

Linked Data Management

Capítulo 1: Linked Data & the Semantic Web Standards

Carmem Hara

18 de outubro de 2016

Dados na *Web*

- Processamento automático de dados da Web: dados com *sintaxe* e *semântica* bem definidas
- Sintaxe:
 - linguagens de marcação (XML, CSV, JSON)
 - XML Schema: associa semântica ao XML
 - Objetivo: validação
 - Não é objetivo: facilitar o processamento automático

Exemplo: “receitas entre amigos”

- Versão 1:
 - Receitas em uma página HTML
 - Ingredientes: nome, quantidade, unidade
 - Ordem e tempo de preparação
 - SGBD relacional com palavras-chave
 - Problema: informações incompletas
 - Receitas sem cítricos (limão, laranja, ...)
 - Onde se vende lula ou polvo próximo da sua casa?

Exemplo: integração com outro grupo

- Problemas:
 - Não há tempo de preparação
 - Mandioca ou Macaxeira
 - Características adicionais: vegetariano, calorias
- As pessoas hoje são produtoras e consumidoras de conteúdo na Web



Necessidade de tornar os dados
“reutilizáveis”:
menos redundância e integração

Web Semântica

- Idéia: afirmar que **limão é uma fruta cítrica** para que esta informação seja “reutilizada” *automaticamente* por qualquer aplicação
- Uma extensão da Web tradicional que possibilita a criação, compartilhamento e reutilização inteligente do conteúdo da Web de forma automatizada

