

CƠ SỞ LOGIC ĐẶC TẢ CỦA OWL

Nhắc lại

- OWL dựa trên logic đặc tả
 - Với ngữ nghĩa hình thức ở dạng frame
- OWL tương thích với các đặc tính của RDF(S)
- 3 dạng:
 - OWL Lite
 - Ngôn ngữ ontology đơn giản với cơ chế suy diễn hiệu quả
 - Không có dữ liệu ở dạng lớp (class)
 - OWL DL
 - Ngôn ngữ ontology với cơ chế suy diễn mạnh hơn
 - Cho phép tham chiếu đến các kiểu dữ liệu, các URI, bao gồm cả tên từ RDF, RDFS và kiểu dữ liệu lược đồ XML.
 - OWL full
 - Ngôn ngữ ontology mạnh nhất
 - Không có sự kế thừa

2

Hệ CSTT

- Các hệ thống dựa trên các phương pháp và kỹ thuật của TTNT
- Để thao tác với CSTT
 - Biểu diễn CSTT dựa trên logic hoặc không
- Các hệ thống không dựa trên logic sử dụng các cấu trúc dữ liệu để thao tác với tri thức
 - Các hệ thống Frame
 - Mạng ngữ nghĩa
- Các hệ thống dựa trên logic dựa trên biến thể của logic vị từ cấp 1 (first-order logic) hoặc logic khác để thao tác
 - Định nghĩa ngữ nghĩa tốt để suy luận

3

First Order Logic

Extracts from slides of Bruijn

- ▶ “Classical” logic
 - ▶ Based on propositional logic (Aristotle, ~300 BC)
 - ▶ Developed in 19th century (Frege, 1879)
- ▶ Semi-decidable logic
 - ▶ Enumerate all true sentences
 - ▶ If a sentence is false, the algorithm might not terminate
- ▶ FOL is the basis for
 - ▶ Logic Programming: Horn Logic
 - ▶ Description Logics: 2-variable fragment
- ▶ A logic for describing *object*, *functions* and *relations*
 - ▶ *Objects* are “things” in the world: persons, cars, etc.
 - ▶ *Functions* take a number of objects as argument and “return” an object, depending on the arguments: addition, father-of, etc.
 - ▶ *Relations* hold between objects: distance, marriage, etc.
 - ▶ Often, a function can also be modeled as a relation

4

FOL syntax (1)

Constants	$a, b, john, \dots$
Function symbols	$f, g, +, married - to, \dots$
Predicate Symbols	$p, q, >, marriage, \dots$
Variables	x, y, \dots
Connectives	$\neg, \wedge, \vee, \leftarrow, \rightarrow, \leftrightarrow$
Quantifiers	\forall, \exists
(Equality)	$=$

Every constant is a term

- ▶ $a, b, john$

Every variable is a term

- ▶ x, y

If f is an n -place function symbol and t_1, \dots, t_n are terms, then $f(t_1, \dots, t_n)$ is a term

- ▶ $f(x), f(a), f(g(a))$
- ▶ $fatherOf(john), marriedTo(mary)$

5

FOL syntax (2)

If p is an n -place predicate symbol and t_1, \dots, t_n are terms, then $p(t_1, \dots, t_n)$ is an atomic formula

- ▶ $p(x), q(f(a), y)$
- ▶ $marriage(fatherOf(john), mary, date(2005, 4, 6))$

If t_1, t_2 are terms, then $t_1 = t_2$ is an atomic formula

- ▶ $f(x) = a, marriedTo(mary) = fatherOf(john)$

Any atomic formula is a formula

If A, B are formulas and x_1, \dots, x_n are variables then:

- ▶ $\neg A$ is a formula
- ▶ $A \wedge B$ is a formula
- ▶ $A \vee B$ is a formula
- ▶ $A \leftarrow B$ is a formula
- ▶ $A \rightarrow B$ is a formula
- ▶ $A \leftrightarrow B$ is a formula
- ▶ $\forall x_1, \dots, x_n. A$ is a formula
- ▶ $\exists x_1, \dots, x_n. A$ is a formula

Examples:

- ▶ $\forall x, y, d. marriage(x, y, d) \rightarrow marriedTo(x) = y \wedge marriedTo(y) = x$
- ▶ $\forall x : number(x) \rightarrow \exists y. y > x$

6

FOL semantics (1)

- ▶ The meaning of a First-Order formula is assigned using an *interpretation*

- ▶ An interpretation \mathcal{I} consists of:

- ▶ Domain Δ : a set of objects
- ▶ A set of relations $R: \Delta \times \dots \times \Delta$
- ▶ A set of functions $F: \Delta \times \dots \times \Delta^n \mapsto \Delta$
- ▶ A mapping function \cdot which:
 - ▶ Maps constants to objects: $c^{\mathcal{I}} \in \Delta$
 - ▶ Maps predicate symbols to relations: $p^{\mathcal{I}} \subseteq \Delta^n$
 - ▶ Maps function symbols to functions: $f^{\mathcal{I}} \subseteq \Delta^n \rightarrow \Delta$

