

# Web ngữ nghĩa

Soạn bởi: Nguyễn Bá Ngọc

## Chương 4

Hà Nội-2021

## Chương 4.

# Ngữ nghĩa hình thức

# Nội dung

- 4.1. Vai trò của ngữ nghĩa hình thức
- 4.2. Ngữ nghĩa lô-gic của RDF(S)
- 4.3. Suy diễn dựa trên luật
- 4.4. Các hạn chế của RDF(S)

# Nội dung



4.1. Vai trò của ngữ nghĩa hình thức

4.2. Ngữ nghĩa lô-gic của RDF(S)

4.3. Suy diễn dựa trên luật

4.4. Các hạn chế của RDF(S)

# Ngữ nghĩa hình thức

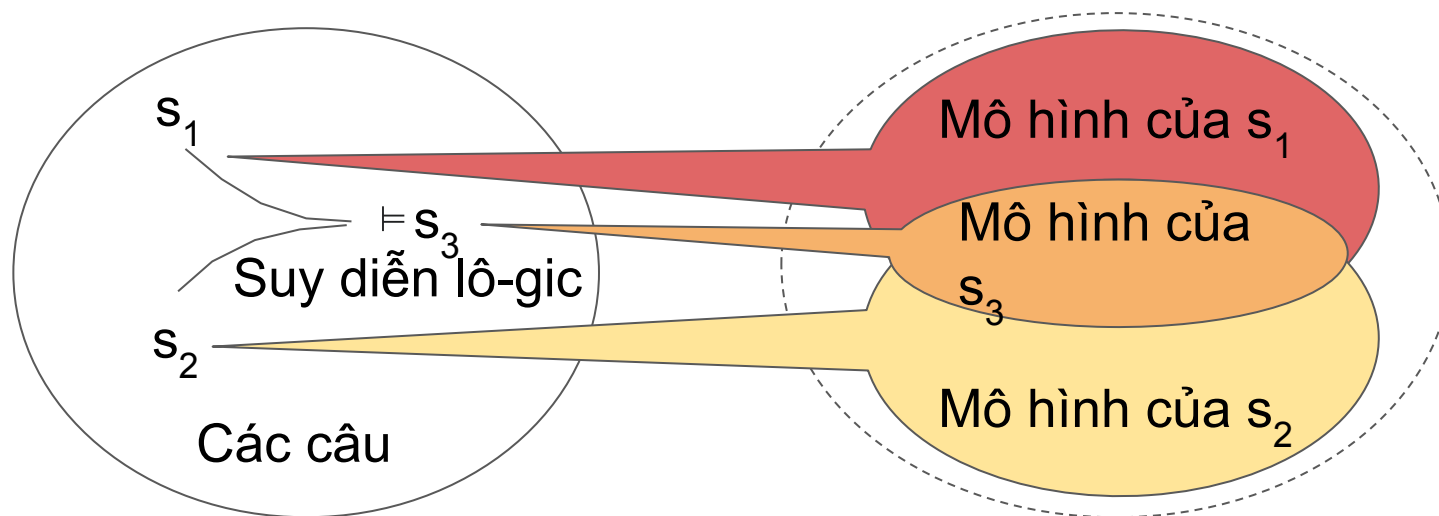
- Vấn đề trong phát triển phần mềm cho RDF(S)
  - Dù đã có quy chuẩn nhưng các phần mềm vẫn không tương thích.
  - Ví dụ: Khi sử dụng các hệ lưu trữ bộ ba khác nhau:
    - Cùng 1 CSDL RDF và 1 truy vấn SPARQL
    - ... Nhưng kết quả truy vấn có thể khác nhau.
- Ngữ nghĩa hình thức đã được biên soạn:
  - Được sử dụng làm cơ sở lý thuyết ...
  - ... qua đó nâng cao tính tương thích giữa các phần mềm.

# Mối liên hệ giữa RDF(S) và lô-gic

- Các thành phần trong câu RDF(S):
  - Tập từ vựng  $V$  để tạo các câu bao gồm: Định danh (IRIs), nút rỗng (bnodes), và các hằng giá trị (literals)
- Bộ ba:
$$(s, p, o) \in (\text{IRI} \cup \text{bnode}) \times \text{IRI} \times (\text{IRI} \cup \text{bnode} \cup \text{literal}).$$
- Tập bộ ba còn được gọi là đồ thị RDF.

# Mối liên hệ giữa RDF(S) và Lô-gic<sub>(2)</sub>

- Phép kéo theo trong lô-gic:
  - Câu hỏi: Đồ thị G có suy ra được đồ thị G' hay không?
  - Ký hiệu là:  $G \models G'$ , trong đó G và G' là các đồ thị RDF(S)
- Chúng ta sẽ định nghĩa các biểu diễn và xác định các điều kiện để 1 biểu diễn là mô hình của đồ thị.



# Nội dung

4.1. Vai trò của ngữ nghĩa hình thức

4.2. Ngữ nghĩa lô-gic của RDF(S)

4.3. Suy diễn dựa trên luật

4.4. Các hạn chế của RDF(S)



# Ngữ nghĩa của RDF(S)

Các biểu diễn đơn giản

Các biểu diễn RDF

Các biểu diễn RDFS

Càng giới hạn tập biểu diễn, càng có nhiều khả năng thực hiện suy diễn (tăng khả năng suy diễn).

# Các ký hiệu cho một biểu diễn đơn giản

Các thành phần của biểu diễn  $\mathcal{I}$  cho đồ thị RDF có bộ từ vựng  $V$ :

- $IR$  - một tập không rỗng, còn được gọi là không gian mô tả của  $\mathcal{I}$
- $LV \subseteq IR$  - tập các hằng giá trị, bao gồm (tối thiểu) tất cả các giá trị không định kiểu từ  $V$ ,
- $IP$  - tập các thuộc tính của  $\mathcal{I}$ , có thể chồng lấn với  $IR$ .
- $\mathcal{I}_S$  - ánh xạ URIs từ  $V$  sang hợp của  $IR$  và  $IP$ ;

