Web ngữ nghĩa

Soạn bởi: Nguyễn Bá Ngọc

Chương 6

Hà Nội-2021

Chương 6.
Ngữ nghĩa hình thức của OWL

Nội dung

6.1. Tổng quan về vấn đề biểu diễn tri thức

6.2. Lô-gic mệnh đề và lô-gic bậc nhất

6.3. Lô-gic mô tả

6.4. Ngữ nghĩa lô-gic của OWL

6.5. Suy diễn tự động với OWL

Nội dung

6.1. Tổng quan về vấn đề biểu diễn tri thức

6.2. Lô-gic mệnh đề và lô-gic bậc nhất

6.3. Lô-gic mô tả

6.4. Ngữ nghĩa lô-gic của OWL

6.5. Suy diễn tự động với OWL

Biểu diễn tri thức

Mục đích cơ bản:

Phát triển các nền tảng hình thức để tạo biểu diễn về thế giới để có thể tính toán hiệu quả và sử dụng để xây dựng các ứng dụng thông minh

Các cách tiếp cận trực quan, ví dụ các cấu trúc mạng

Các cách tiếp cận dựa trên lô-gic, ví dụ lô-gic bậc nhất (FOL), lô-gic mô tả

Lịch sử biểu diễn tri thức nhận biết

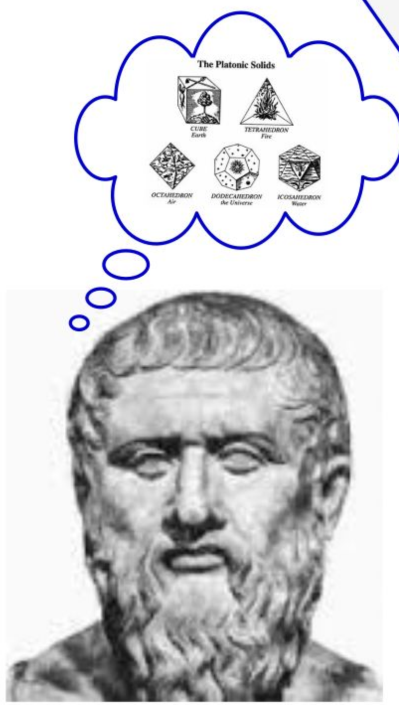
Plato: "***Knowledge** is justified true belief*"

Tri thức là niềm tin đã được xác minh đúng.

