

Engenharia de Ontologias (Ontology Engineering)

Universidade Federal de Uberlândia

Faculdade de Computação

Programa de Pós-Graduação em Ciência da
Computação

Prof. Fabiano Azevedo Dorça

RDF Schema

RDF E RDF SCHEMA

- RDF Schema
- Os conceitos RDF Schema são fornecidos sob a forma de um **vocabulário** RDF;
- Isto é, como um **conjunto especializado de recursos RDF** com significados especiais.
- Os recursos no vocabulário RDF Schema têm URIrefs com o prefixo <http://www.w3.org/2000/01/rdf-schema#> (associado com o prefixo **rdfs:**).
- As descrições de vocabulário (esquemas, ontologias) escritas na linguagem RDF Schema são grafos RDF. Em outras palavras, usamos RDF para representar informações RDFS

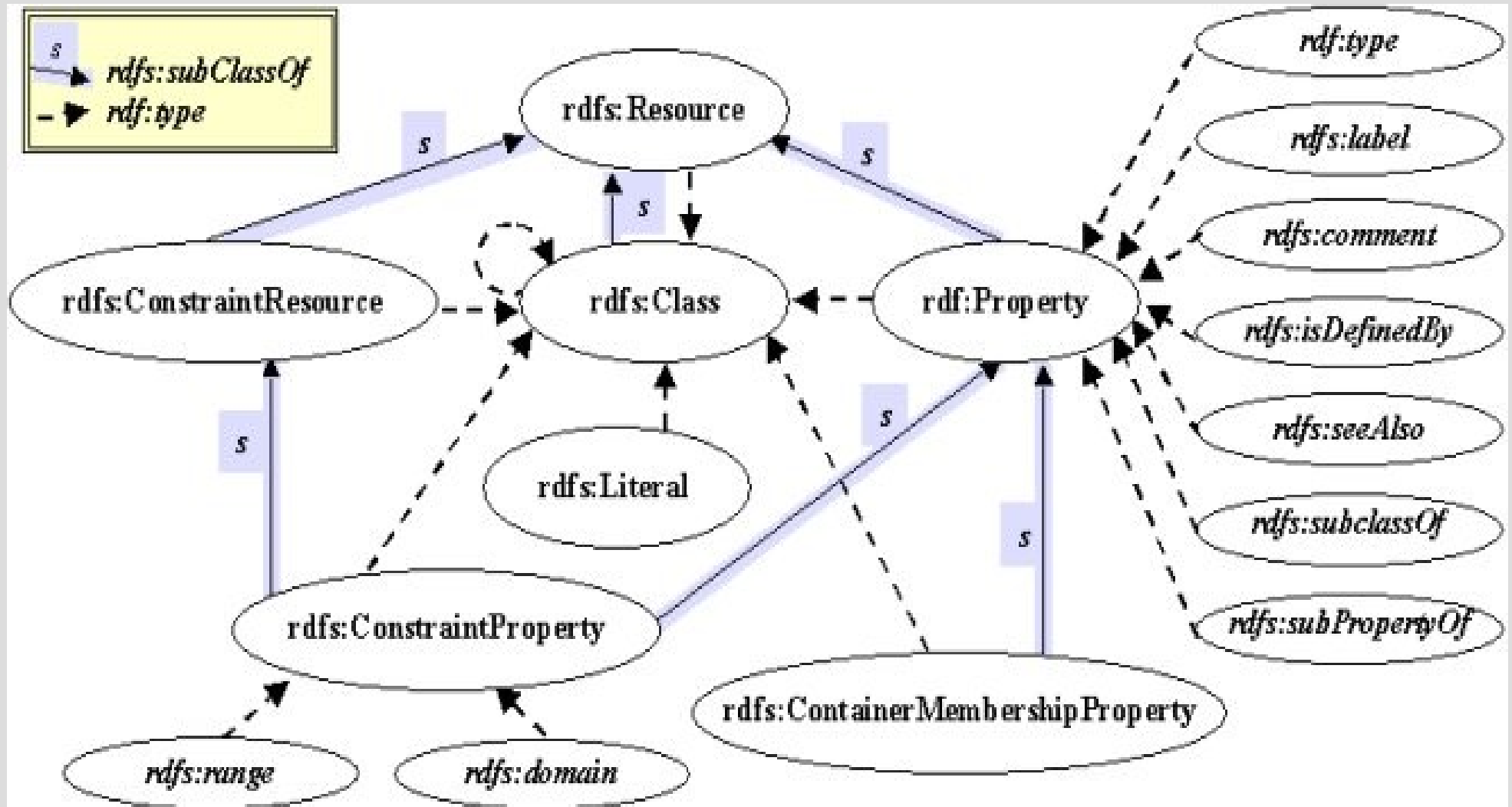
- Definição de RDF Schema (RDFS):
 - sistema de tipos para RDF, permitindo definir:
 - classes
 - hierarquia de classes
 - propriedades de classes
 - usa o próprio RDF para especificar o sistema de tipos
 - provê um conjunto de *recursos e propriedades*:
 - com significado definido
 - utilizado para descrever **classes e propriedades** de uma aplicação
 - forma um vocabulário reservado (de RDF Schema) estendendo o vocabulário de RDF