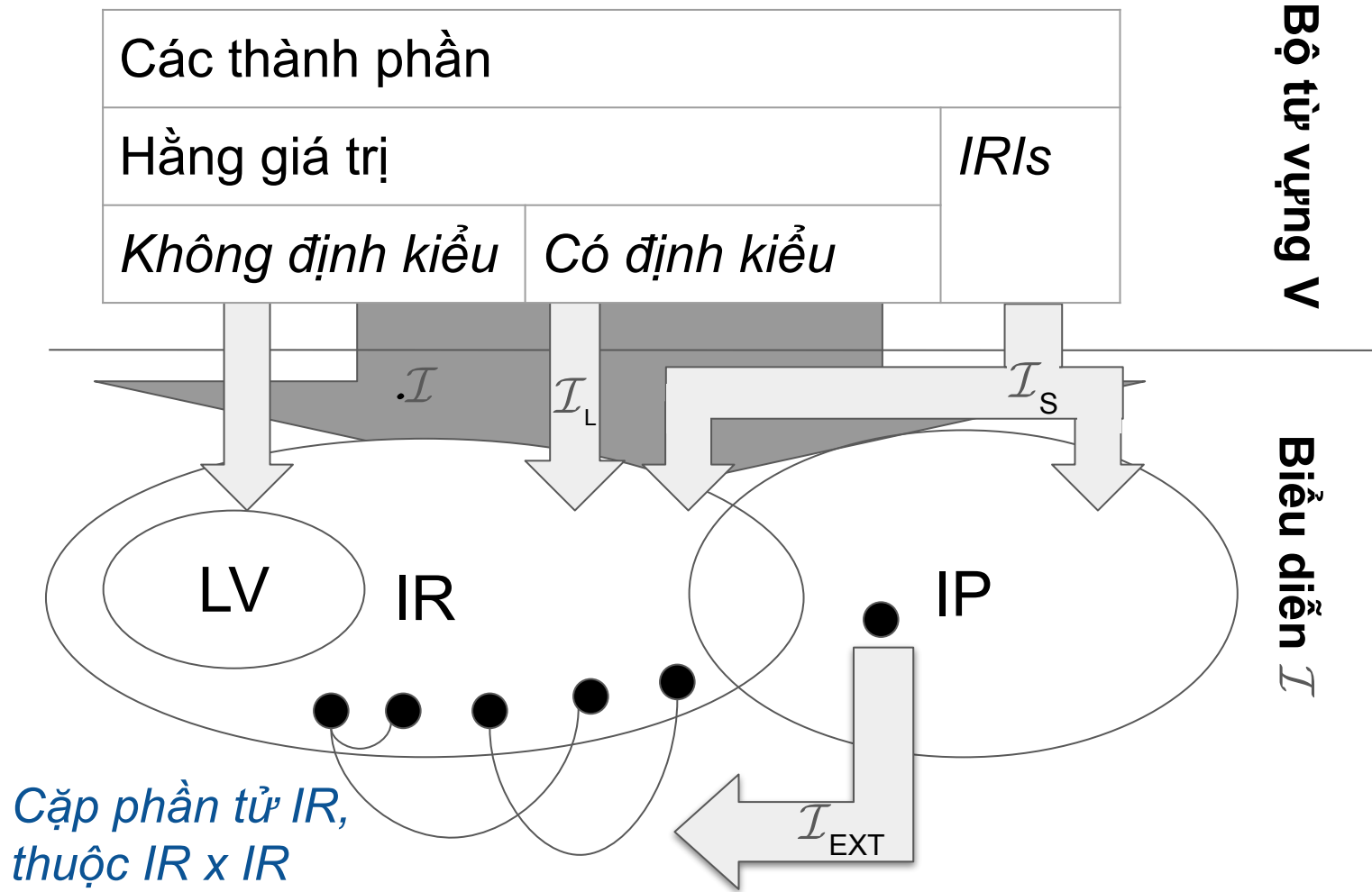
$$\mathcal{I}_S : V \rightarrow IR \cup IP,$$

- $\mathcal{I}_L$  - ánh xạ giá trị có định kiểu từ  $V$  sang tập  $IR$ .
- $\mathcal{I}_{EXT}$  - ánh xạ thuộc tính tới 1 tập cặp phần tử của  $IR$ ;

$$\mathcal{I}_{EXT} : IP \rightarrow IR \times IR, \mathcal{I}_{EXT}(p)$$

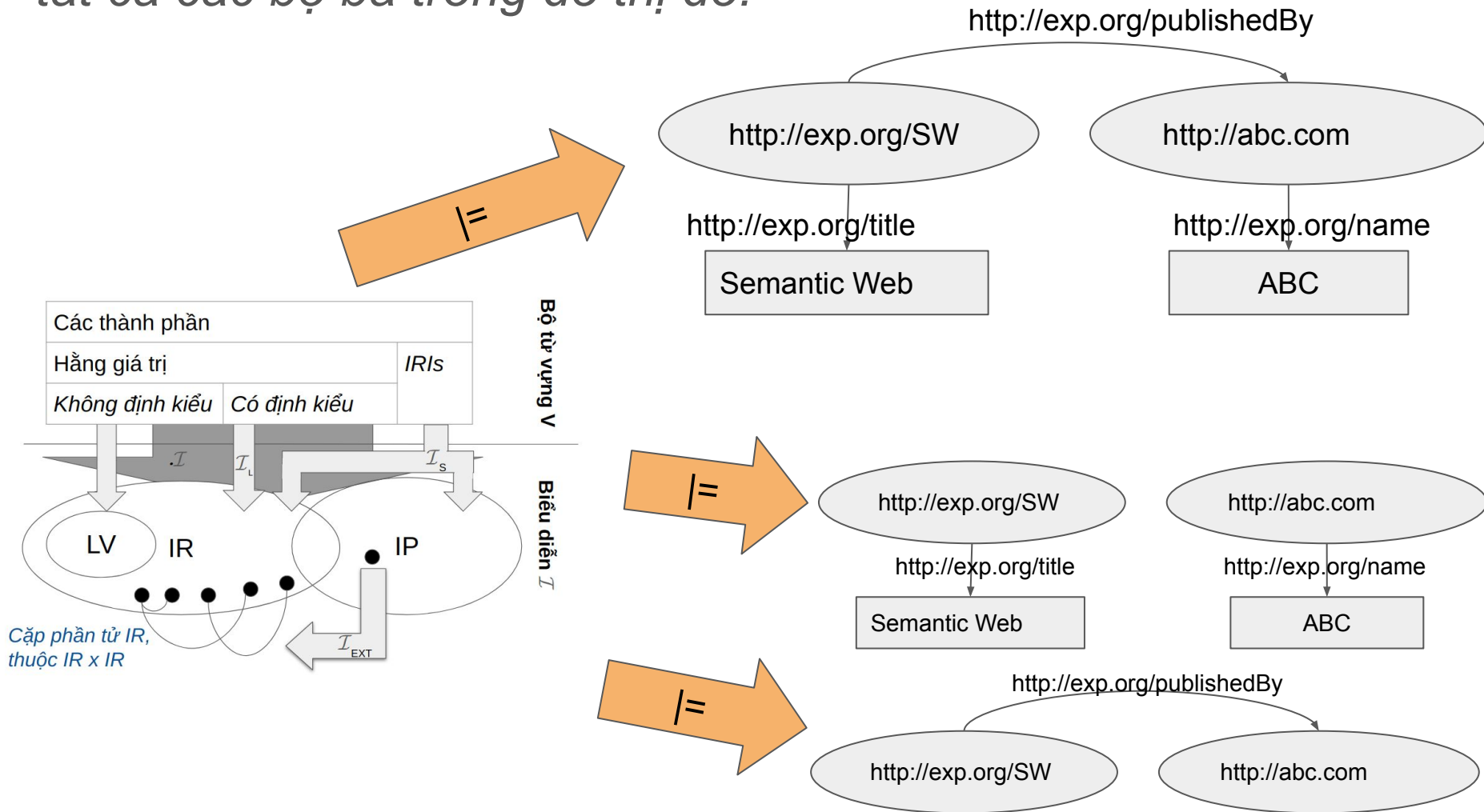
được gọi là mở rộng của thuộc tính  $p$ .

