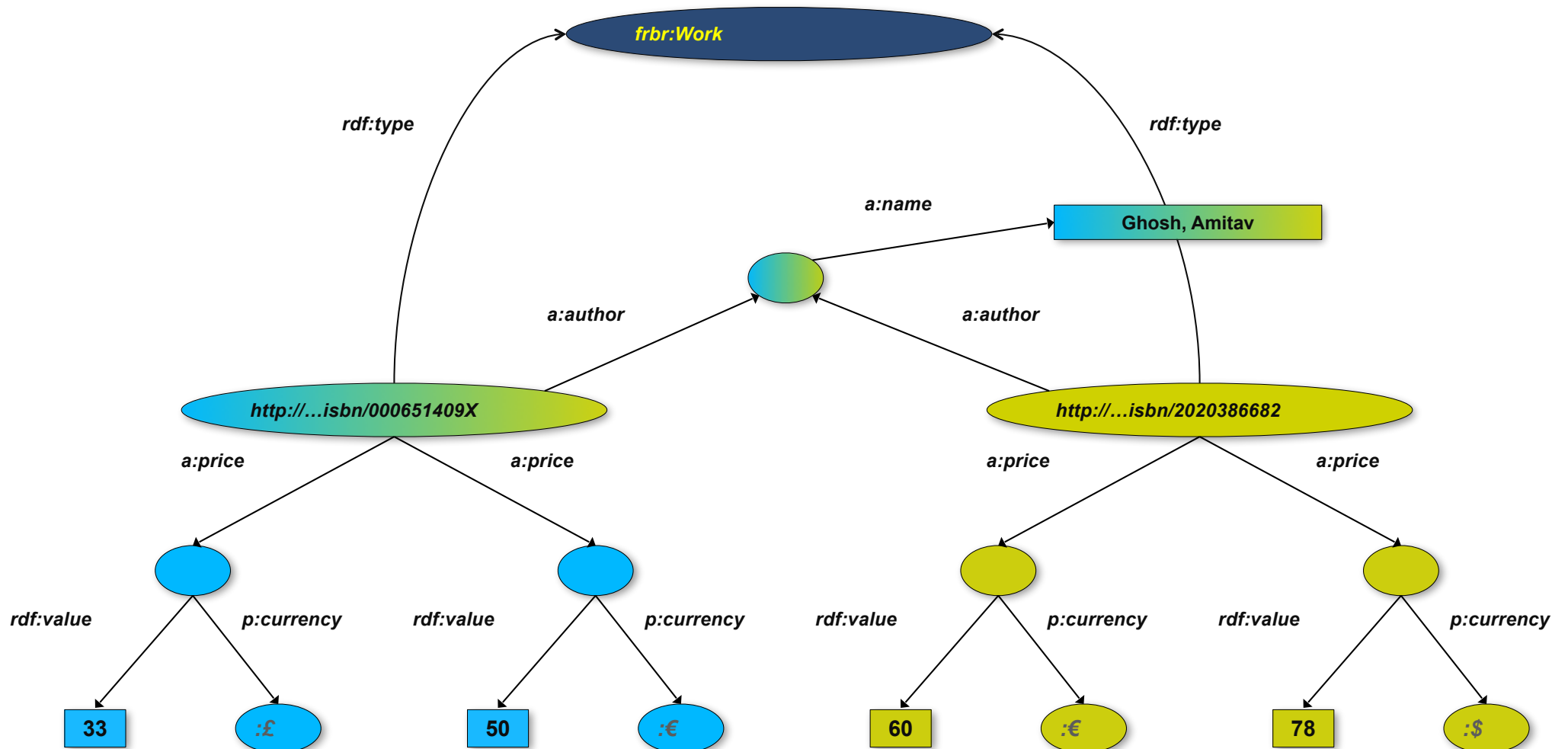
```
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```



UPDATE: INSERT

```
INSERT {?isbn rdf:type frbr:Work}
```

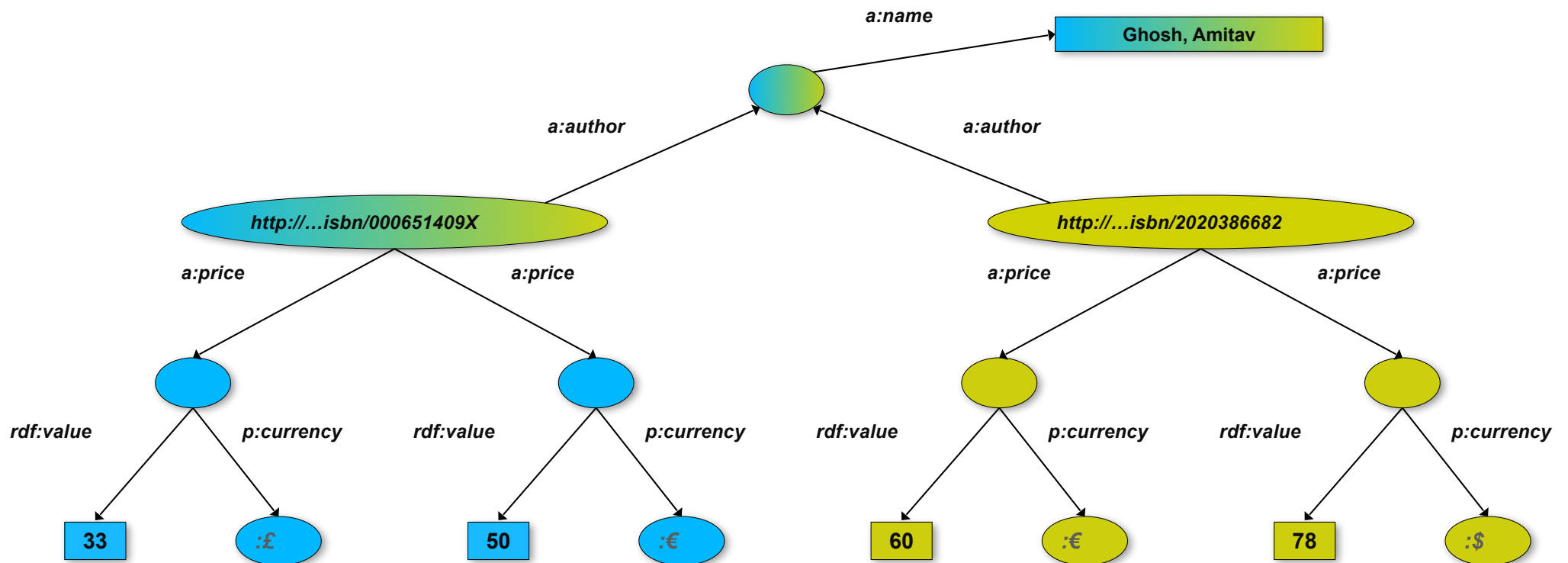
```