Vocabulário de RDF Schema

<i>Conceito</i>	<i>Recurso</i>	<i>Propriedade</i>
Classe	<code>rdfs:Class</code> <code>rdfs:Resource</code>	<code>rdf:type</code> <code>rdfs:subClassOf</code>
Propriedade de classe	<code>rdf:Property</code>	<code>rdf:type</code> <code>rdfs:domain</code> <code>rdfs:range</code> <code>rdfs:subPropertyOf</code>

RDF E RDF SCHEMA

- Elementos RDF/RDFS



RDF E RDF SCHEMA

- Classes

- Um passo básico em qualquer tipo de processo de descrição é identificar os **vários tipos de coisas** a serem descritas.
- RDF Schema refere-se a estes "tipos de coisas" como **classes**.
- Uma classe no RDF Schema corresponde ao conceito genérico de um **tipo ou categoria**,
- Semelhante à noção de uma **classe em linguagens de programação orientadas a objetos**, como Java ou modelos de dados orientados a objetos.

RDF E RDF SCHEMA

- Exemplo:
 - Usar o RDF Schema para fornecer informações sobre veículos motorizados, vans e caminhões.
 - Para definir classes que representam estas categorias de veículos, escrevemos as seguintes afirmações (triplas):
 - `ex:MotorVehicle rdf:type rdfs:Class .`
 - Definir uma classe explicitamente é opcional; Se escrevemos a tripla
 - `I rdf:type C`
 - Então C é inferido como uma classe (uma instância de `rdfs:Class`) em RDFS.

Exemplo:

tripla: ex:MotorVehicle rdf:type rdfs:Class .

```
<?xml version="1.0"?>
```

```
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"  
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
  <rdf:Description rdf:ID="MotorVehicle">  
    <rdf:type rdf:resource="#Class"/>  
  </rdf:Description>
```

```
</rdf:RDF>
```


RDF E RDF SCHEMA

- Notação
 - Os nomes de **classe são escritos com uma letra maiúscula** inicial
 - Os nomes de **propriedade e instância são escritos com uma letra minúscula** inicial.
- Definindo **Subclasses**
 - Definir que *vans e trucks* são tipos especializados de veículos motorizados.
 - Isso pode ser feito com as seguintes declarações RDF:

ex:Van **rdfs:subClassOf** ex:MotorVehicle .
ex:Truck **rdfs:subClassOf** ex:MotorVehicle

RDF E RDF SCHEMA

- A propriedade predefinida **rdfs:subClassOf** é usada para declarar que uma **classe** é uma **especialização** de outra classe **mais geral**.
- Classes e instâncias
- O significado da propriedade predefinida **rdfs:subClassOf** em uma declaração da forma
 - **C1 rdfs:subClassOf C2**
 - é que qualquer instância da classe C1 é também uma instância da classe C2.