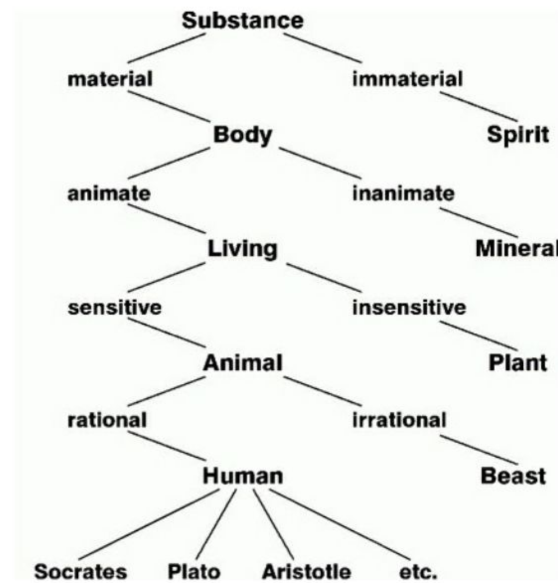
350 BC

Bên trong

Bên ngoài



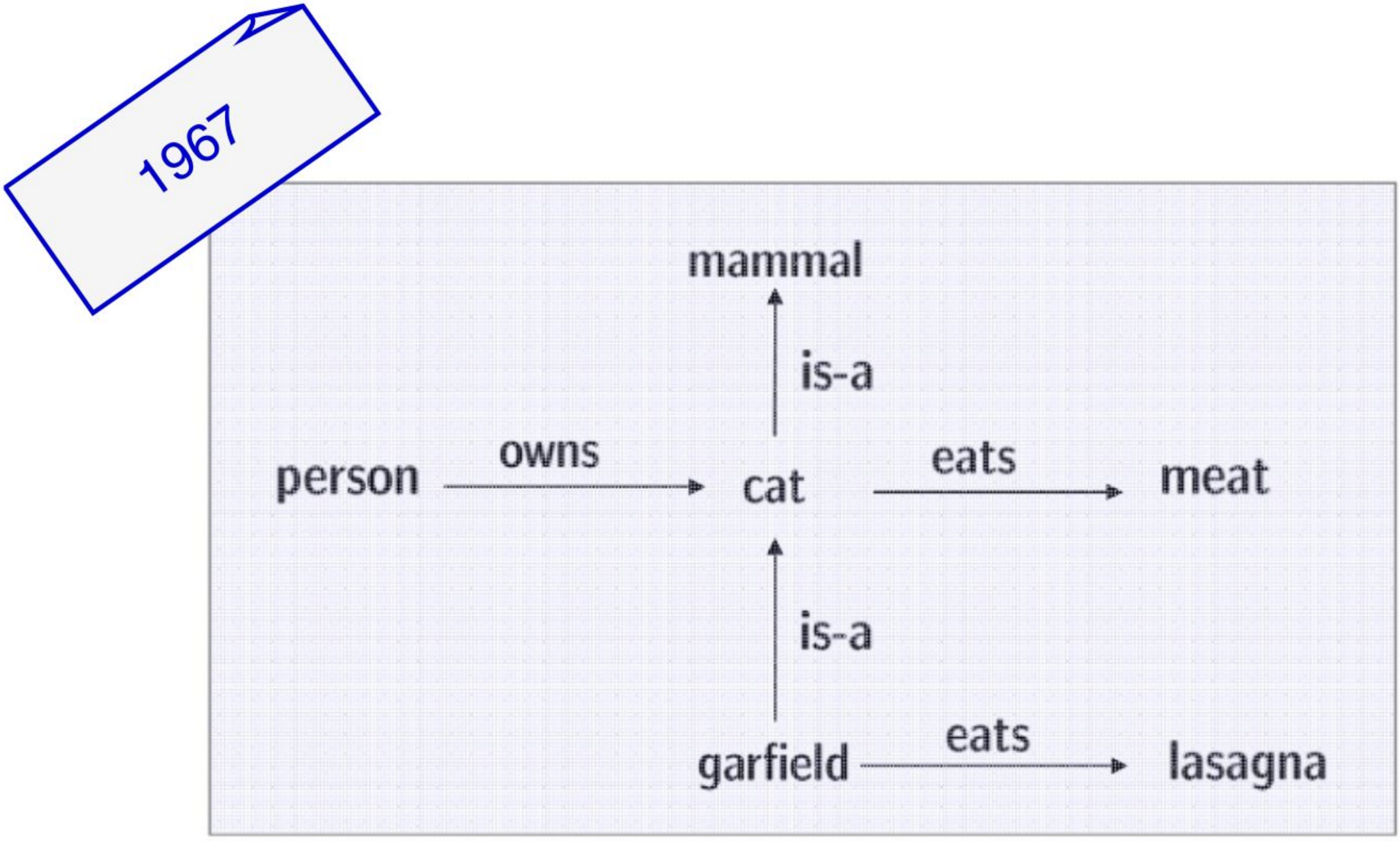
Thế giới quan



Cây tri thức

Lịch sử biểu diễn tri thức nhận biết(2)

Mạng ngữ nghĩa được giới thiệu trong [Quillan, 1967]



Lịch sử biểu diễn tri thức dựa trên lô-gic

- Những năm 1950: Biểu diễn tri thức bằng Lô-gic bậc nhất [McCarthy, 1959] - *(không khả quyết)*
- Những năm 1960: Biểu diễn tri thức bằng Các cấu trúc mạng [Quillan, 1967]. - *(không có ngữ nghĩa hình thức)*
- Những năm 1970: Biểu diễn cấu trúc mạng bằng FOL [Hayes, 1979].
- Những năm 1980:
  - Các khung lô-gic [Minsky, 1985]
  - Biểu diễn tri thức bằng lô-gic mô tả (DL)
    - Các phần khả quyết của FOL;
    - Các tri thức được biểu diễn bằng DLs cũng được gọi là ontology;
    - Nhiều phiên bản DLs với khả năng diễn đạt và tính toán khác nhau;
    - Thường được sử dụng để suy diễn khái niệm;

Nội dung

6.1. Tổng quan về vấn đề biểu diễn tri thức

6.2. Lô-gic mệnh đề và lô-gic bậc nhất

6.3. Lô-gic mô tả

6.4. Ngữ nghĩa lô-gic của OWL

6.5. Suy diễn tự động với OWL

Lô-gic mệnh đề và lô-gic bậc nhất

Thử biểu diễn những ý sau:

(1) Aristotle là người (2) Socrates là người

(3) Người không bất tử

=> Aristotle không bất tử và Socrates không bất tử.