WHERE {?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.}
```



UPDATE: DELETE

DELETE {*?x p:currency ?currency*}

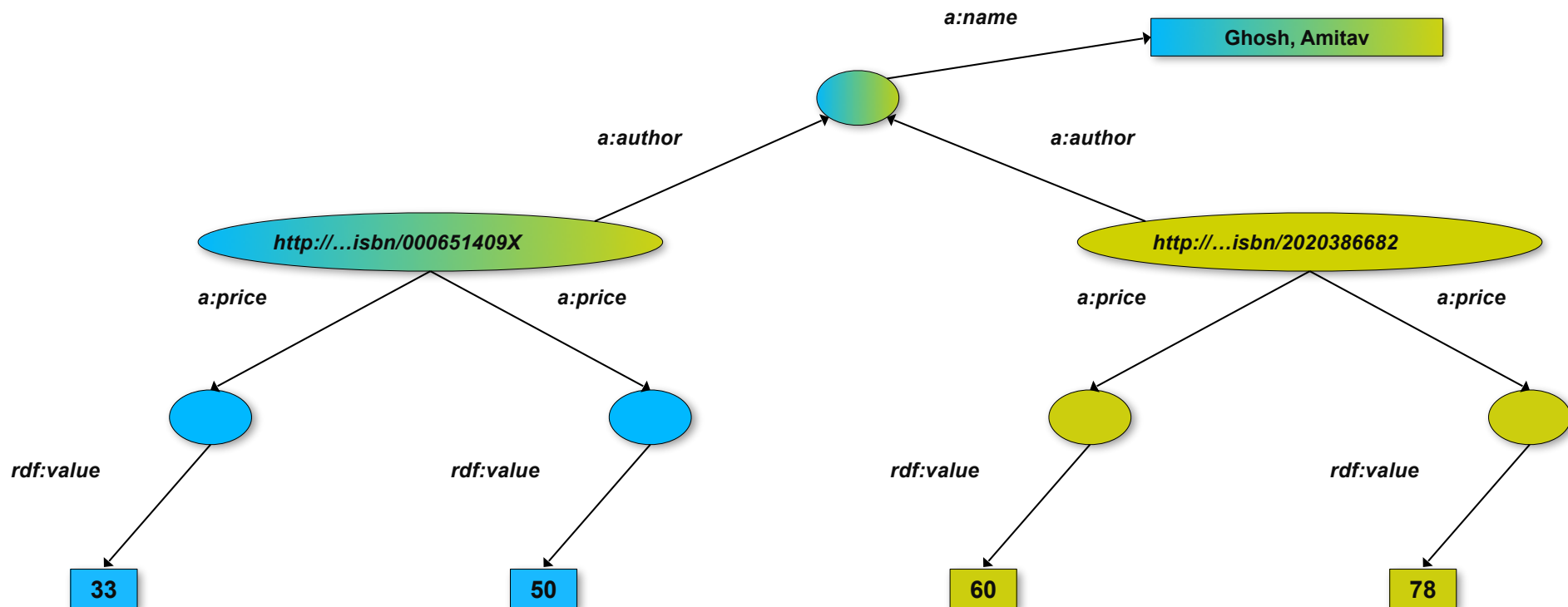
WHERE {*?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.*}