RDF E RDF SCHEMA

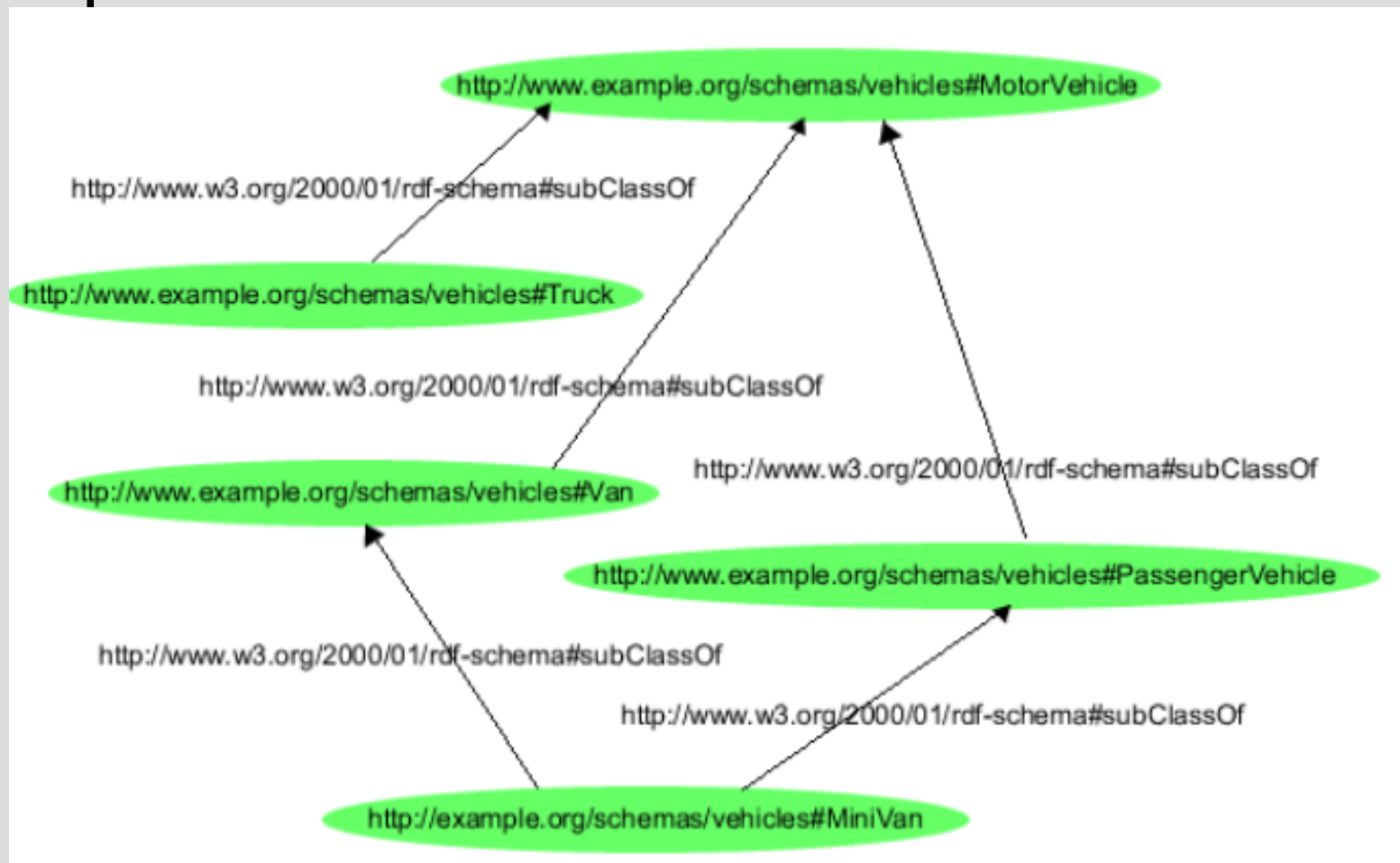
- Exemplo: Se tivermos as declarações
 - `ex:Van rdfs:subClassOf ex:MotorVehicle .`
`exthings:myCar rdf:type ex:Van .`
- Então o RDFS permite **inferir** a afirmação
 - `exthings:myCar rdf:type ex:MotorVehicle .`

RDF E RDF SCHEMA

- A propriedade **rdfs:subClassOf** é reflexiva e transitiva.
- Exemplos:
 - Se temos uma classe `ex:MotorVehicle` então RDFS permite inferir a afirmação
`ex:MotorVehicle rdfs:subClassOf ex:MotorVehicle.`
 - Se tivermos as declarações
`ex:Van rdfs:subClassOf ex:MotorVehicle.`
`ex:MiniVan rdfs:subClassOf ex:Van.`
- Então o RDFS permite inferir a afirmação
`ex:MiniVan rdfs:subClassOf ex:MotorVehicle.`

RDF E RDF SCHEMA

- Exemplo Grafo RDFS



```
<?xml version="1.0"?>
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<rdf:Description rdf:ID="MotorVehicle">
<rdf:type rdf:resource="#Class"/>
</rdf:Description>
<rdf:Description rdf:ID="PassengerVehicle">
<rdf:type rdf:resource="#Class"/>
<rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="Truck">
<rdf:type rdf:resource="#Class"/>
<rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="Van">
<rdf:type rdf:resource="#Class"/>
<rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="MiniVan">
<rdf:type rdf:resource="#Class"/>
<rdfs:subClassOf rdf:resource="#Van"/>
<rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>
</rdf:RDF>
```