- **Lô-gic mệnh đề (PL - Propositional Logic)**: Các biến mệnh đề, ¬, ∨, ∧, →
    - AristotleIsAMan = true;
    - SocratesIsAMan = true
    - AristotleIsAMan →AristotleIsMortal
    - SocratesIsAMan→SocratesIsMortal

Khả năng diễn đạt của PL tương đối hạn chế ...

Lô-gic mệnh đề và lô-gic bậc nhất$_{(2)}$

Thử biểu diễn những ý sau:

(1) Aristotle là người   (2) Socrates là người

(3) Người không bất tử

=> Aristotle không bất tử và Socrates không bất tử.

- **Lô-gic bậc nhất (FOL - First Order Logic):** Các thuộc tính, hằng, biến, hàm ký tự, ¬, ∨, ∧, ∀, ∃, →
  - Man(Socrates);
  - Man(Aristotel);
  - ∀X(Man(X)) → Mortal(X)

FOL có khả năng diễn đạt cao nhưng không khả quyết trong trường hợp tổng quát.

Cú pháp của FOL

- FOL cung cấp những thành phần sau:
  - Các biến: x, y, …
  - Các toán tử: Tương tự lô-gic mệnh đề (PL): phủ định (¬), và (⊓), hoặc (⊔), suy ra (=>), khi và chỉ khi (<=>)
  - Định lượng: Tất cả (∀) và tồn tại (∃)
- Người dùng tự định nghĩa các thành phần sau:
  - Các ký hiệu: "Các phần tử" trong thế giới;
    - Ví dụ, Mary, 3, …
  - Các hàm: Ánh xạ các phần tử tới các phần tử;
    - Ví du, father-of(Mary) = John, color-of(Sky) = Blue
  - Các mệnh đề: Ánh xạ các thành phần tới các giá trị chân lý;
    - Ví dụ: greater(5, 3), green(Grass), color(Grass, Green)

Các định lượng

- Định lượng tất cả
  - Tương tự liên kết và ("and") cho tất cả thành phần
  - $(\forall x)P(x)$ - Với tất cả x mà P(x) đúng
    - Ví dụ, $(\forall x)$dolphin(x) => mammal(x)
- Định lượng tồn tại
  - Tương tự liên kết hoặc ("or")
  - $(\exists x)P(x)$ - Tồn tại x mà P(x) đúng
    - Ví dụ, $(\exists x)$ mammal(x) ⊓ lays-eggs(x)
- Định lượng tất cả thường được sử dụng với liên kết "suy ra" để thiết lập luật nếu-thì
  - Ví dụ, $(\forall x)$ IT6390-student(x) => hard-working(x)
    - "Tất cả các học viên IT6390 đều chăm học"

Các định lượng$_{(2)}$

- Định lượng tồn tại thường được sử dụng với liên kết "và" để xác định các tính chất
  - Ví dụ, (∃x) IT6390-student(x) ⊓ female(x)
    - "Có nữ học viên IT6390"
- Có thể đảo trật tự các định lượng mà không làm thay đổi ý nghĩa
  - (∀x)(∀y)P(x, y) tương đương với (∀y)(∀x)P(x, y)
    - *(Tương tự cũng có thể đảo trật tự các toán tử tồn tại)*
  - (∀x)(∃y)likes(x, y) tương đương với (∃y)(∀x)likes(x, y)
    - Tất cả mọi người đều thích ai đó

Cú pháp FOL

- Các câu được xây dựng dựa trên từ và các thành phần:
  - Từ: Chỉ 1 đối tượng trong thế giới thực, 1 hằng ký tự, 1 tên biến, hoặc 1 hàm;
    - Ví dụ, left-leg-of(); x và $f(x_1, ..., x_n)$ là các từ, trong đó x_i là từ
  - Thành phần: Có giá trị đúng hoặc sai
    - Nếu P và Q là các thành phần, thì ¬P, P ⊓ Q, P ⊔ Q, P=>Q, P <=> Q cũng là các thành phần.
- Câu có thể là:
  - Thành phần, hoặc
  - Với P là câu và x là biến, thì (∀x)P và (∃x)P cũng là câu
- Điều kiện diễn đạt đúng quy cách
  - Câu không chứa biến tự do
    - Các biến đều được gắn với các định lượng
    - Ví dụ (∀x)P(x, y) - x được gắn với định lượng tất cả, còn y là biến tự do.