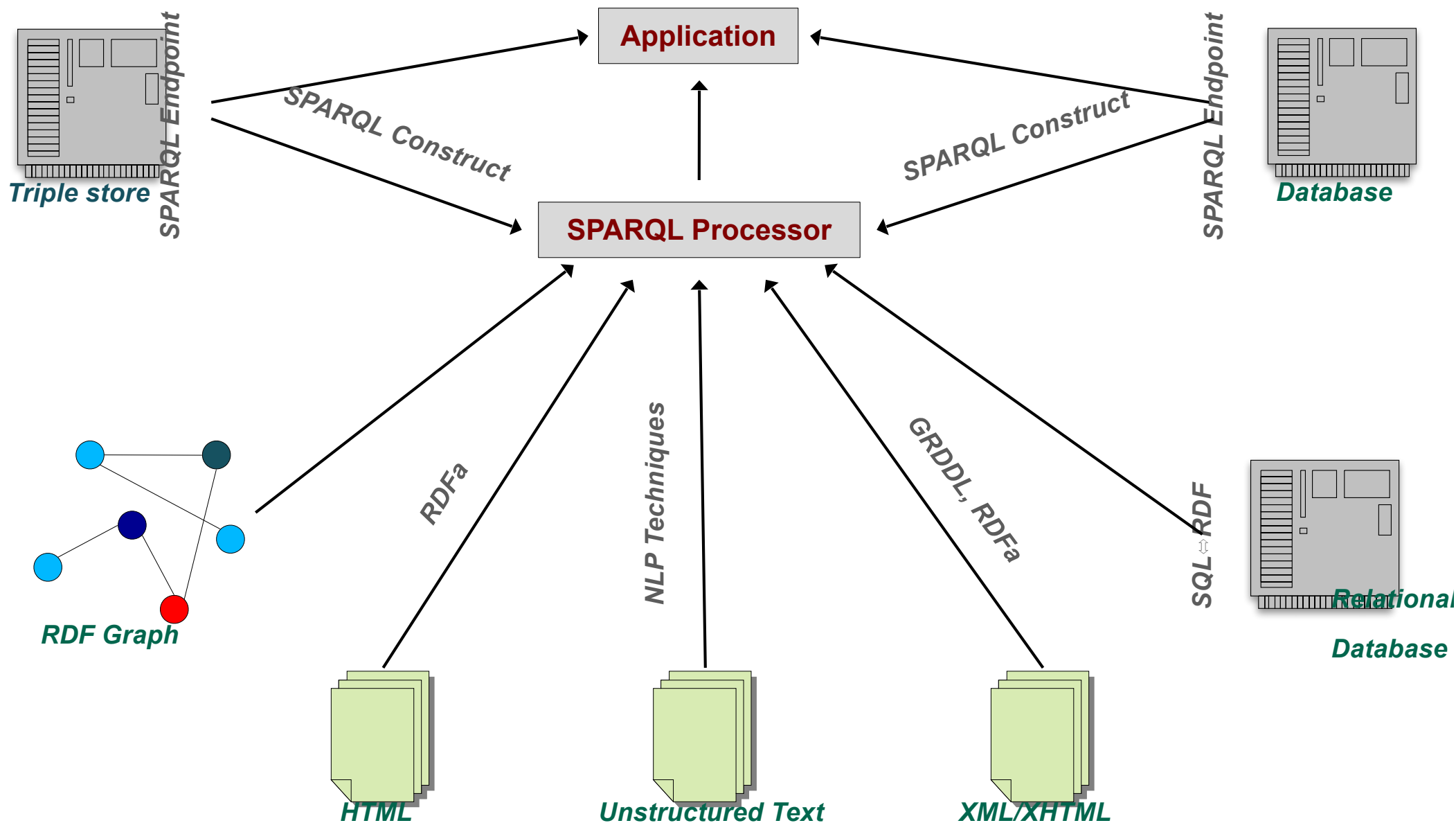
UPDATE: DELETE

DELETE {*?x p:currency ?currency*}

WHERE {*?isbn a:price ?x. ?x rdf:value ?price. ?x p:currency ?currency.*}



SPARQL COMO PUNTO DE UNIFICACION



SPARQL 1.1 COMO PUNTO DE UNIFICACION