RDF E RDF SCHEMA

- **Exemplo:**

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.animals.fake/animals">
```

```
<rdfs:Class rdf:ID="animal" />
```

```
<rdfs:Class rdf:ID="horse">
```

```
  <rdfs:subClassOf rdf:resource="#animal"/>
```

```
</rdfs:Class>
```

```
</rdf:RDF>
```

RDF E RDF SCHEMA

Definição de instância de classe:

- sujeito: recurso identificando a instância da classe
- propriedade: `rdf:type`
- objeto: o recurso identificando o nome da classe

• Exemplo:

`ex:companyCar rdf:type ex:MotorVehicle .`

RDF E RDF SCHEMA

Definição de instância de classe em RDF/XML:

– tripla: `ex:companyCar rdf:type ex:MotorVehicle .`

```
<?xml version="1.0"?>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ex="http://example.org/schemas/vehicles">
```

```
  <rdf:Description rdf:ID="companyCar">
```

```
    <rdf:type rdf:resource =
```

```
"http://example.org/schemas/vehicles#MotorVehicle"/>
```

```
  </rdf:Description>
```

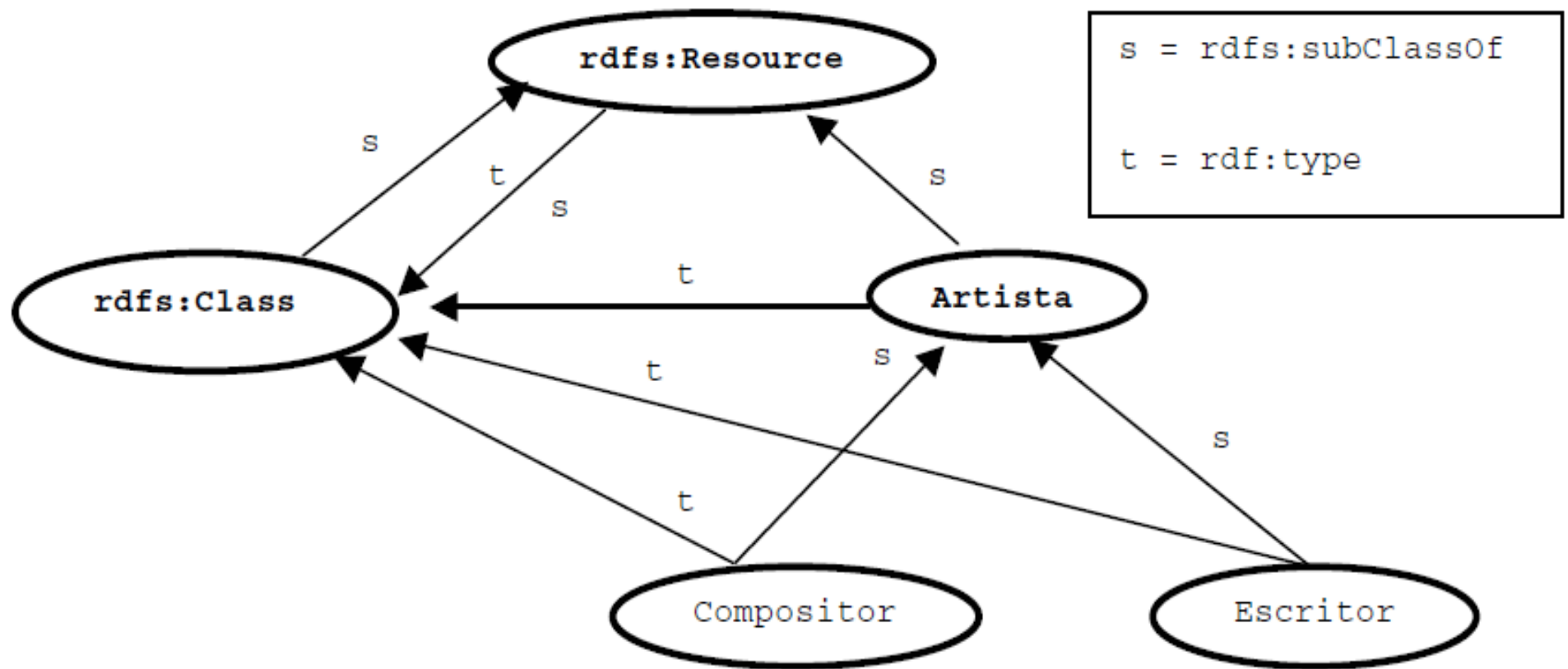
```
</rdf:RDF>
```

RDF E RDF SCHEMA

- Todas as coisas descritas pelo RDF são chamadas recursos e são **instâncias da classe rdfs:Resource**.
 - **rdfs:Resource** é a classe de tudo.
 - Todas as outras classes são subclasses desta classe.
 - Por exemplo, **rdfs:Class** é uma subclasse de **rdfs:Resource**.

RDF E RDF SCHEMA

- Exemplo:



RDF E RDF SCHEMA

A seguir é mostrado o código RDF/XML correspondente ao grafo.