Pilha de padrões da Web Semântica



Modelo de Dados: RDF

- RDF: *Resource Description Framework*
- Estrutura básica: triplas

sujeito **propriedade** objeto

<http://dbpedia.org/resource/Lemon>

<http://dbpedia.org/ontology/genus>

<http://dbpedia.org/resource/Citrus>



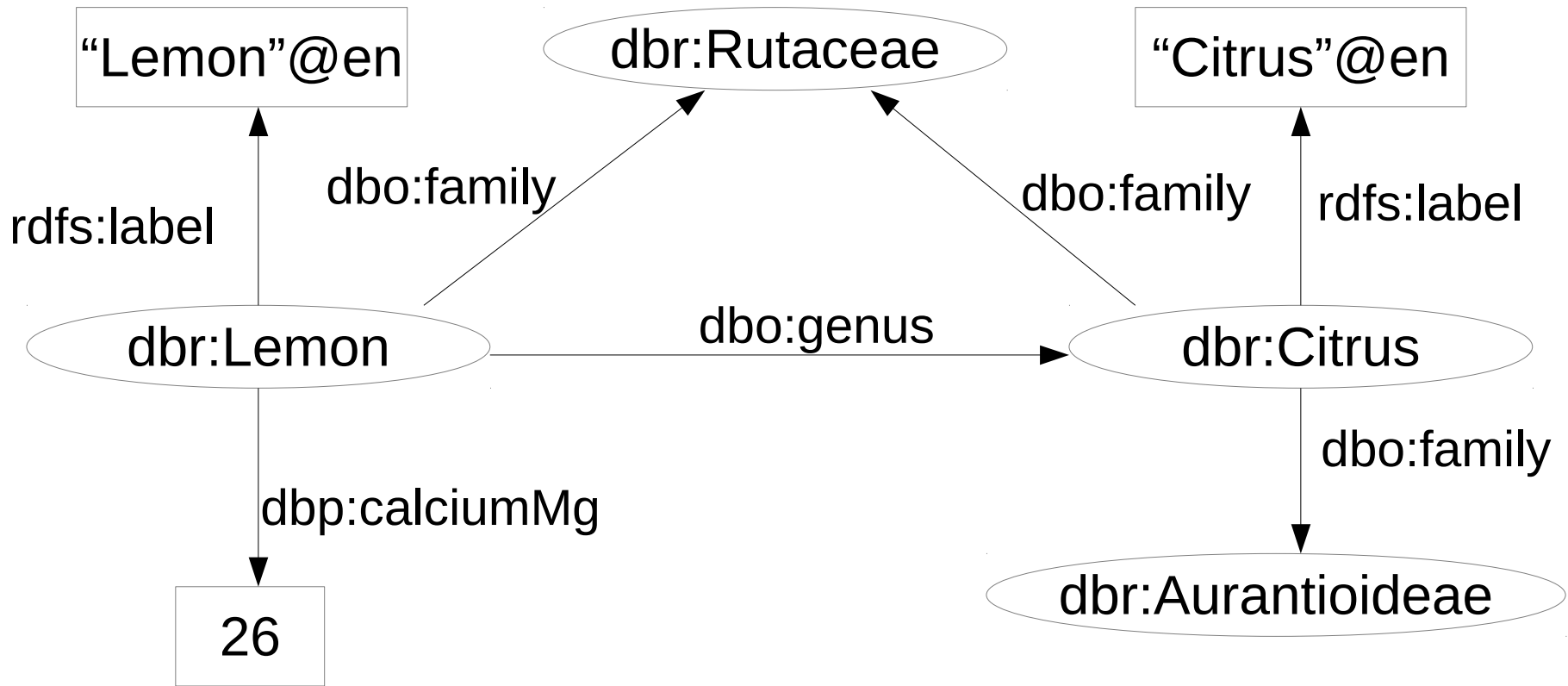
Termos RDF

- Termos RDF podem ser:
 - URIs (U): `<http://dbpedia.org/resource/Lemon>`
 - literais (L):
 - `"Hello World"@en`
 - `"2"^^xsd:int`
 - nodos brancos (B): `_:bnode1`
- Sujeito: URI ou Nodo branco
- Predicados: URI
- Objeto: URI, Literal ou Nodo branco

Sintaxe Turtle

```
#PREFIX Declarations
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbp: <http://dbpedia.org/property/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

#RDF Triples
dbr:Lemon rdfs:label "Lemon"@en .
dbr:Lemon dbp:calciumMg 26 .
dbr:Lemon dbo:family dbr:Rutaceae .
dbr:Lemon dbo:genus dbr:Citrus .
dbr:Citrus rdfs:label "Citrus"@en .
dbr:Citrus dbo:family dbr:Rutaceae .
dbr:Citrus dbo:family dbr:Aurantioideae .
```



Pergunta: como inserir

`dbo:genus rdfs:label "Genus"@en`

Vocabulário RDF

- Associação a classes

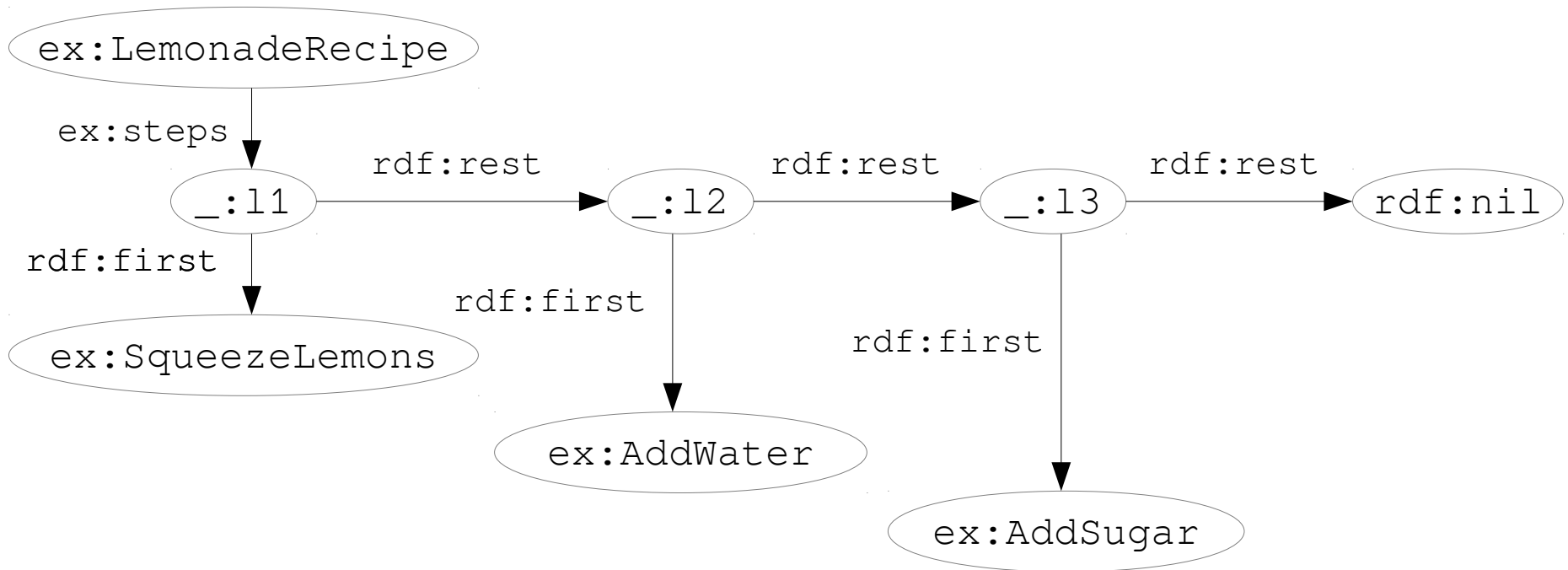
```
#Prefixos
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex:<http://minhasreceitas.br>

#Triplas RDF
ex:LemonPieRecipe rdf:type ex:Recipe .
ex:RisottoRecite rdf:type ex:Recipe .
dbr:Lemon rdf:type dbo:Plant .
dbr:Lemon rdf:type dbo:Eukariote .
dbr:Citrus rdf:type dbo:Plant .
dbr:Citrus rdf:type dbo:Species .
dbo:genus A rdf:Property .
dbo:ORDER A rdf:Property
```