Phiên dịch NNTN sang FOL

- Tất cả người làm vườn đều thích mặt trời
  - Every gardener likes the sun
  - (∀x) gardener(x) => likes (x, Sun)
- Bạn luôn có thể đánh lừa ai đó
  - You can fool some of the people all of the time
  - (∃x)(∀t) (person(x) ⊓ time(t)) => can-fool(x, t)
- Đôi khi bạn có thể nói dối tất cả mọi người
  - (∀x)(∃t) (person(x) ⊓ time(t)) => can-fool(x, t)
- Tất cả nấm mầu tím đều có độc
  - (∀x) (mushroom(x) ⊓ purple(x)) => poisonous (x)

Phiên dịch NNTN sang FOL (2)

- Không có nấm tím nào có độc
  - No purple mushroom is poisonous
  - $\neg(\exists x)\ purple(x) \sqcap mushroom(x) \sqcap poisonous(x)$, hoặc:
  - $(\forall x)\ (mushroom(x) \sqcap purple(x)) \Rightarrow \neg poisonous(x)$
- Có đúng 2 nấm màu tím
  - There are exactly two purple mushrooms
  - $(\exists x)(\exists y)\ mushroom(x) \sqcap purple(x) \sqcap mushroom(y) \sqcap purple(y) \sqcap \neg(x=y) \sqcap (\forall z)\ (mushroom(z) \sqcap purple(z)) \Rightarrow (x=z) \sqcup (y=z)$
- NVA không cao:
  - NVA is not tall
  - $\neg tall(NVA)$

Phiên dịch NNTN sang FOL (3)

- X ở phía trên Y khi và chỉ khi X trực tiếp nằm trên Y hoặc có nhiều đối tượng được xếp chồng lên nhau bắt đầu với Y và kết thúc với X
  - $(\forall x)(\forall y)\ above(x, y) <=> (on(x, y) \sqcup (\exists z)\ (on(x, z) \sqcap above(z, y)))$

Suy diễn

- Suy diễn trong lô-gic hình thức là tiến trình sinh các câu hợp lệ từ những câu hợp lệ đang có (trong CSTT) bằng cách áp dụng các luật suy diễn
  - Mô-tơ suy diễn là đúng đắn nếu
    - Sinh ra những câu mới có thể được suy ra theo lô-gic từ CSTT
    - - Không chứng minh các hệ quả sai
  - Mô-tơ suy diễn là đầy đủ nếu
    - Có thể sinh tất cả các câu có thể suy ra theo lô-gic từ CSTT

Vắn tắt về tính khả quyết

Tính khả quyết

Một lớp vấn đề được gọi là khả quyết nếu tồn tại giải thuật cho bất kỳ vấn đề nào thuộc lớp này, với đầu vào bất kỳ giải thuật có thể kiểm tra "đúng" hoặc "sai" trong thời gian hữu hạn.

Giải thuật có tính dừng

Lô-gic khả quyết

Trong phạm vi lô-gic vấn đề khái quát sau được nghiên cứu

Đầu vào: Một tập phát biểu $\mathcal{T}$ và phát biểu ϕ

Đầu ra: "Đúng" nếu $\mathcal{T}$ suy ra ϕ, và "Sai" nếu ngược lại.

Trong trường hợp không có nhập nhằng với vấn đề được xử lý, tùy theo khả năng giải quyết vấn đề lô-gic còn được gọi là khả quyết hoặc không khả quyết.

Vắn tắt về tính khả quyết (2)

Các câu hỏi tương đương trong PL

- $\mathcal{T} \vDash \phi$?
- $\mathcal{T} \wedge \neg\phi$ không thỏa mãn/sai với tất cả các giá trị?

Vấn đề (không) được đáp ứng ((un)satisfiability) trong PL được gọi là (UN)SAT. Lô-gic mệnh đề là khả quyết, bởi vì (UN)SAT là khả quyết.

Nội dung

6.1. Tổng quan về vấn đề biểu diễn tri thức

6.2. Lô-gic mệnh đề và lô-gic bậc nhất

6.3. Lô-gic mô tả

6.4. Ngữ nghĩa lô-gic của OWL

6.5. Suy diễn tự động với OWL

Lô-gic mô tả: Khái quát

- Phần khả quyết của lô-gic bậc nhất
- Một phần nghiên cứu DL đáng kể hướng tới làm rõ chi phí tính toán cho các bài toán suy diễn trong trường hợp độ phức tạp cao nhất.
- Tuy có độ phức tạp cao nhưng DLs cũng có khả năng diễn đạt cao, vẫn có các giải thuật suy diễn được tối ưu hóa có hành vi tốt trong môi trường ứng dụng
  - Ví dụ, SAT solving: Thuộc lớp NP-khó nhưng hoạt động tốt trong thực tế.