# Sơ đồ khái niệm



# Suy diễn đơn giản

*Một biểu diễn là mô hình của 1 đồ thị nếu nó là mô hình của tất cả các bộ ba trong đồ thị đó.*

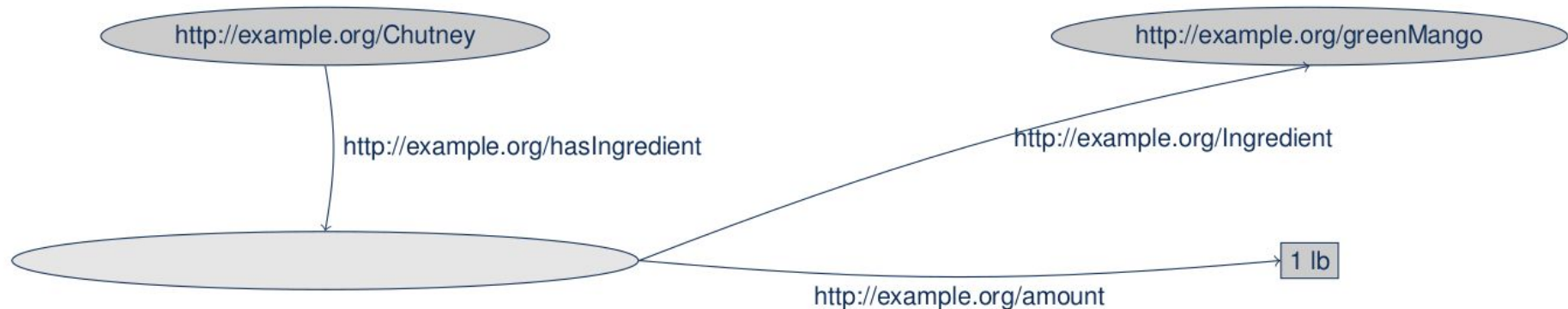




# Suy diễn đơn giản<sub>(3)</sub>: Nút rỗng

- Đặt  $A$  là ánh xạ nút rỗng trong  $G$  tới phần tử trong  $IR$ 
  - Đồng thời ký hiệu  $\mathcal{I} + A$  là 1 biểu diễn kết hợp  $\mathcal{I}$  và  $A$ , có các đặc điểm giống hệt như  $\mathcal{I}$  trên tập URIs và hằng giá trị
  - $(b)^{\mathcal{I} + A} = A(b)$ , đồng thời  $\mathcal{I} + A$  cũng có thể được áp dụng cho bộ ba và cho đồ thị RDF.
- Chúng ta có thể mở rộng phạm vi suy diễn với đồ thị RDF có chứa nút rỗng như sau:
  - $\mathcal{I}$  là mô hình của  $G$ , nếu tồn tại  $A$  sao cho  $\mathcal{I} + A$  là mô hình của tất cả các bộ ba trong  $G$ ;  $\exists A: G^{\mathcal{I} + A} = \text{true}$

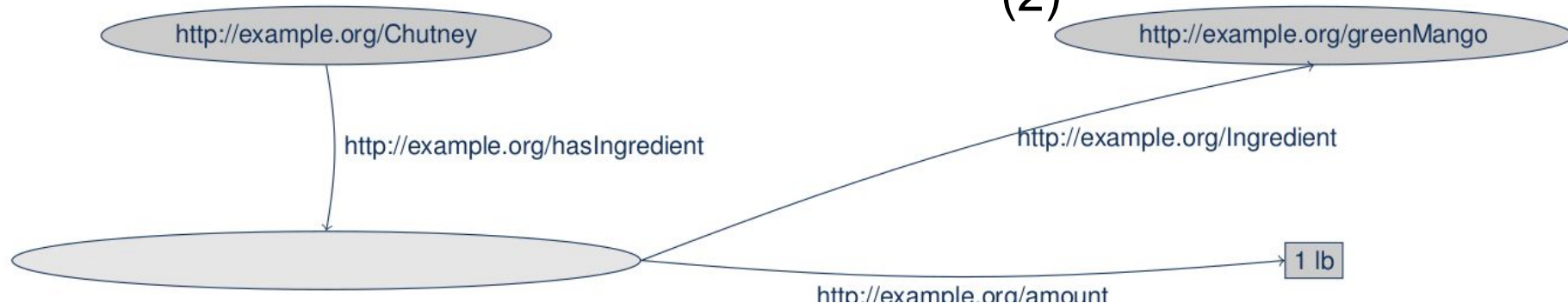
# Ví dụ 4.1. Biểu diễn đơn giản



IR	=	$\{c, g, h, z, l, m, 1 \text{ lb}\}$	$I_S =$	ex:Chutney	$\mapsto c$
IP	=	$\{h, z, m\}$		ex:greenMango	$\mapsto g$
LV	=	$\{1 \text{ lb}\}$		ex:hasIngredient	$\mapsto h$
$I_{EXT}$	=	$h \mapsto \{\langle c, l \rangle\}$		ex:ingredient	$\mapsto z$
		$z \mapsto \{\langle l, g \rangle\}$		ex:amount	$\mapsto m$
		$m \mapsto \{\langle l, 1 \text{ lb} \rangle\}$	Trong ví dụ này $\mathcal{I}_L$ là hàm rỗng		

*$\mathcal{I}$  có phải là mô hình của  $G$  hay không?*

# Ví dụ 4.1. Biểu diễn đơn giản (2)



$$\begin{array}{llll}
 \text{IR} & = & \{c, g, h, z, l, m, 1 \text{ lb}\} & \text{I}_S = \text{ex:Chutney} & \mapsto c \\
 \text{IP} & = & \{h, z, m\} & \text{ex:greenMango} & \mapsto g \\
 \text{LV} & = & \{1 \text{ lb}\} & \text{ex:hasIngredient} & \mapsto h \\
 \text{I}_{\text{EXT}} & = & h \mapsto \{\langle c, l \rangle\} & \text{ex:ingredient} & \mapsto z \\
 & & z \mapsto \{\langle l, g \rangle\} & \text{ex:amount} & \mapsto m \\
 & & m \mapsto \{\langle l, 1 \text{ lb} \rangle\} & & 
 \end{array}$$

Trong ví dụ này  $\mathcal{I}_L$  là hàm rỗng

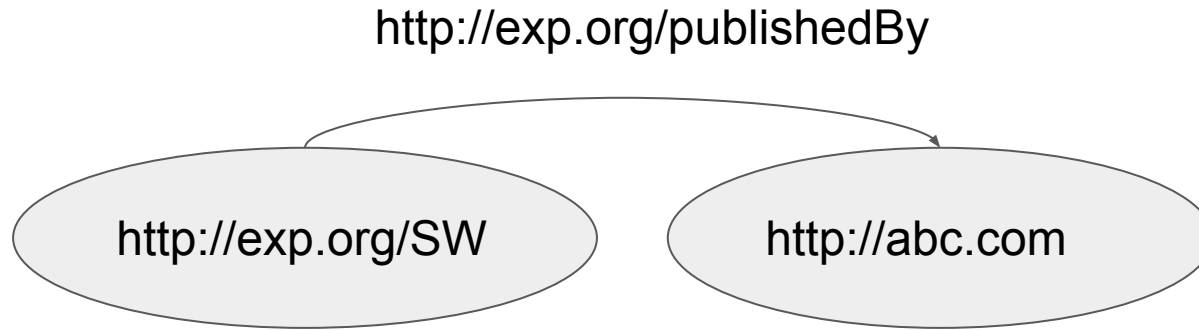
Nếu chọn A:  $\_:\text{id1} \mapsto l$  thì:

$$\begin{array}{llll}
 \langle \text{ex:Chutney}^{\mathcal{I}+A}, \_:\text{id1}^{\mathcal{I}+A} \rangle & = \langle c, l \rangle & \in \text{I}_{\text{EXT}}(h) & = \text{I}_{\text{EXT}}(\text{ex:hasIngredient}^{\mathcal{I}+A}) \\
 \langle \_:\text{id1}^{\mathcal{I}+A}, \text{ex:greenMango}^{\mathcal{I}+A} \rangle & = \langle l, g \rangle & \in \text{I}_{\text{EXT}}(z) & = \text{I}_{\text{EXT}}(\text{ex:ingredient}^{\mathcal{I}+A}) \\
 \langle \_:\text{id1}^{\mathcal{I}+A}, "1 \text{ lb}"^{\mathcal{I}+A} \rangle & = \langle l, 1 \text{ lb} \rangle & \in \text{I}_{\text{EXT}}(m) & = \text{I}_{\text{EXT}}(\text{ex:amount}^{\mathcal{I}+A})
 \end{array}$$