```
<?xml version="1.0" ?>
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<rdfs:Class rdf:ID="Artista">
  <rdfs:subClassOf rdf:resource="#Resource" />
</rdfs:Class>
<rdfs:Class rdf:ID="Compositor">
  <rdfs:subClassOf rdf:resource="#Artista" />
</rdfs:Class>
<rdfs:Class rdf:ID="Escritor">
  <rdfs:subClassOf rdf:resource="#Artista" />
</rdfs:Class>
</rdf:RDF>
```

RDF E RDF SCHEMA

Propriedades

- **rdf:Property:**
- Permite representar as propriedades (relacionamentos) em RDF.
- Ex: `<rdf:Property rdf:ID="faz"/>`

RDF E RDF SCHEMA

- **rdfs:subPropertyOf**: especifica a relação hierárquica entre propriedades.
- Exemplos:
 - `ex:driver rdf:type rdf:Property .`
 - `ex:primaryDriver rdf:type rdf:Property .`
 - `ex:primaryDriver rdfs:subPropertyOf ex:driver .`

RDF E RDF SCHEMA

- Se os recursos S e O estão conectados pela propriedade P1, e P1 é uma sub-propriedade da propriedade P2, então o RDFS permite inferir que S e O também estão conectados pela propriedade P2.
- Exemplo: Se tivermos as declarações

```
ex:john ex:primaryDriver ex:companyCar  
ex:primaryDriver rdfs:subPropertyOf ex:driver .
```

Pode-se inferir que

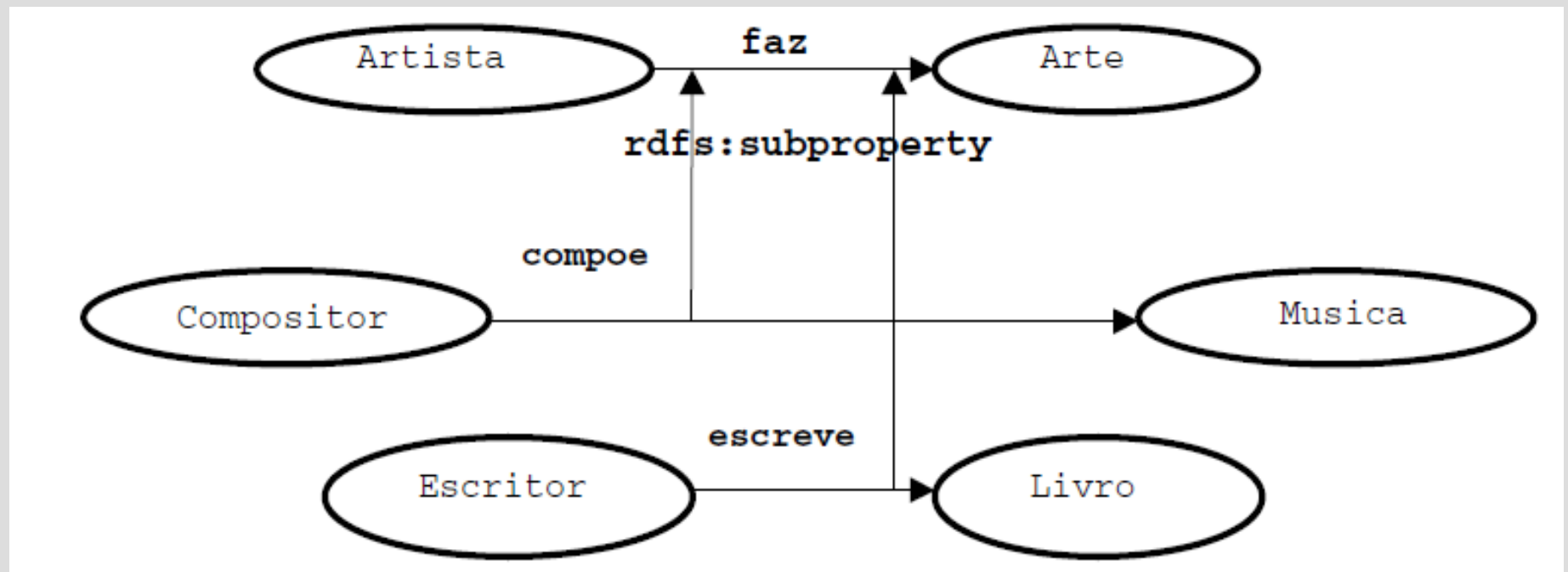
```
ex:john ex:driver ex:companyCar .
```

RDF E RDF SCHEMA

- `rdfs:subPropertyOf` é **reflexiva** e **transitiva**.
- Exemplos:
 - Se temos a propriedade **ex:driver** então RDFS permite inferir a tripla
 - `ex:driver rdfs:subPropertyOf ex:driver .`
 - Se tivermos as triplas
 - `ex:primaryDriver rdfs:subPropertyOf ex:driver .`
 - `ex:driver rdfs:subPropertyOf ex:isResponsibleFor .`
 - Então RDFS permite inferir a tripla
 - `ex:primaryDriver rdfs:subPropertyOf ex:isResponsibleFor .`

RDF E RDF SCHEMA

- Exemplo: Foram definidas a propriedade 'faz' e duas subpropriedades de 'faz': 'compõe' e 'escreve'.
- Em seguida, é mostrado o código RDF/XML correspondente à parte do grafo que se refere a propriedades.



RDF E RDF SCHEMA