Lô-gic mô tả: Khái quát$_{(2)}$

- Lô-gic mô tả (DLs) là nền tảng biểu diễn tri thức tiêu biểu hiện nay
- Đã ảnh hưởng đáng kể tới sự hình thành của các ngôn ngữ Web ngữ nghĩa
  - OWL chủ yếu dựa trên lô-gic mô tả
- Có nhiều mô-tơ suy diễn

Jena Mastro HermiT FaCT++ Pellet ELK

...

Các thành phần DL

- Các phần tử: john, mary, sun
  - ⇝ các hằng trong FOL, các tài nguyên trong RDF
- Các khái niệm: Person, Course, Student, Film, v.v.
  - ⇝ các thuộc tính 1 đối số (FOL), các lớp (RDFS)
- Các vai trò: hasFather, attends, worksWith, v.v.
  - ⇝ các thuộc tính 2 đối số (FOL), các thuộc tính (RDFS)
  - Có thể tiếp tục phân rã thành vai trò trừu tượng và vai trò cụ thể (thuộc tính đối tượng và thuộc tính dữ liệu - OWL).
- Mỗi ngôn ngữ DL quy định các cấu trúc để mô tả khái niệm, vai trò, và các phần tử.

Tập hợp tất cả các phần tử, khái niệm, và vai trò được gọi là chữ ký hoặc bộ từ vựng của CSTT.

Các thành phần của CSTT DL

TBox $\mathcal{T}$	Biểu diễn các khái niệm và mối quan hệ giữa các khái niệm
ABox $\mathcal{A}$	Biểu diễn các phần tử, gắn kết các phần tử với các khái niệm, mối quan hệ giữa các phần tử

Ngoài ra CSTT DLs còn có thể có:

RBox $\mathcal{R}$	Biểu diễn các vai trò và mối quan hệ giữa các vai trò

Các thành phần của CSTT DL$_{(2)}$

- Hộp khái niệm (TBox $\mathcal{T}$), ví dụ:
  - $\mathcal{T}$ = {Father ≡ Human ⊓ Male ⊓ ∃hasChild, HappyFather ⊑ Father ⊓ ∀hasChild.(Doctor ⊔ Lawyer)}
- Hộp vai trò (RBox $\mathcal{R}$), ví dụ
  - $\mathcal{R}$ = {hasFather ⊑ hasParent}
- Hộp dữ kiện (ABox $\mathcal{A}$), ví dụ
  - $\mathcal{A}$ = {HappyFather (john), hasChild (john, mary)}

Dựa trên ngữ nghĩa có thể suy diễn được gì trên CSTT?

- $\mathcal{T} \cup \mathcal{A} \models$ (Doctor ⊔ Lawyer)(mary)

Các ngôn ngữ DL

Ngôn ngữ thuộc tính với phần bù

$\mathcal{ALC}$ (Attributive Language with Complement) - ngôn ngữ DL cơ bản nhất thuộc lớp Boolean đóng.

Biểu thức khái niệm trong $\mathcal{ALC}$ được tạo lập như sau:

- Tên khái niệm đồng thời là biểu thức khái niệm;
- $\top$ và $\bot$ là các biểu thức khái niệm;
- C và D là các khái niệm, $\neg C$, $C \sqcap D$, and $C \sqcup D$ là các biểu thức khái niệm.
- Với 1 vai trò r và 1 khái niệm C, $\exists r.C$ và $\forall r.C$ là các biểu thức khái niệm.
    - Ví dụ: Student $\sqcup$ $\forall$attendsCourse.MasterCourse

Ngôn ngữ thuộc tính với phần bù$_{(2)}$

Mỗi khẳng định $\mathcal{ALC}$ có thể thuộc 1 trong các dạng sau:

- C(a), được gọi là khẳng định khái niệm
- r(a, b) - khẳng định vai trò
- ¬r(a, b) - phủ định vai trò
- a ≈ b - phát biểu tương đương
- a ≉ b - phát biểu không tương đương (khác nhau)

OWL DL và $\mathcal{ALC}$

Các cấu trúc OWL có thể được biểu diễn trong $\mathcal{ALC}$

- Lớp, thuộc tính, và phần tử
- Phần tử thuộc lớp và các quan hệ
- owl:Thing và owl:Nothing
- Bao hàm lớp, lớp tương đương, và lớp không giao nhau
- Giao, hợp và phần bù của lớp
- Giới hạn thuộc tính với owl:allValuesFrom và owl:someValuesFrom
- rdfs:domain và rdfs:range