- ▶ An interpretation is a *model* of a formula A if it makes the formula *true*:

- ▶ $\mathcal{I} \models A$

7

FOL semantics (2)

$p(t_1, \dots, t_n)$ (atomic formula)	is true iff	$\langle t_1^{\mathcal{I}}, \dots, t_n^{\mathcal{I}} \rangle \in p^{\mathcal{I}}$
$\neg A$	is true iff	$A^{\mathcal{I}}$ is not true
$A \wedge B$	is true iff	$A^{\mathcal{I}}$ and $B^{\mathcal{I}}$ are true
$A \vee B$	is true iff	$A^{\mathcal{I}}$ or $B^{\mathcal{I}}$ is true (or both)
$A \rightarrow B$	is true iff	in every case where $A^{\mathcal{I}}$ is true, $B^{\mathcal{I}}$ is true

8

FOL semantics (3)

- ▶ We have not discussed semantics of variables
- ▶ Variables *have no semantics*
- ▶ What to do with variables?
- ▶ Assign values to variables using an assignment B
 - ▶ e.g., $\{x \mapsto a, y \mapsto \text{john}\}$
- ▶ An interpretation \mathcal{I} makes a formula A *true* under a variable assignment B :
 - ▶ $\mathcal{I} \models_B A$
- ▶ Quantifiers:
 - ▶ $\exists x.A$: there exists an assignment for x which makes A true
 - ▶ $\forall x.A$: for all possible assignments of x , A is true

9

Ví dụ

Biểu thức $A = \forall x.\text{man}(x) \rightarrow \text{person}(x)$

- Miền $\Delta = \{b, h, c, d, e\}$
- Cho phép dịch I sau:
 - $\text{bill}^I = b, \text{hillary}^I = h, \text{chelsea}^I = c$
 - $\text{person}^I = \{b, h, c\}$
 - $\text{man}^I = \{b\}$
- Theo phép dịch đó, biểu thức A có giá trị *true*, vì vậy, I là một mô hình của A
- Nếu chọn I' giống I nhưng $\text{man}^{I'} = \{b, d\}$, khi đó A có giá trị *false*, I' không là một mô hình của A

10

Logic mô tả - Description Logics

- Dựa trên khái niệm (concept) và vai trò (role)
 - Khái niệm là tập các đối tượng
 - Vai trò là các quan hệ nhị phân trên các đối tượng
- DL cho phép xây dựng các khái niệm và vai trò phức tạp từ những cái đơn giản hơn, bao gồm
 - Conjunction, disjunction, negation
 - Một số dạng của lượng từ
- Là tập con của First Order Logic

11

Các khái niệm cơ bản của DL

- **Concepts** (classes)
 - vd., Person, Doctor, Parent
- **Roles** (properties)
 - vd., hasChild, hasName, hasAncestor
- **Individuals** (objects)
 - vd., John, Mary, Italy
- **Constructors**
 - Để tạo các khái niệm: vd., cộng gộp, kết nối, tách, ...
 - Để tạo các vai trò (axiom): vd., đảo, đối xứng, ...

12

Ví dụ về cú pháp và ngữ nghĩa của DL

Semantics given by **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

Constructor	Syntax	Example	Semantics
atomic concept	A	Human	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	R	has-child	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
and for C, D concepts and R a role name			
conjunction	$C \sqcap D$	Human \sqcap Male	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	Doctor \sqcup Lawyer	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negation	$\neg C$	\neg Male	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
exists restr.	$\exists R.C$	\exists has-child.Male	$\{x \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
value restr.	$\forall R.C$	\forall has-child.Doctor	$\{x \mid \forall y. \langle x, y \rangle \in R^{\mathcal{I}} \implies y \in C^{\mathcal{I}}\}$

13

Các tầng ngôn ngữ trong OWL

- OWL DL
 - Có cách biểu diễn chuẩn trong DL
 - Tương đương với **SHOIN**
- OWL Lite
 - Là tập con “dễ cài đặt” của OWL DL
 - Tương đương với **SHIN**
- OWL Full
 - là hợp của OWL DL và RDFS
 - Ngữ nghĩa RDF mở rộng với các điều kiện ngữ nghĩa liên quan và các biểu diễn dưới dạng bộ ba

14

OWL Constructs

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	$\{\text{john}\} \sqcup \{\text{mary}\}$
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer
maxCardinality	$\leq nP$	≤ 1 hasChild
minCardinality	$\geq nP$	≥ 2 hasChild