*A = rdf:type
na sintaxe Turtle*

Coleções RDF - Listas

```
ex:LemonadeRecipe ex:steps _:l1 .
_:l1 rdf:first ex:SqueezeLemons .
_:l1 rdf:rest _:l2 .
_:l2 rdf:first ex:AddWater .
_:l2 rdf:rest _:l3 .
_:l3 rdf:first ex:AddSugar .
_:l3 rdf:rest rdf:nil .
```



Sintaxe RDF

- Diferentes formatos:
 - [XML](#), Turtle, N-Triples, RDFa, JSON

```
<rdf:RDF
  xmlns:rdf = "http://www.w3c.org/1999/02/22/22-rdf-syntax-ns#">
  <Description about="http://minhasreceitas.br/LemonadeRecipe">
    <steps>
      <rdf:Seq>
        <rdf:li>SqueezeLemons</rdf:li>
        <rdf:li>AddWater</rdf:li>
        <rdf:li>AddSugar</rdf:li>
      </rdf:Seq>
    </steps>
  </rdf:Description>
</rdf:RDF>
```

Semântica RDF

- Dedução (*entailment*)

A partir de:

```
dbr:Lemon dbo:family dbr:Rutacase.
```

Podemos deduzir que:

```
dbr:family A rdf:Property.
```

Regra geral:

$\underbrace{?s \ ?p \ ?o \ .}_{\text{corpo}} \longrightarrow \underbrace{?p \ a \ rdf:Property.}_{\text{cabeça}}$