Các cấu trúc khác có thể được thêm vào $\mathcal{ALC}$

Các mở rộng cho $\mathcal{ALC}$

($\mathcal{ALC}$)[$\mathcal{S}$][$\mathcal{H}$|$\mathcal{R}$][$\mathcal{O}$][$\mathcal{I}$][$\mathcal{F}$|$\mathcal{N}$| $\mathcal{Q}$]($\mathcal{D}$)

- $\mathcal{S}$ - Các tính chất của vai trò (bắc cầu, đối xứng, v.v..)
- $\mathcal{H}$ - Bao gồm vai trò, hình thành cây vai trò
- $\mathcal{O}$ - Các lớp đóng
- $\mathcal{I}$ - Các vai trò nghịch
- $\mathcal{N}$ - Giới hạn cơ số tùy ý
- $\mathcal{Q}$ - Giới hạn cơ số tùy ý có định kiểu
- $\mathcal{F}$ - Vai trò hàm
- $\mathcal{D}$ - Kiểu dữ liệu
- $\mathcal{R}$ - Bao gồm vai trò khái quát - bao gồm chuỗi vai trò

!Ghi chú: $\mathcal{F}$ có thể được diễn đạt qua $\mathcal{N}$; $\mathcal{N}$ (và $\mathcal{F}$) có thể được diễn đạt qua $\mathcal{Q}$

Cú pháp DL: Sơ lược

$\mathcal{ALC}$	Nguyên tố Phủ định Giao Hợp Tồn tại Tất cả Bao hàm Tương đương Thành viên Quan hệ Giống nhau Khác nhau	A, B $\neg C$ $C \sqcap D$ $C \sqcup D$ $\exists r.C$ $\forall r.C$ $C \sqsubseteq D$ $C \equiv D$ C(a) r(a, b) $a \approx b$ $a \neq b$
$\mathcal{N}$	Tối thiểu Tối đa	$\geq n\ r$ $\leq n\ r$

Cú pháp DL: Sơ lược$_{(2)}$

$\mathcal{N}$	*Tối thiểu* *Tối đa*	$\geq n\ r$ $\leq n\ r$
$\mathcal{Q}$	Tối thiểu Tối đa	≥n r.C ≤n r.C
$\mathcal{O}$	Lớp đóng	$\{i_1, i_2, ..., i_n\}$
$\mathcal{I}$	Nghịch đảo	r^-
$\mathcal{H}$	Bao hàm	$r \sqsubseteq s$
$\mathcal{S}$	Bắc cầu Bất đối xứng Phản xạ	Trans(r) Asy(r)* Ref(r)*
$\mathcal{R}$	Chuỗi vai trò	$r \circ r' \sqsubseteq s$

Mệnh đề bao gồm khái niệm

Mệnh đề bao gồm khái niệm (*GCI - General Concept Inclusion*) có dạng:

$C \sqsubseteq D$, trong đó C và D là các khái niệm

- $C \equiv D$ là viết tắt của $C \sqsubseteq D$ và $D \sqsubseteq C$

Mệnh đề bao gồm vai trò

- Một vai trò có thể là
  - Một tên vai trò r hoặc
  - Tên vai trò nghịch đảo r^-, hoặc
  - vai trò toàn thể **u**
- Mệnh đề bao gồm vai trò khái quát có dạng:
  - $r_1 \circ \ldots \circ r_n \sqsubseteq r$, trong đó $r_1, \ldots, r_n$, r là các vai trò
- $r \equiv s$ là viết tắt của $r \sqsubseteq s$ và $s \sqsubseteq r$

Ví dụ 6.1. Biểu thức khái niệm

- *Singer $\sqcap$ Actor - Những ca sĩ kiêm diễn viên*
- *Doctor $\sqcup$ Lawyer - Bác sĩ và luật sư*
- *$\forall$hasChild.Doctor - Người có tất cả con đều là bác sĩ*
- *$\exists$hasChild.Doctor - Người có con là bác sĩ*
- *$\neg$(Doctor $\sqcup$ Lawyer) - Người không phải bác sĩ hay luật sư*
- *($\geq$ 2 hasChild) $\sqcap$ ($\leq$ 1 sibling) - Người có từ 2 con trở lên và 1 anh em.*
- *($\geq$ 2 hasChild.Doctor) - Người có từ 2 con là bác sĩ trở lên*
- *$\forall$hasChild^{-}.Doctor - Người có song thân đều là bác sĩ*