15

OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	$\{\text{President.Bush}\} \equiv \{\text{G.W.Bush}\}$
differentFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	$\{\text{john}\} \sqsubseteq \neg \{\text{peter}\}$
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent $^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor $^+$ \sqsubseteq ancestor
functionalProperty	$\top \sqsubseteq \leq 1P$	$\top \sqsubseteq \leq 1$ hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leq 1P^-$	$\top \sqsubseteq \leq 1$ hasSSN $^-$

16

OWL and DL

- ▶ OWL Lite corresponds to the DL $SHIN(\mathbf{D})$
 - ▶ Named classes (A)
 - ▶ Named properties (P)
 - ▶ Individuals ($C(o)$)
 - ▶ Property values ($P(o, a)$)
 - ▶ Intersection ($C \sqcap D$)
 - ▶ Union(!) ($C \sqcup D$)
 - ▶ Negation(!) ($\neg C$)
 - ▶ Existential value restrictions ($\exists P.C$)
 - ▶ Universal value restrictions ($\forall P.C$)
 - ▶ Unqualified number restrictions ($\geq nP, \leq nP, = nP$)
- ▶ OWL DL corresponds to the DL $SHOIN(\mathbf{D})$
 - ▶ Property value ($\exists P.\{o\}$)
 - ▶ Enumeration ($\{o_1, \dots, o_n\}$)

17

OWL Species

- OWL Full for users who want maximum expressiveness and the syntactic freedom of RDF
 - E.g., Classes as instances
- Every legal OWL Lite ontology is a legal OWL DL ontology
- Every legal OWL DL ontology is a legal OWL Full ontology
- Every OWL (Lite, DL, Full) document is an RDF document, and every RDF document is an OWL Full document
- But only some RDF documents will be a legal OWL Lite or OWL DL document

18

Syntaxes of OWL

- Abstract syntax
 - Human readable
 - Not defined for OWL Full
- RDF/XML
 - Official exchange syntax (for the Semantic Web)
 - Harder for human reading

19

OWL Abstract syntax

- DL syntax
 - `associateProfessor \sqsubseteq academicStaffMember`
 - `fullProfessor \sqsubseteq academicStaffMember`
 - `fullProfessor \sqsubseteq \neg associateProfessor`
 - `facultyMember \equiv academicStaffMember`
- OWL Abstract syntax
 - `Class(associateProfessor partial academicStaffMember)`
 - `Class(fullProfessor partial academicStaffMember)`
 - `DisjointClasses (fullProfessor associateProfessor)`
 - `Class(facultyMember complete academicStaffMember)`

20

RDF/XML syntax

```
<owl:Class rdf:ID="associateProfessor">
  <rdfs:subClassOf
    rdf:resource="#academicStaffMember"/>
</owl:Class>
<owl:Class rdf:ID="fullProfessor">
  <rdfs:subClassOf
    rdf:resource="#academicStaffMember"/>
  <owl:disjointWith
    rdf:resource="#associateProfessor"/>
</owl:Class>
<owl:Class rdf:ID="facultyMember">
  <owl:equivalentClass
    rdf:resource="#academicStaffMember" />
</owl:Class>
```

21

More example

- In DL
 $\text{Wine} \sqsubseteq \text{PotableLiquid} \cap \forall \text{hasMaker.Winery}$
- In abstract syntax
Class (Wine partial intersectionOf (PotableLiquid
restriction (hasMaker allValuesFrom (Winery)))
))
- In RDF/XML

```
<owl:Class rdf:ID="Wine" rdf:parseType="Collection">
  <rdfs:subClassOf><owl:Class>
    <owl:intersectionOf>
      <owl:Class rdf:about="#PotableLiquid"/>
      <owl:Restriction>
        <owl:onProperty
          rdf:resource="#hasMaker"/>
        <owl:allValuesFrom
          rdf:resource="#Winery"/>
      </owl:Restriction>
    </owl:intersectionOf> </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

22

Bài tập

- "Tác giả" bao gồm tất cả những người viết sách hoặc một phần cuốn sách
- Tacgia, nguoi, viet, sach, motphancua.sach
- Tacgia = nguoi $\cap \forall \text{viet.}(\text{sach} \cup \exists \text{motphancua.sach})$
- Happy parent is a person all of whose children are either Doctors or have a child who is a Doctor
- {Doctor \sqsubseteq Person,
HappyParent \equiv Person $\cap \forall \text{hasChild.}(\text{Doctor} \cup \exists \text{hasChild.Doctor})$ }

23

Further reading

- Jos de Bruijn: Using Ontologies. Enabling Knowledge Sharing and Reuse on the Semantic Web. DERI Technical Report DERI-2003-10-29, 2003.
<http://www.deri.org/publications/techpapers/documents/DERI-TR-2003-10-29.pdf>
- OWL Guide: <http://www.w3.org/TR/owl-guide/>
- OWL Reference: <http://www.w3.org/TR/owl-ref/>
- OWL Abstract syntax and Semantics: <http://www.w3.org/TR/owl-semantics/>

24