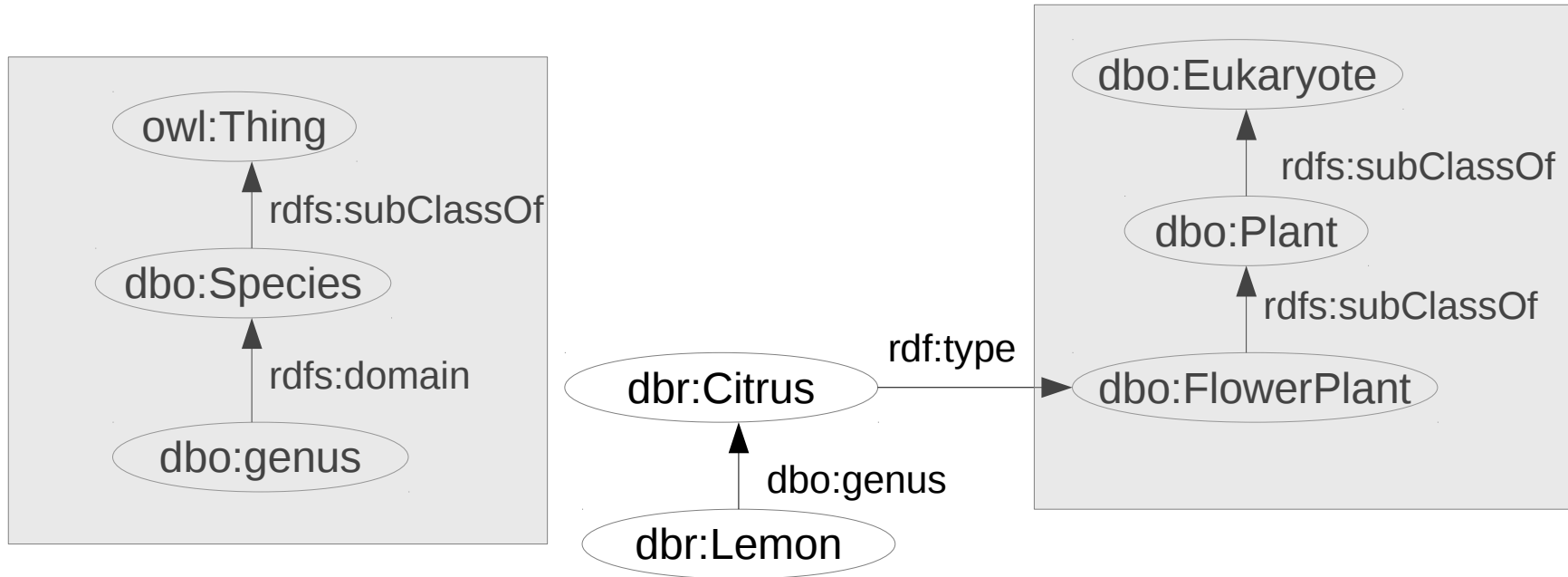
RDF Schema (RDFS)

- RDF define apenas o modelo de dados
- É preciso definir um **vocabulário**, uma linguagem que permita definir estrutura semântica
- RDFS permite definir **propriedades** de recursos (título, autor, etc.) e **relacionamentos** entre essas propriedades

Primitivas básicas do RDFS

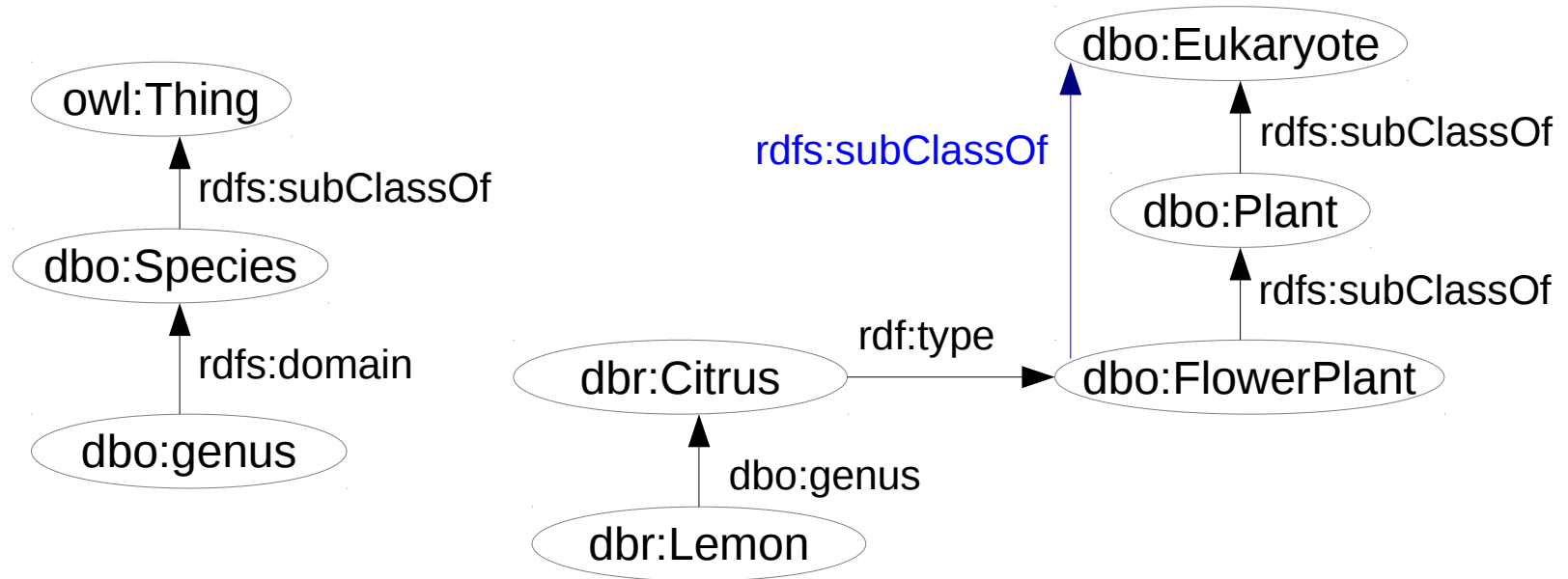
- Classes básicas
 - Classe raiz `rdfs:Resource`
 - Meta-Classe `rdfs:Class`
 - Literais `rdfs:Literal`
- Propriedade básicas
 - Hierarquia de classes `rdfs:subClassOf`
 - Hierarquia de propriedades `rdfs:subPropertyOf`
 - Domínio de propriedade `rdfs:domain`
 - Imagem de propriedade `rdfs:range`