Các phiên bản OWL và lô-gic mô tả

OWL Full	Rộng hơn lô-gic mô tả
OWL DL	$\mathcal{SHOIN}$
OWL Lite	$\mathcal{SHIF}$
OWL 2 Full	Rộng hơn lô-gic mô tả
OWL 2 DL	$\mathcal{SROIQ}$
OWL 2 EL	$\mathcal{EL}^{++}$ - Lược bỏ định lượng tất cả, thay vì định lượng tồn tại
OWL 2 QL	DL-Lite
OWL 2 RL	DLP (Description Logic Program)

Tính đơn của vai trò

- Dựa trên các RIAs các vai trò được chia thành đơn và phức
- Một vai trò là phức nếu nó xuất hiện trong rhs (vế phải) của 1 RIA phức
- Cụ thể:
  - Với 1 RIA bất kỳ $r_1 \circ r_2 \circ \ldots \circ r_n \sqsubseteq r$ với $n > 1$, r là 1 vai trò phức
  - Với 1 RIA bất kỳ $s \sqsubseteq r$ với s là 1 vai trò phức, thì r là 1 vai trò phức và
  - Tất cả các vai trò còn lại là các vai trò đơn

Ví dụ: $p \circ q \sqsubseteq r$ $\quad r \sqsubseteq s$ $\quad p \sqsubseteq r$ $\quad q \sqsubseteq s$

Thử: phân biệt các vai trò phức và vai trò đơn?

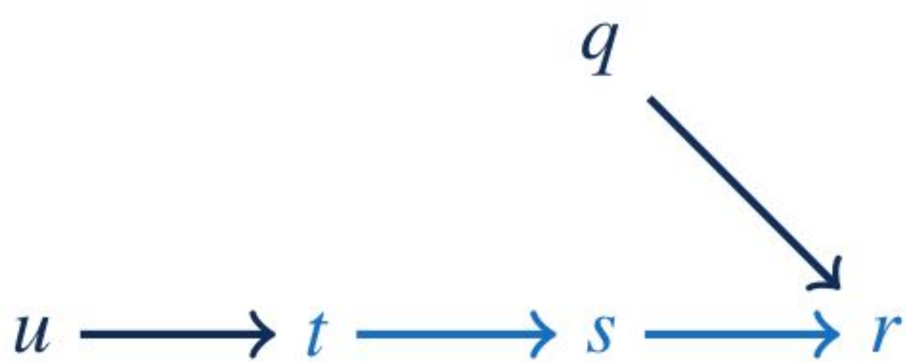
Tính đơn của vai trò$_{(2)}$

Cách khác để phân biệt vai trò đơn và phức:

- Các vai trò sau là vai trò phức:
  - Vai trò có tính chất bắc cầu Trans(s).
  - Vai trò có vai trò ngược là bắc cầu Trans(s^-)
  - Vai trò bao hàm 1 vai trò bắc cầu $t \sqsubseteq s$ và Trans(t)
  - Vai trò có vai trò ngược bao hàm 1 vai trò bắc cầu $t \sqsubseteq s^-$ và Trans(t)
- Tất cả các vai trò khác là các vai trò đơn.
- Ví dụ: $\mathcal{R} = \{u \sqsubseteq t, t \sqsubseteq s, s \sqsubseteq$

Phức: t, s, r

Đơn: q, u



Ví dụ 6.2. CSTT DL

RBox $\mathcal{R}$

$\text{own} \sqsubseteq \text{careFor}$

"Nếu ai đó sở hữu cái gì thì họ quan tâm về nó"

TBox $\mathcal{T}$

$\text{Healthy} \sqsubseteq \neg\text{Dead}$

$\text{Cat} \sqsubseteq \text{Dead} \sqcup \text{Alive}$

$\text{HappyCatOwner} \sqsubseteq \exists\text{owns.Cat} \sqcap \forall\text{caresFor.Healthy}$

"Sống khỏe mạnh là không chết"; " Mèo có thể đã chết hoặc còn sống"; "Chủ nhân hạnh phúc của mèo là người sở hữu mèo và tất cả mèo mà người đó quan tâm đều khỏe mạnh."

ABox $\mathcal{A}$

$\text{HappyCatOwner(NVA)}$

"NVA là người sở hữu mèo hạnh phúc."