**Kết luận  $\mathcal{I} + A$  là mô hình của G.**



## Ví dụ 4.2. Biểu diễn đơn giản



*Thiết lập 1 biểu diễn đơn giản  $\mathcal{I}$  sao cho  $\mathcal{I}$  là mô hình của đồ thị RDF đã cho?*

# Biểu diễn RDF

- Biểu diễn RDF là trường hợp riêng của biểu diễn đơn giản, trong đó các điều kiện bổ xung được thiết lập trên các URIs của bộ từ vựng RDF để thể hiện ngữ nghĩa của chúng.
  - Một biểu diễn RDF cho đồ thị có bộ từ vựng  $V$  là biểu diễn đơn giản với bộ từ vựng  $V \cup V_{\text{RDF}}$  thỏa mãn các điều kiện ràng buộc ngữ nghĩa cho các từ khóa RDF.

# Biểu diễn RDF: Các ràng buộc

1.  $x \in IP \iff \langle x, \text{rdf:Property}^{\mathcal{I}} \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ .

Đối với tất cả các thuộc tính của bộ ba, có thể suy ra nó là phần tử của lớp thuộc tính.

***!Ghi chú:***  $x$  là 1 thuộc tính khi và chỉ khi có bộ ba  $x \text{ rdf:type rdf:Property}$ .

○ Hệ quả trực tiếp là trong biểu diễn RDF thì  $IP \subseteq IR$ .

2. Nếu " $s$ "<sup>^^</sup>rdf:XMLLiteral nằm trong  $V$  và  $s$  là 1 hằng XML đúng quy cách thì:

- $\mathcal{I}_L("s"^{^^}\text{rdf:XMLLiteral})$  là giá trị XML của  $s$ ;
- $\mathcal{I}_L("s"^{^^}\text{rdf:XMLLiteral}) \in LV$ ;
- $\langle \mathcal{I}_L("s"^{^^}\text{rdf:XMLLiteral}), \text{rdf:XMLLiteral}^{\mathcal{I}} \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$

## Biểu diễn RDF: Các ràng buộc<sub>(2)</sub>

3. Nếu "s"^^rdf:XMLLiteral thuộc V và s là một hằng không đúng quy cách thì

- $\mathcal{I}_L("s"^^\text{rdf:XMLLiteral}) \notin LV$ ;
- $\langle \mathcal{I}_L("s"^^\text{rdf:XMLLiteral}), \text{rdf:XMLLiteral} \rangle \notin \mathcal{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$

*Không gian giá trị của kiểu rdf:XMLLiteral có đúng 1 giá trị (được gọi là giá trị XML) cho 1 chuỗi XML được định kiểu hợp lệ. RDF chỉ yêu cầu giá trị này không phải chuỗi XML hay giá trị của bất kỳ kiểu dữ liệu nào trong lược đồ XML hoặc chuỗi Unicode.*

# Biểu diễn RDF: Các ràng buộc<sub>(3)</sub>

- Yêu cầu bổ xung: Biểu diễn RDF phải là mô hình của các bộ ba tiên đề RDF:

<code>rdf:type</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:subject</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:predicate</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:object</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:first</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:rest</code>	<code>rdf:type</code>	<code>rdf:Property</code> .
<code>rdf:nil</code>	<code>rdf:type</code>	<code>rdf:List</code> .

# Biểu diễn RDFS: Các ràng buộc

Các biểu diễn RDFS là các biểu diễn RDF với các ràng buộc bổ xung cho các URIs của bộ từ vựng RDFS

<code>rdfs:domain</code>	<code>rdfs:range</code>	<code>rdfs:Resource</code>
<code>rdfs:Literal</code>	<code>rdfs:Datatype</code>	<code>rdfs:Class</code>
<code>rdfs:subClassOf</code>	<code>rdfs:subPropertyOf</code>	<code>rdfs:Container</code>
<code>rdfs:member</code>	<code>rdfs:ContainerMembershipProperty</code>	
<code>rdfs:comment</code>	<code>rdfs:seeAlso</code>	<code>rdfs:isDefinedBy</code>
<code>rdfs:label</code>		

Đề đảm bảo ngữ nghĩa của những URIs đó theo thiết kê.

## Biểu diễn RDFS: Các ràng buộc <sub>(2)</sub>

- Để giản lược các mô tả, cho một biểu diễn  $\mathcal{I}$ , chúng ta ký hiệu hàm  $\mathcal{I}_{\text{CEXT}}$  là ánh xạ các tài nguyên tới các tập tài nguyên:  $\mathcal{I}_{\text{CEXT}}(y)$  chứa đúng các phần tử  $x$ , sao cho  $\langle x, y \rangle$  nằm trong  $\mathcal{I}_{\text{EXT}}(\text{rdf:type}^{\mathcal{I}})$ . Chúng ta gọi  $\mathcal{I}_{\text{CEXT}}(y)$  là lớp mở rộng của  $y$ .
- Chúng ta ký hiệu  $\text{IC}$  là mở rộng của  $\text{rdfs:Class}$ , như vậy:  
 $\text{IC} = \mathcal{I}_{\text{CEXT}}(\text{rdfs:Class}^{\mathcal{I}})$ .

*!Lưu ý:*  $\mathcal{I}_{\text{CEXT}}$  và  $\text{IC}$  được xác định đầy đủ bởi  $\cdot^{\mathcal{I}}$  và  $\mathcal{I}_{\text{EXT}}$ .

# Biểu diễn RDFS: Các ràng buộc <sup>(3)</sup>

Một biểu diễn RDFS cho 1 bộ từ vựng  $V$  là 1 biểu diễn RDF cho bộ từ vựng  $V \cup V_{\text{RDFS}}$  và thỏa mãn các điều kiện sau:

- $\text{IR} = \mathcal{I}_{\text{CEXT}}(\text{rdfs:Resource}^{\mathcal{I}})$  - Tất cả tài nguyên đều có kiểu `rdfs:Resource`;
- $\text{LV} = \mathcal{I}_{\text{CEXT}}(\text{rdfs:Literal}^{\mathcal{I}})$  - Tất cả hằng không định kiểu và hằng có định kiểu đúng cú pháp đều có kiểu `rdfs:Literal`;
- Nếu  $\langle x, y \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:domain}^{\mathcal{I}})$  và  $\langle u, v \rangle \in \mathcal{I}_{\text{EXT}}(x)$ , thì  $u \in \mathcal{I}_{\text{CEXT}}(y)$ .
  - Nếu thuộc tính `rdfs:domain` kết nối  $x$  với  $y$  và thuộc tính  $x$  kết nối  $u$  với  $v$ , thì  $u$  thuộc kiểu  $y$ .