Exemplo



```
dbr:Citrus rdf:type dbo:FlowerPlant .  
dbr:Lemon dbo:genus dbr:Citrus .  
dbo:FlowerPlant rdfs:subClassOf dbo:Plant .  
dbo:Plant rdfs:subClassOf dbo:Eukaryote .  
dbo:genus rdfs:domain dbo:Species .  
dbo:Species rdfs:subClassOf owl:Thing
```

Derivação

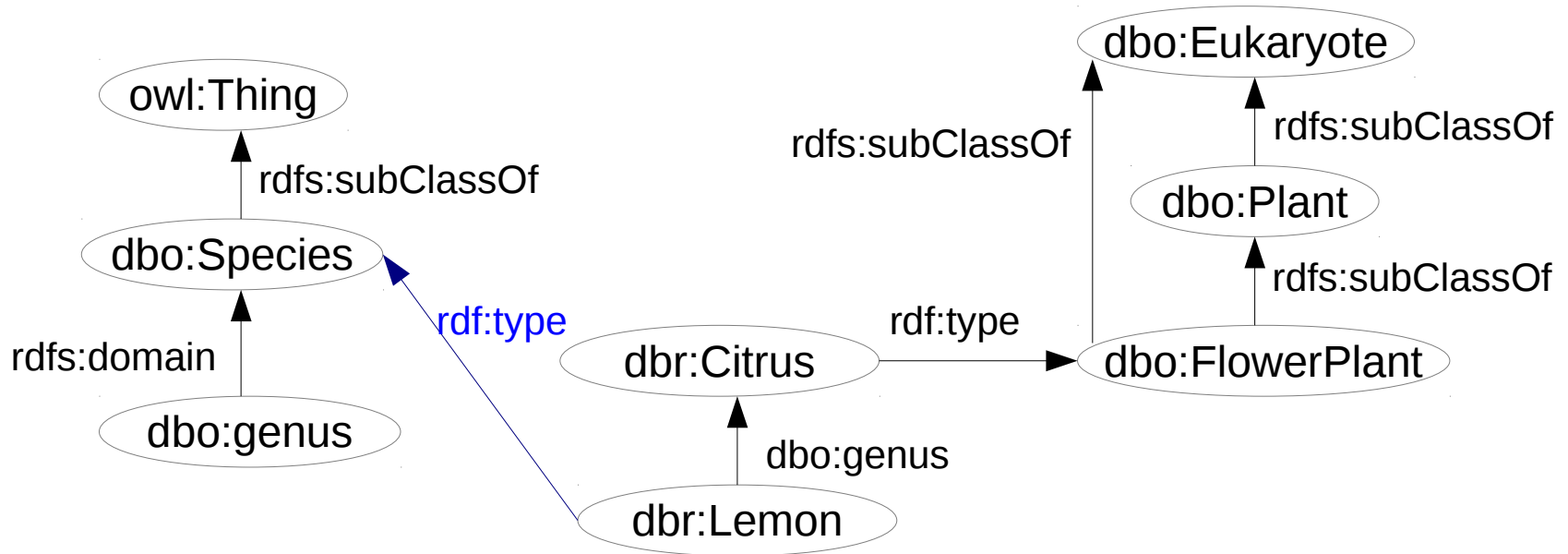


Regra:

```
?c1 rdfs:subClassOf ?c2. ?c2 rdfs:subClassOf ?c3  
→ ?c1 rdfs:subClassOf ?c3
```

```
dbo:FlowerPlant rdfs:subClassOf dbo:Eukaryote .
```