```
<?xml version="1.0" ?>  
  
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"  
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">  
  
  <rdf:Property rdf:ID="faz">  
    <rdfs:subClassOf rdf:resource="#Resource"/>  
  </rdf:Property>  
  
  <rdf:Property rdf:ID="compoe">  
    <rdfs:subPropertyOf rdf:resource="#faz" />  
  </rdf:Property>  
  
  <rdf:Property rdf:ID="escreve">  
    <rdfs:subPropertyOf rdf:resource="#faz"/>  
  </rdf:Property>  
  
</rdf:RDF>
```

RDF E RDF SCHEMA

- No RDFS as definições de **propriedade** são **independentes** das definições de classe.
- Em outras palavras, uma definição de propriedade pode ser feita sem qualquer referência a uma classe.

Exemplo:

```
<?xml version="1.0" ?>
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
<rdf:Property rdf:ID="faz">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdf:Property>
<rdf:Property rdf:ID="compoe">
  <rdfs:subPropertyOf rdf:resource="#faz" />
</rdf:Property>
<rdf:Property rdf:ID="escreve">
  <rdfs:subPropertyOf rdf:resource="#faz"/>
</rdf:Property>
```

```
<rdfs:Class rdf:ID="Artista">
  <rdfs:subClassOf rdf:resource="#Resource" />
</rdfs:Class>
<rdfs:Class rdf:ID="Compositor">
  <rdfs:subClassOf rdf:resource="#Artista" />
</rdfs:Class>
<rdfs:Class rdf:ID="Escritor">
  <rdfs:subClassOf rdf:resource="#Artista" />
</rdfs:Class>
```

```
</rdf:RDF>
```

RDF E RDF SCHEMA

- Unindo...

...