# Biểu diễn RDFS: Các ràng buộc <sub>(4)</sub>

- Nếu  $\langle x, y \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:range}^{\mathcal{I}})$  và  $\langle u, v \rangle \in \mathcal{I}_{\text{EXT}}(x)$ , thì  $v \in \mathcal{I}_{\text{CEXT}}(y)$ .
  - Nếu thuộc tính `rdfs:range` kết nối  $x$  với  $y$  và thuộc tính  $x$  kết nối tài nguyên  $u$  và  $v$ , thì kiểu của  $v$  là  $y$ .
- $\mathcal{I}_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$  có tính phản xạ và bắc cầu trên IP.
  - Nếu  $p$  a `rdf:Property` thì  $p$  `rdfs:subPropertyOf`  $p$ .
  - Nếu  $x$  `rdfs:subPropertyOf`  $y$  và  $y$  `rdfs:subPropertyOf`  $z$  thì  $x$  `rdfs:subPropertyOf`  $z$ .

# Biểu diễn RDFS: Các ràng buộc<sub>(5)</sub>

- Nếu  $\langle x, y \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$ , thì  $x, y \in \text{IP}$  và  $\mathcal{I}_{\text{EXT}}(x) \subseteq \mathcal{I}_{\text{EXT}}(y)$ .
  - Nếu  $\text{rdfs:subPropertyOf}$  kết nối  $x$  với  $y$ , thì cả  $x$  và  $y$  là cặp thuộc tính mà tất cả các cặp tài nguyên nằm trong mở rộng của  $x$  cũng nằm trong mở rộng của  $y$ .
- Nếu  $x \in \text{IC}$ , thì  $\langle x, \text{rdfs:Resource}^{\mathcal{I}} \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ .
  - Nếu  $x$  biểu diễn một lớp, thì nó phải là lớp con của lớp  $\text{rdfs:Resource}$ ; Cặp  $x$  và  $\text{rdfs:Resource}$  phải nằm trong mở rộng của  $\text{rdfs:subClassOf}$ .

# Biểu diễn RDFS: Các ràng buộc<sub>(6)</sub>

- Nếu  $\langle x, y \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ , thì  $x, y \in \text{IC}$  và  $\mathcal{I}_{\text{CEXT}}(x) \subseteq \mathcal{I}_{\text{CEXT}}(y)$ .
  - Nếu  $x \text{ rdfs:subClassOf } y$ , thì cả  $x$  và  $y$  đều là các lớp, và mở rộng của  $x$  là tập con của mở rộng của  $y$ .
- $\mathcal{I}_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$  có tính chất phản xạ và bắc cầu trên IC.
  - Nếu  $C \text{ a rdfs:Class}$ , thì  $C \text{ rdfs:subClassOf } C$ .
  - Nếu  $C \text{ rdfs:subClassOf } D$  . và  $D \text{ rdfs:subClassOf } E$ , thì  $C \text{ rdfs:subClassOf } E$ .

# Biểu diễn RDFS: Các ràng buộc<sub>(7)</sub>

- Nếu  $x \in \mathcal{I}_{\text{CEXT}}(\text{rdfs:ContainerMembershipProperty}^{\mathcal{I}})$ , thì  $\langle x, \text{rdfs:member}^{\mathcal{I}} \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:subPropertyOf}^{\mathcal{I}})$ 
  - Nếu  $x$  là một thuộc tính có kiểu `rdfs:ContainerMembershipProperty` thì nó là thuộc tính con của `(rdfs:subPropertyOf) rdfs:member`.
- Nếu  $x \in \mathcal{I}_{\text{CEXT}}(\text{rdfs:Datatype}^{\mathcal{I}})$  thì  $\langle x, \text{rdfs:Literal}^{\mathcal{I}} \rangle \in \mathcal{I}_{\text{EXT}}(\text{rdfs:subClassOf}^{\mathcal{I}})$ .
  - Nếu  $x$  là phần tử của lớp `rdfs:Datatype`, thì nó phải là lớp con của lớp `rdfs:Literal` (lớp của tất cả các hằng giá trị).

# Biểu diễn RDFS: Các ràng buộc<sub>(8)</sub>

*Ngoài ra các bộ ba tiên đề trong RDFS cũng phải được đáp ứng*

<code>rdf:type</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:domain</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:range</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:subPropertyOf</code>	<code>rdfs:domain</code>	<code>rdf:Property</code> .
<code>rdfs:subClassOf</code>	<code>rdfs:domain</code>	<code>rdfs:Class</code> .
<code>rdf:subject</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .
<code>rdf:predicate</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .

# Biểu diễn RDFS: Các ràng buộc<sub>(9)</sub>

<code>rdf:object</code>	<code>rdfs:domain</code>	<code>rdf:Statement</code> .
<code>rdfs:member</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdf:first</code>	<code>rdfs:domain</code>	<code>rdf:List</code> .
<code>rdf:rest</code>	<code>rdfs:domain</code>	<code>rdf:List</code> .
<code>rdfs:seeAlso</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .
<code>rdfs:isDefinedBy</code>	<code>rdfs:domain</code>	<code>rdfs:Resource</code> .