Derivação

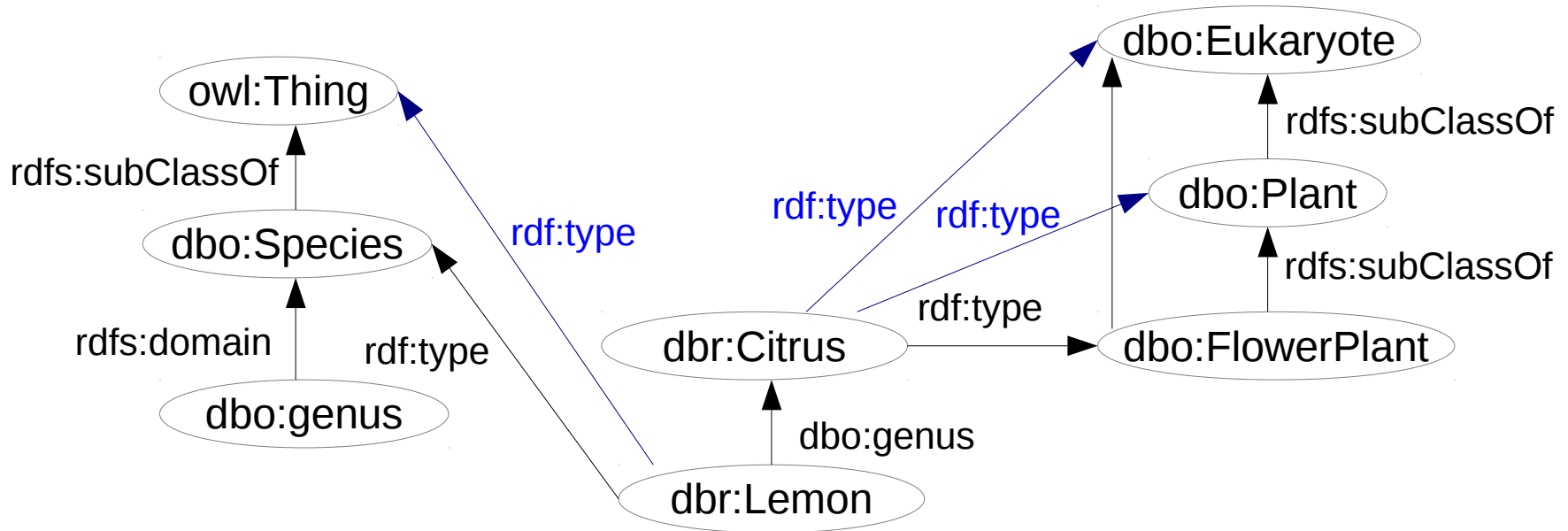


Regra:

`?p rdfs:domain ?c. ?x ?p ?y → ?x rdf:type ?c`

`dbr:Lemon rdf:type dbo:Species .`

Derivação



Regra:

```
?c1 rdfs:subClassOf ?c2. ?x rdf:type ?c1  
→ ?x rdf:type ?c2
```

```
dbr:Lemon rdf:type owl:Thing .  
dbr:Citrus rdf:type dbo:Plant .  
dbr:Citrus rdf:type dbo:Eukaryote .
```

Ontologia

- Ontologia na Ciência da Computação
 - “tudo que existe deve poder ser representado por um formalismo”
 - Constituída por um vocabulário e um conjunto de declarações
 - Base para a comunicação entre humanos e máquinas e/ou agentes

Web Ontology Language (OWL)