```
<rdf:Description rdf:ID="Musica"/>
```

```
<rdf:Description rdf:ID="Joao"/>
```

```
<rdf:Description rdf:about="#Joao">
```

```
  <rdf:type rdf:resource="#Escritor"/>
```

```
  <rdfs:compoes rdf:resource="#Musica"/>
```

```
</rdf:Description>
```

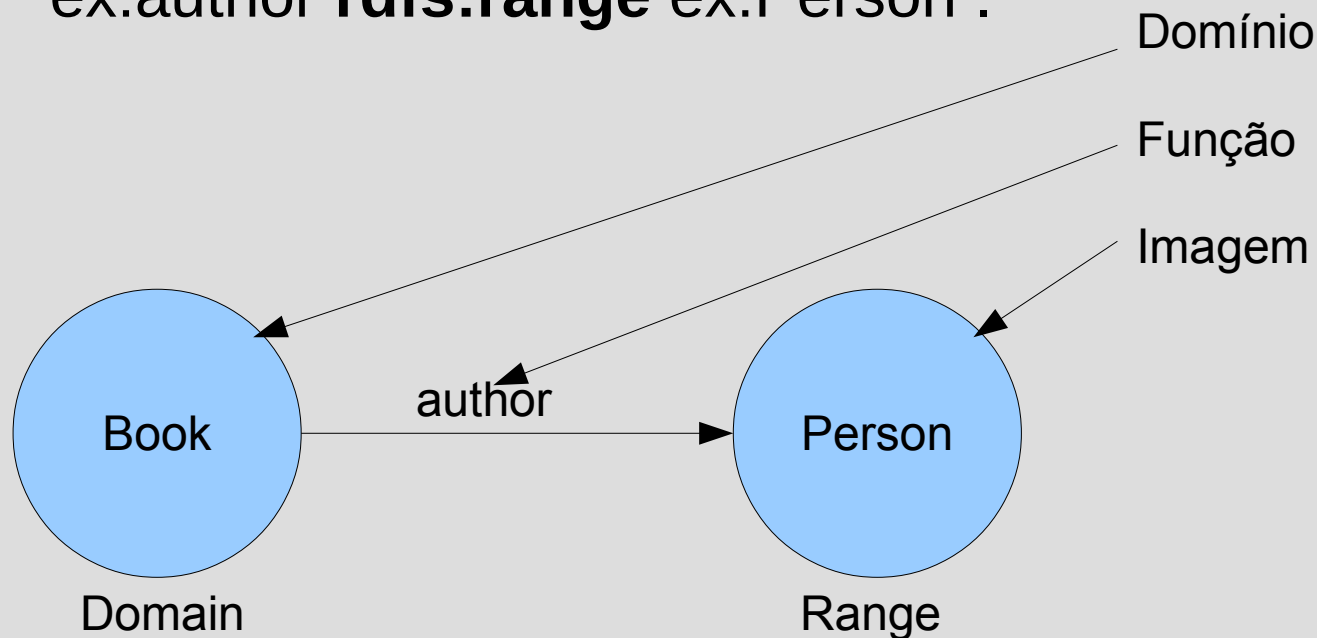
```
</rdf:RDF>
```

RDF E RDF SCHEMA

- Domínio e intervalo
- Opcionalmente, as propriedades podem ser declaradas para serem aplicadas a determinadas instâncias de classes definindo seu **domínio (domain) e intervalo (range)**.
- O RDFS fornece vocabulário para descrever como as **propriedades e classes** devem ser usadas em conjunto em dados RDF.
- O predicado **rdfs:domain** é usado para indicar que uma **propriedade se aplica a instâncias** de uma classe designada (isto é, define o domínio da propriedade).
- O predicado **rdfs:range** é usado para indicar que os **valores de uma propriedade** são instâncias de uma classe designada (isto é, define o intervalo da propriedade).

RDF E RDF SCHEMA

- Exemplo
 - `ex:Book rdf:type rdfs:Class .`
 - `ex:Person rdf:type rdfs:Class .`
 - `ex:author rdf:type rdf:Property .`
 - `ex:author rdfs:domain ex:Book .`
 - `ex:author rdfs:range ex:Person .`



RDF E RDF SCHEMA

- Deduções em relação a domain e range
- Se
 - hasMother **rdfs:domain** Person .
 - frank **hasMother** frances .
- Então, pode-se inferir que:
 - frank **rdf:type** Person.

Se

- hasMother **rdfs:range** Person .
- frank **hasMother** frances
- Então, pode-se inferir que:
 - frances **rdf:type** Person.

RDF E RDF SCHEMA

- Para uma propriedade, podemos ter zero, um ou mais de um domínio ou range.
- Desta forma, se tivermos
 - `P rdfs:range C1 .`
 - `P rdfs:range C2 .`
- Então podemos inferir que os valores de propriedade P são instâncias de ambos C1 e C2.

RDF E RDF SCHEMA

- Exemplo: Se tivermos
 - `hasMother rdfs:range Female .`
 - `hasMother rdfs:range Person .`
 - `frank ex:hasMother frances .`
 - Então podemos inferir que frances é uma instância (subtipo) de Female e Person.

RDF E RDF SCHEMA

- Tipos de dados para range
- A propriedade **rdfs:range** também pode ser usada para indicar que o valor de uma propriedade é dado por um literal tipado.
- Exemplo:
 - `ex:age rdf:type rdf:Property .`
 - `ex:age rdfs:range xsd:integer .`
- Opcionalmente, também podemos afirmar que `xsd:integer` é um tipo de dados:
- `xsd:integer rdf:type rdfs:Datatype .`

Exemplo:

```
<?xml version="1.0" ?>
```

```
<rdf:RDF xml:base="http://www.w3.org/2000/01/rdf-schema"
```

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
<rdfs:Class rdf:ID="ResourceAccessRule">
```

```
  <rdfs:label>Access Rule</rdfs:label>
```

```
  <rdfs:comment>Assertion of access privileges to a resource.</rdfs:comment>
```

```
</rdfs:Class>
```

```
<rdf:Property rdf:ID="access">
```

```
  <rdfs:label>access</rdfs:label>
```

```
  <rdfs:comment>Access privileges extended to an accessor.</rdfs:comment>
```

```
  <rdfs:domain rdf:resource="#ResourceAccessRule"/>
```

```
  <rdfs:range rdf:resource="#Literal"/>
```

```
</rdf:Property>
```

```
<rdf:Description rdf:ID="companyCar">
```

```
  <rdf:type rdf:resource="#ResourceAccessRule"/>
```

```
  <rdfs:access rdf:resource="TESTE"/>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

RDF E RDF SCHEMA

- A propriedade **rdfs:label**
- rdfs:label é uma instância de **rdf:Property** que pode ser usada para fornecer uma versão legível do nome de um recurso.
 - O rdfs:domain de rdfs:label é rdfs:Resource.
 - O rdfs:range de rdfs:label é rdfs:Literal.

RDF E RDF SCHEMA

- A propriedade **rdfs:comment**
- rdfs:comment é uma instância de rdf:Property que pode ser usada para fornecer uma **descrição legível** por **humanos** de um recurso.
- O rdfs:domain de rdfs:comment é rdfs:Resource. O rdfs:range de rdfs:comment é rdfs:Literal.

RDF E RDF SCHEMA

- **rdfs:Literal**
 - A classe `rdfs:Literal` é a classe de valores literais, como strings e inteiros.
 - Valores de propriedade como strings são exemplos de literais RDF.
 - Literais podem ser simples ou tipados.
 - Um literal tipado é uma instância da classe **`rdfs:Datatype`**.
- `rdfs:Literal` é uma instância (tipo) de `rdfs:Class`.
- `rdfs:Literal` é uma subclasse de `rdfs:Resource`.
- `rdfs:Datatype` é uma subclasse de `rdfs:Literal`.

RDF E RDF SCHEMA

- A propriedade `rdfs:seeAlso`
 - `rdfs:seeAlso` é uma instância de `rdf:Property` que é usada para indicar um recurso que pode fornecer **informações adicionais** sobre o **sujeito**.
 - A tripla **S rdfs:see Also O** indica que o recurso **O** pode fornecer informações adicionais sobre **S**.
 - O `rdfs:domain` de `rdfs:seeAlso` é `rdfs:Resource`.
 - O `rdfs:range` de `rdfs:seeAlso` é `rdfs:Resource`.

RDF E RDF SCHEMA

- Exemplo:

```
<rdf:Description about="http://www.w3.org/2001/02/acls/ns#">  
  <rdfs:comment>A namespace for describing Access Control  
Lists</rdfs:comment>  
  <rdfs:comment>$Revision: 1.3 $ $Date: 2001/07/11 00:11:43 $  
</rdfs:comment>  
  <rdfs:seeAlso resource="http://www.w3.org/2001/02/acls/acls-for-ns"/>  
</rdf:Description>
```

RDF E RDF SCHEMA

- A propriedade `rdfs:isDefinedBy`
 - `rdfs:isDefinedBy` é uma instância de `rdf:Property` que é usada para indicar um **recurso que define o subject**.
 - Esta propriedade pode ser usada para indicar um **vocabulário RDF no qual um recurso é descrito**.
 - Uma tripla **S `rdfs:isDefinedBy` O** indica que a fonte O define S.
 - `rdfs:isDefinedBy` é uma subpropriedade de `rdfs:seeAlso`.
 - O `rdfs:domain` de `rdfs:isDefinedBy` é um `rdfs:Resource`.
 - O `rdfs:range` de `rdfs:isDefinedBy` é `rdfs:Resource`.

RDF E RDF SCHEMA

- Exemplo:

```
<rdf:Property ID="hasAccessTo">  
  <rdfs:label>has access to</rdfs:label>  
  <rdfs:comment>Relates an Access Rule to the resources to which the  
rule applies. The inverse relation is 'accessedBy'</rdfs:comment>  
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-  
schema#Resource"/>  
  <rdfs:domain rdf:resource="#ResourceAccessRule"/>  
  <rdfs:isDefinedBy resource="http://www.w3.org/2001/02/acls/ns#"/>  
</rdf:Property>
```

- Fim