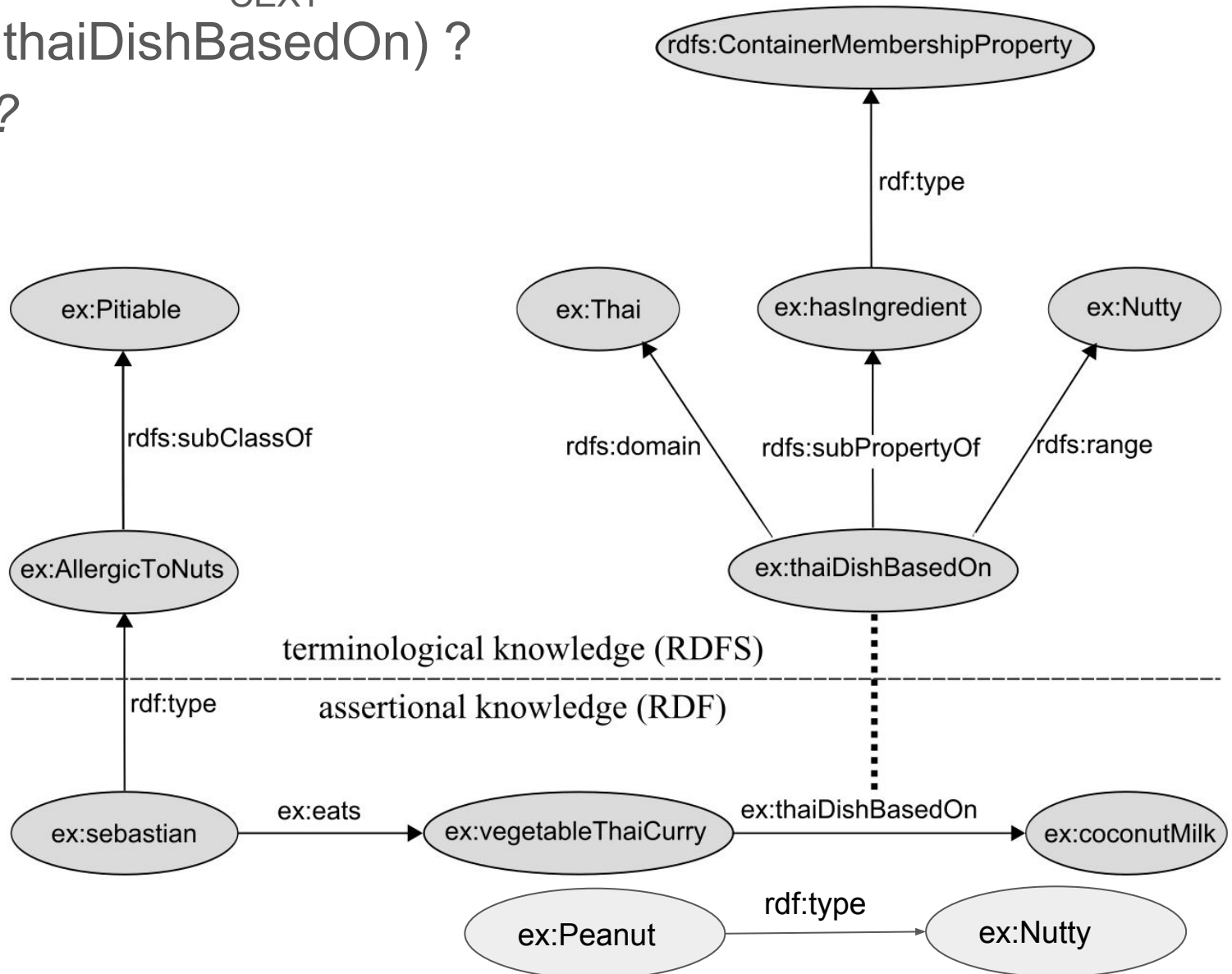
[ [FoSWT] ]

1) Trong phạm vi ngữ nghĩa RDFS hãy xác định các tập:

$\mathcal{I}_{\text{CEXT}}(\text{ex:Nutty})$ ,  $\mathcal{I}_{\text{CEXT}}(\text{ex:Pitiable})$ ,

$\mathcal{I}_{\text{EXT}}(\text{ex:thaiDishBasedOn})$  ?

2)  $IC = ?$



# Chuyển các mô tả sau thành 1 CSTT RDF(S)

1. Mèo, Vẹt, Chó là thú cưng.
  - a. Sử dụng các tên Cat, Parrot, Dog, Pet.
2. Sở hữu là 1 thuộc tính có miền là chủ sở hữu và khoảng là thú cưng
3. Chủ sở hữu là người
  - a. Sử dụng các tên Owner, Human, own
4. Sinh ra là 1 thuộc tính có miền và khoảng đều là thú cưng
  - a. Sử dụng tên offspring.
5. Sở hữu là thuộc tính con của thuộc tính chịu trách nhiệm cho.
  - a. Sử dụng tên responsibleFor
6. Mực là 1 con chó, sinh ra Mun. NVA sở hữu Mực.
7. Tìm 3 bộ 3 có thể được suy diễn dựa trên 1-6.

Thiết lập 1 biểu diễn đơn giản cho đồ thị RDF thu được theo 1-6?



# Nội dung

4.1. Vai trò của ngữ nghĩa hình thức

4.2. Ngữ nghĩa lô-gic của RDF(S)

4.3. Suy diễn dựa trên luật

4.4. Các hạn chế của RDF(S)



# Biểu diễn đơn giản: Luật suy diễn

- Luật suy diễn cho phép chuyển đổi ngữ nghĩa thành mô hình tính toán
- Luật suy diễn cho các biểu diễn đơn giản

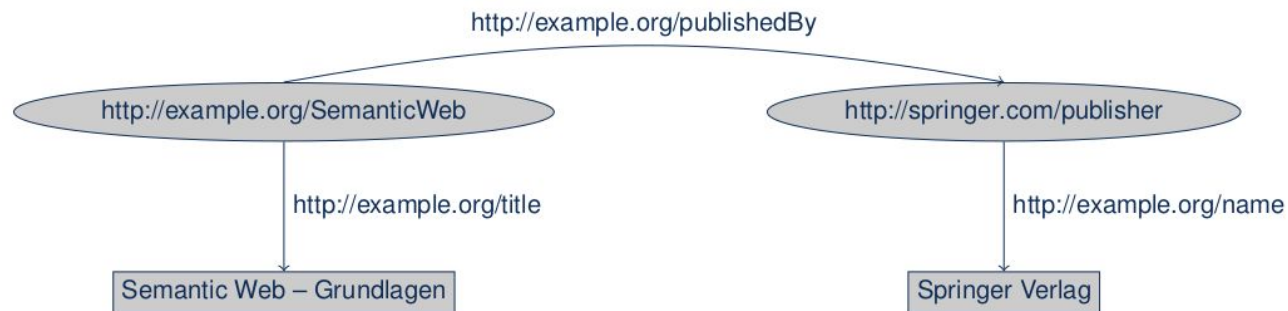
$$\begin{array}{ccc} u & a & x \cdot \\ \hline u & a & \_ : n \cdot \end{array} \quad \text{se1}$$
$$\begin{array}{ccc} u & a & x \cdot \\ \hline \_ : n & a & x \cdot \end{array} \quad \text{se2}$$

Tiền điều kiện: các nút rỗng chưa được gắn với URI hoặc hằng giá trị khác.

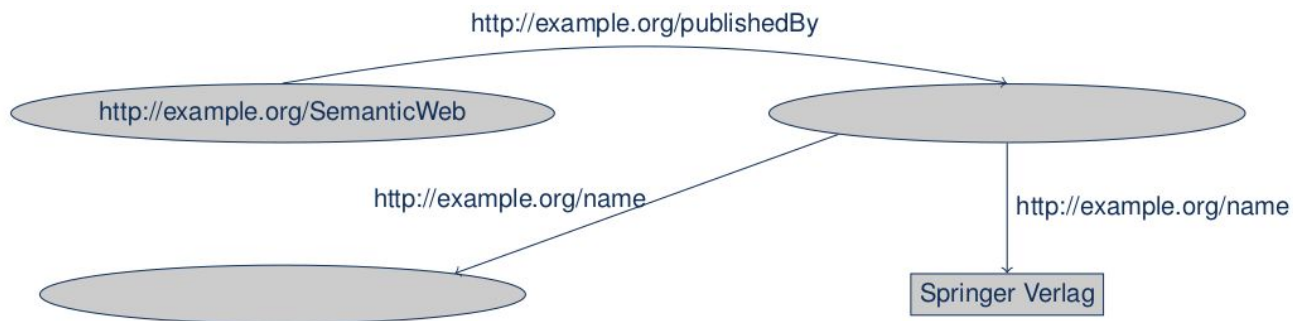
# Biểu diễn đơn giản: Mệnh đề

Một đồ thị  $G_2$  được suy ra từ  $G_1$  theo cách đơn giản nếu như  $G_1$  có thể được mở rộng thành  $G_1'$  bằng cách áp dụng các luật se1 và se2 sao cho  $G_2$  nằm trong  $G_1'$ .