- Propriedades do RDF/RDFS, mais
 - owl:equivalentClass Ex: Pessoa e Humano
 - owl:disjointWith Ex: Pessoa e Flor
 - owl:equivalentProperty Ex: temFilho e paiDe
 - owl:disjointPropertyWith Ex: irmaoDe e irmaDe
 - owl:inverseOf Ex: paiDe e filhoDe
 - owl:sameAs Ex: ufpr:CarmemHara e li:CarmemHara
 - owl:differentFrom Ex: ufpr:EduardoCunha gov:EduardoCunha
 - owl:FunctionalProperty Ex: temPaiBiologico
 - owl:InverseFuncionalProperty Ex: ehPaiBiologicoDe
 - owl:TransitiveProperty Ex: ancestralDe
 - owl:SymmetricProperty Ex: parceiroDe

Exemplo

```
#PREFIXOS
@prefix fb: <http://rdf.freebase.com/ns/> .

dbr:Lemon dbo:genus dbr:Citrus .
dbr:Citrus owl:sameAs fb:en.citrus .
```

- Consequência:
 - Todas as informações da base Freebase podem ser utilizadas para inferir novos conhecimentos

SPARQL

- Linguagem de consulta
- Ortogonal ao RDFS e OWL

```
#Prefixos
```

```
@prefix dbr:<http://dbpedia.org/resource/>
```

```
@prefix dbo:<http://dbpedia.org/ontology/>
```

```
#BASES
```

```
From <http://dbpedia.org/data/Lemon.xml>
```

```
#Resultado
```

```
SELECT distinct ?genus ?order
```

```
#Padrão da consulta
```

```
WHERE {
```

```
    dbr:Lemon dbo:genus ?genus .
```

```
    dbr:Lemon dbo:ORDER ?order .
```

```
}
```

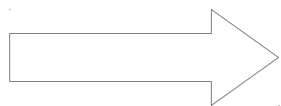
```
#Modificadores
```

```
LIMIT 2
```


Exemplo

```
SELECT distinct ?genus ?order
WHERE {
    dbr:Lemon dbo:genus ?genus .
    dbr:Lemon dbo:ORDER ?order .
}
```

```
dbr:Lemon dbo:genus dbr:Citrus .
dbr:Lemon dbo:ORDER dbr:Rosids .
dbr:Lemon dbo:ORDER dbr:Sapindales .
```



```
dbr:Citrus, dbr:Rosids
dbr:Citrus, dbr:Sapindales
```

Linked Data

- Conjunto de princípios através dos quais os padrões da Web podem facilitar a descoberta e interoperabilidade de dados estruturados
- Bom exemplo: FOAF (Friend of a Friend)
 - Define um vocabulário para descrever pessoas

Princípios

- Usar URIs para nomear recursos
- Usar Http URIs para que os recursos possam ser encontrados e de-referenciados
- Ligar seus dados com outros dados
- Retornar informações úteis em um formato livre (RDF)

Perguntas

1) Faça a correspondência entre a linguagem (ou padrão) com a sua funcionalidade que ela desempenha na pilha de padrões da Web Semântica.

- | | |
|-------------|-------------------------|
| (a) URI | (1) Sintaxe / Estrutura |
| (b) XML | (2) Modelo de dados |
| (c) RDF | (3) Ontologia |
| (d) OWL | (4) Internacionalização |
| (e) Turtle | |
| (f) RDFS | |
| (g) Unicode | |

2) Considere as regras::

?p rdfs:domain ?c . ?x ?p ?y → ?x rdf:type ?c

?p rdfs:range ?c . ?x ?p ?y → ?y rdf:type ?c

?c1 rdfs:subClassOf ?c2 . ?x rdf:type ?c1 →
 ?x rdf:type ?c2

?c1 rdfs:subClassOf ?c2 . ?c2 rdfs:subClassOf ?c3
 → ?c1 rdfs:SubClassOf ?c3

E o documento RDF:

```
@prefix e2: <http://www.inf.ufpr.br/carmem/ex2/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
e2:joao e2:orienta e2:maria .
e2:orienta rdfs:domain e2:professor .
e2:orienta rdfs:range e2:aluno .
e2:professor rdfs:subClassOf foaf:Person .
e2:aluno rdfs:subClassOf foaf:Person
```

- Quais são os fatos `rdf:type` que podem ser derivados?