*Ví dụ: Đồ thị*



*Trong phạm vi  
ngữ nghĩa đơn  
giản suy ra:*



# Biểu diễn RDF: Luật suy diễn

$$\frac{u \ a \ y}{a \ \text{rdf:type} \ \text{rdf:Property}} \quad \text{rdf1}$$

$l$  là một hằng,  $_:n$  là một nút tự do và nếu luật  $lg$  đã gắn  $+:n$  với hằng XML

$$\frac{u \ a \ l}{u \ a \ _:n} \quad lg$$

$$\frac{u \ a \ _:n}{_:n \ \text{rdf:type} \ \text{rdf:XMLLiteral}} \quad \text{rdf2}$$

$$\frac{}{u \ a \ x} \quad \text{rdfax} \quad \text{Các bộ ba tiên đề "u a x" luôn có thể được suy ra}$$

# Mệnh đề RDF

Mệnh đề: Đồ thị  $G_2$  được suy ra từ đồ thị  $G_1$  trong giới hạn ngữ nghĩa RDF, nếu tồn tại một đồ thị  $G_1'$  sao cho:

- $G_1'$  có thể được suy ra từ  $G_1$  thông qua lg, rdf1, rdf2 và rdfa và
- $G_2$  có thể được suy ra từ  $G_1'$  trong giới hạn ngữ nghĩa đơn giản

*Tiến trình suy diễn hai pha*

# Biểu diễn RDFS: Luật suy diễn

$$\frac{}{u \ a \ x \ .} \text{ rdfsax}$$

Các bộ ba tiên đề "u a x" đều có thể được suy diễn

$$\frac{u \ a \ \_ : n \ .}{u \ a \ l \ .} \text{ gl}$$

Luật suy diễn theo chiều ngược lại so với lg;  $\_ : n$  và  $l$  đã được gắn kết theo luật lg.

$$\frac{u \ a \ l \ .}{\_ : n \ \text{rdf:type} \ \text{rdfs:Literal}} \text{ rdfs1}$$

$\_ : n$  và  $l$  đã được gắn kết theo luật lg.

$$\frac{a \ \text{rdfs:domain} \ x \ . \quad u \ a \ y \ .}{u \ \text{rdf:type} \ x \ .} \text{ rdfs2}$$

Ràng buộc miền cho thuộc tính.

$$\frac{a \ \text{rdfs:range} \ x \ . \quad u \ a \ v \ .}{v \ \text{rdf:type} \ x \ .} \text{ rdfs3}$$

Ràng buộc khoảng cho thuộc tính.

# Biểu diễn RDFS: Luật suy diễn<sup>(2)</sup>

$$\frac{u \ a \ x \ .}{u \ \text{rdf:type} \ \text{rdfs:Resource} \ .} \quad \text{rdfs4a}$$

Chủ ngữ của bộ ba là tài nguyên

$$\frac{u \ a \ v \ .}{v \ \text{rdf:type} \ \text{rdfs:Resource} \ .} \quad \text{rdfs4b}$$

Giá trị thuộc tính không phải hằng giá trị là tài nguyên.

$$\frac{u \ \text{rdfs:subPropertyOf} \ v \ . \ v \ \text{rdfs:subPropertyOf} \ x \ .}{u \ \text{rdfs:subPropertyOf} \ x \ .} \quad \text{rdfs5}$$

Tính chất bắc cầu của `rdfs:subPropertyOf`

$$\frac{u \ \text{rdf:type} \ \text{rdf:Property} \ .}{u \ \text{rdfs:subPropertyOf} \ u \ .} \quad \text{rdfs6}$$

Tính chất phản xạ của `rdfs:subPropertyOf`

$$\frac{a \ \text{rdfs:subPropertyOf} \ b \ . \quad u \ a \ y \ .}{u \ b \ y \ .} \quad \text{rdfs7}$$

Suy diễn phân tử theo ngữ nghĩa của `rdfs:subPropertyOf`

# Biểu diễn RDFS: Luật suy diễn<sup>(3)</sup>

$$\frac{u \text{ rdf:type rdfs:Class .}}{u \text{ rdf:subClassOf rdfs:Resource .}} \text{ rdfs8}$$

Lớp chỉ chứa các tài nguyên.

$$\frac{u \text{ rdfs:subClassOf } x . \quad v \text{ rdf:type } u .}{v \text{ rdf:type } x .} \text{ rdfs9}$$

Suy diễn phần tử theo ngữ nghĩa của lớp con.

$$\frac{u \text{ rdf:type rdfs:Class .}}{u \text{ rdfs:subClassOf } u .} \text{ rdfs10}$$
 Tính chất phản xạ của  
rdfs:subClassOf
$$\frac{u \text{ rdfs:subClassOf } v . \quad v \text{ rdfs:subClassOf } x .}{u \text{ rdfs:subClassOf } x .} \text{ rdfs11}$$

Tính chất bắc cầu

$$\frac{u \text{ rdf:type rdfs:ContainerMembershipProperty .}}{u \text{ rdfs:subPropertyOf rdfs:member .}} \text{ rdfs12}$$
$$\frac{u \text{ rdf:type rdfs:Datatype .}}{u \text{ rdfs:subClassOf rdfs:Literal .}} \text{ rdfs10}$$
 rdfs:Datatype là  
lớp con của  
rdfs:Literal



# Biểu diễn RDFS: Luật suy diễn<sub>(4)</sub>

- Định nghĩa quan trọng: Xung đột XML

ex:hasSmiley rdfs:range rdfs:Literal .

ex:evilRemark ex:hasSmiley ">:->^^rdf:XMLLiteral .

- Xảy ra khi một nút có kiểu rdfs:Literal được gán 1 hằng giá trị sai quy cách.

# Mệnh đề RDFS

**Mệnh đề:** Đồ thị  $G_2$  được suy ra từ  $G_1$  trong giới hạn ngữ nghĩa RDFS, nếu tồn tại một đồ thị  $G_1'$  thu được bằng cách áp dụng các luật lg, gl, rdfax, rdf1, rdf2, rdfs1 - rdfs3 và rdfsax cho  $G_1$ , sao cho:

- $G_2$  được suy ra từ  $G_1'$  bằng suy diễn đơn giản, hoặc
- $G_1'$  chứa một xung đột XML.

# Nội dung

4.1. Vai trò của ngữ nghĩa hình thức

4.2. Ngữ nghĩa lô-gic của RDF(S)

4.3. Suy diễn dựa trên luật

4.4. Các hạn chế của RDF(S)



# Hạn chế của RDF+RDFS

bio:Human rdfs:subClassOf bio:Animal.

bio:Cat rdfs:subClassOf bio:Animal.

bio:Dog rdfs:subClassOf bio:Animal.

bio:hasPet a rdfs:Property;

    rdfs:domain bio:Human;

    rdfs:range bio:Dog;

    rdfs:range bio:Cat.

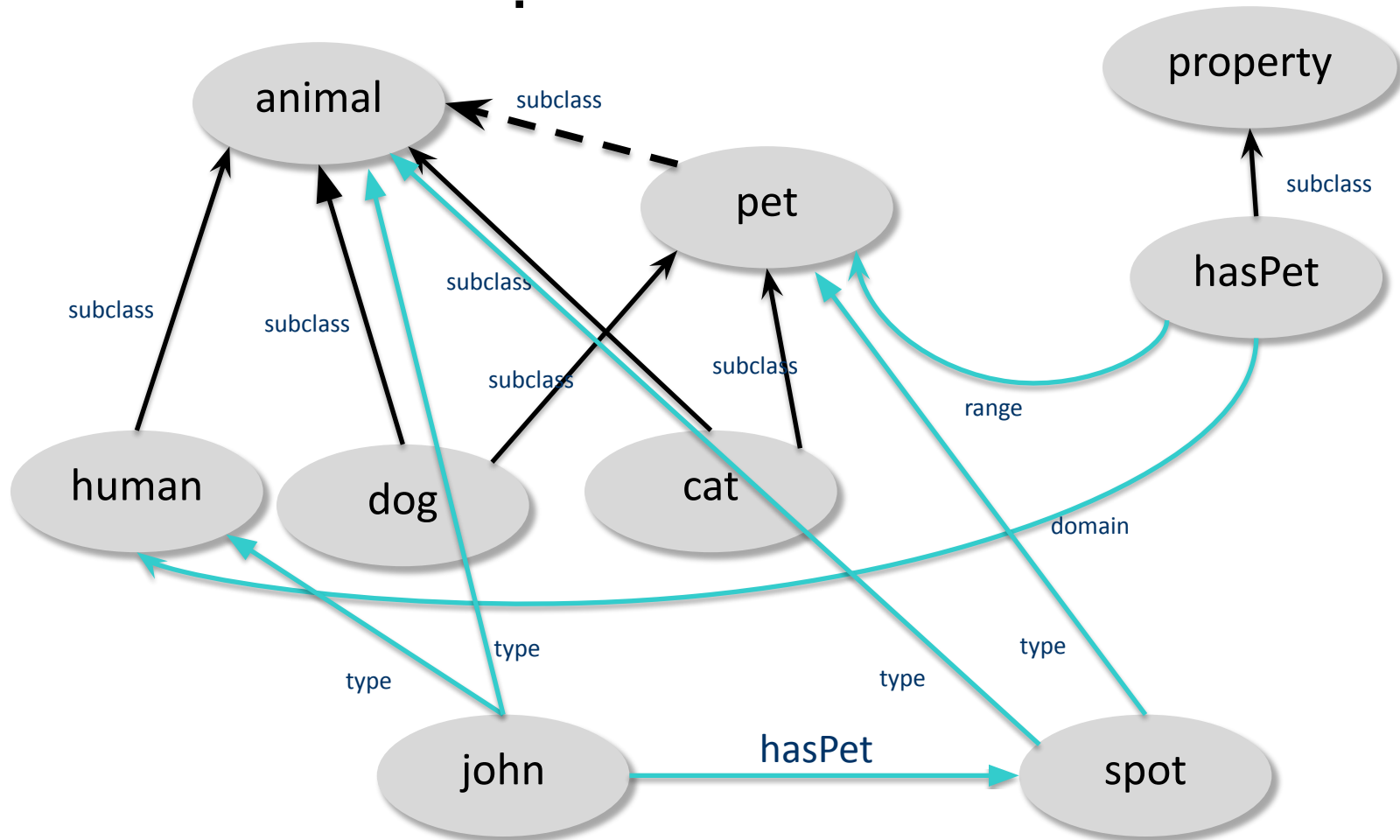
**:john bio:hasPet :spot =>**  
-----

**:john a bio:Human,  
          bio:Animal.**

**:spot a bio:Dog, bio:Cat,  
          bio:Animal.**

- Không thể thiết lập ràng buộc dog và cat là 2 lớp tách biệt
  - RDF+RDFS không thể bác bỏ các giả thuyết tương tự
  - :fido a bio:Dog . :milo a bio:Cat ; bio:offspring :fido .
    - Một con mèo (milo) sinh ra một con chó (fido) => Vô lý

# Biểu diễn đồ thị



**:john bio:hasPet :spot =>**  
:john a bio:Human, bio:Animal.  
:spot a bio:Pet, bio:Animal.

# Lớp và phần tử

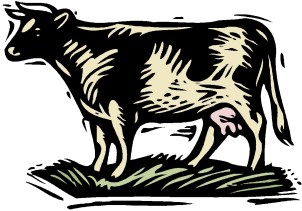
- Trong các ngôn ngữ HDT, lớp và đối tượng là hai khái niệm tách biệt, 1 thứ không thể vừa là lớp vừa là đối tượng
  - Nhiều hệ biểu diễn tri thức cũng áp dụng quy tắc này
- Trong RDFS một tài nguyên vừa có thể là lớp vừa có thể là đối tượng thuộc 1 lớp khác
  - Mô hình tập hợp

```
bio:Species rdf:type rdfs:Class.  
bio:Dog rdf:type rdfs:Species;  
           rdfs:subClassOf bio:Animal.
```

*Dog là phần tử của  
lớp Species và là lớp  
con của lớp Animal.*

# Ví dụ 4.3. Mô hình lớp dựa trên tập hợp

Không gian



Mô hình

Biểu diễn

Daisy isA Cow

Cow kindOf Animal

Mary isA Person

Person kindOf Animal

Z123ABC isA Car

Mary drives Z123ABC

$\Delta$

{... Danh sách các dữ kiện  
về các phần tử ...}

# Các hạn chế của RDFS

- Không có ràng buộc tồn tại/cơ sở
  - Không thể nói mỗi chiếc xe đạp có 2 bánh;
  - Mỗi người có đúng 2 phụ huynh; mỗi khóa học được dạy bởi tối thiểu 1 giảng viên.
- Không có các tính chất cho thuộc tính:
  - Tính chất bắc cầu (giống hasAncestor)
  - Thuộc tính duy nhất (giống hasMother)
  - Nghịch đảo của thuộc tính khác (eats và eatenBy).
  - v.v..
- Không diễn đạt được câu phủ định



# Các hạn chế của RDFS<sub>(2)</sub>

- Không thể giới hạn khoảng và miền với lớp chi tiết.
  - `rdfs:range` thiết lập phạm vi cho 1 thuộc tính (ví dụ, `eats`) đối với tất cả các phần tử của 1 lớp
  - Tuy nhiên không thể khai báo giới hạn phạm vi áp dụng cho 1 phần, Ví dụ:
  - Không thể mô tả `hasChild` có khoảng giá trị là voi khi áp dụng cho voi và có khoảng là mèo khi áp dụng cho mèo
  - Hoặc giới hạn miền của `eat` khi áp dụng cho `animal` là `living_thing`, còn khi áp dụng cho `cow` là `plant`.
    - `:eat rdfs:domain :animal; range :living_thing`
    - `:eat rdfs:domain :cow; range :plant`
  - V.V..

# Hạn chế của RDFS<sub>(3)</sub>

- Các lớp không giao nhau
  - Đôi khi chúng ta muốn nói rằng các lớp không giao nhau (ví dụ, male và female)
- Tổ hợp Boolean của các lớp
  - Chúng ta có thể muốn định nghĩa các lớp mới bằng cách kết hợp các lớp khác thông qua union, intersection, và complement
  - Ví dụ, person bằng hợp của male và female
  - Ví dụ, weekdays bằng tập hợp { :Mondy, ..., :Sunday }

*Các vấn đề đã nêu có thể được giải quyết trong OWL*