Ngữ nghĩa hình thức của DL

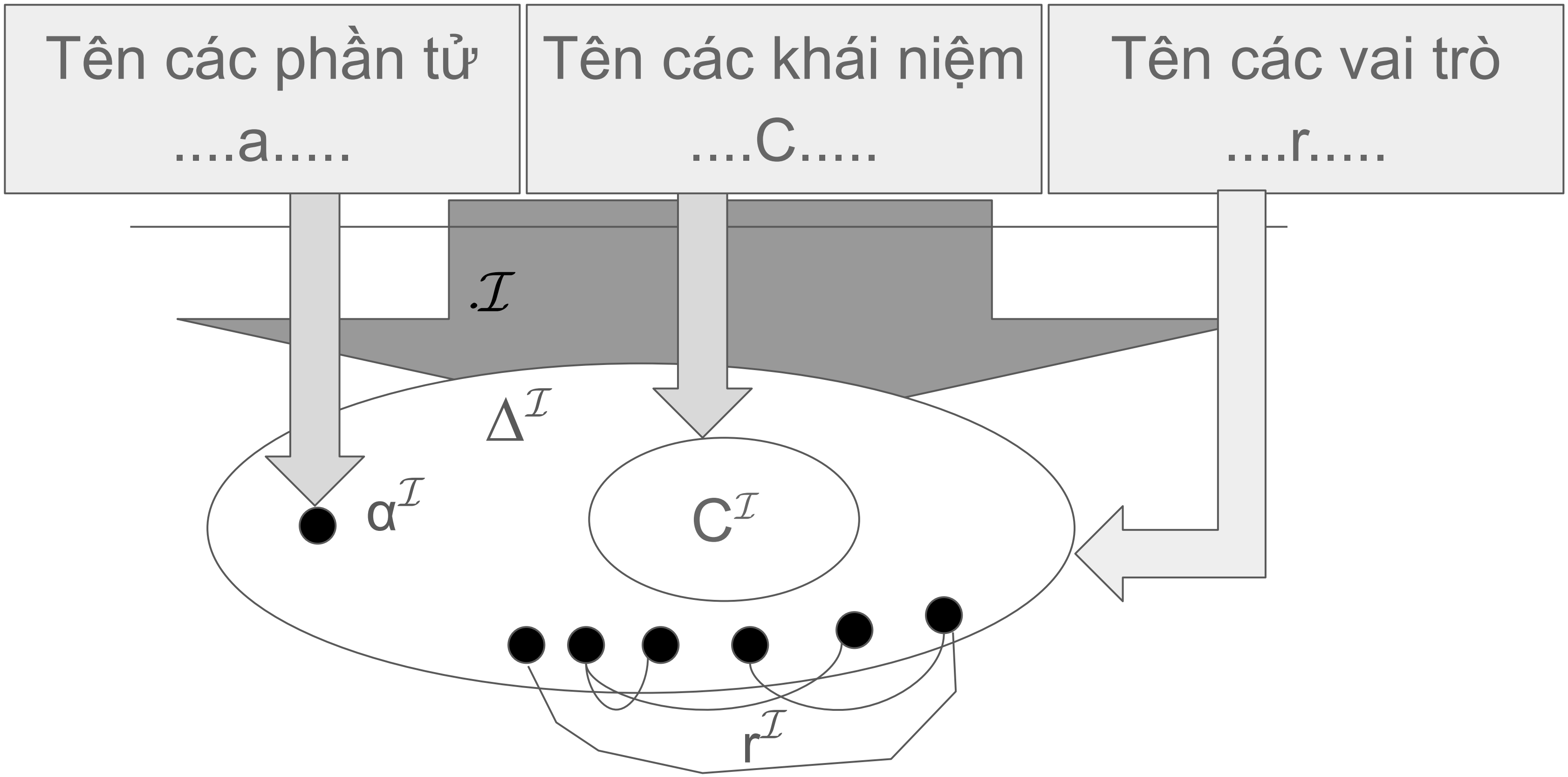
Các biểu diễn

- Ngữ nghĩa hình thức của DLs được xây dựng theo mô hình-lý thuyết dựa trên các không gian trừu tượng, được gọi là các biểu diễn.

Định nghĩa: Một biểu diễn $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ bao gồm:

- Một tập không rỗng $\Delta^{\mathcal{I}}$, được gọi là miền biểu diễn
- và hàm $.^{\mathcal{I}}$ ánh xạ
    - các tên riêng α tới các phần tử lĩnh vực $\alpha^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
    - tên khái niệm C tới tập con $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
    - tên vai trò r tới các tập con $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Sơ đồ của 1 biểu diễn



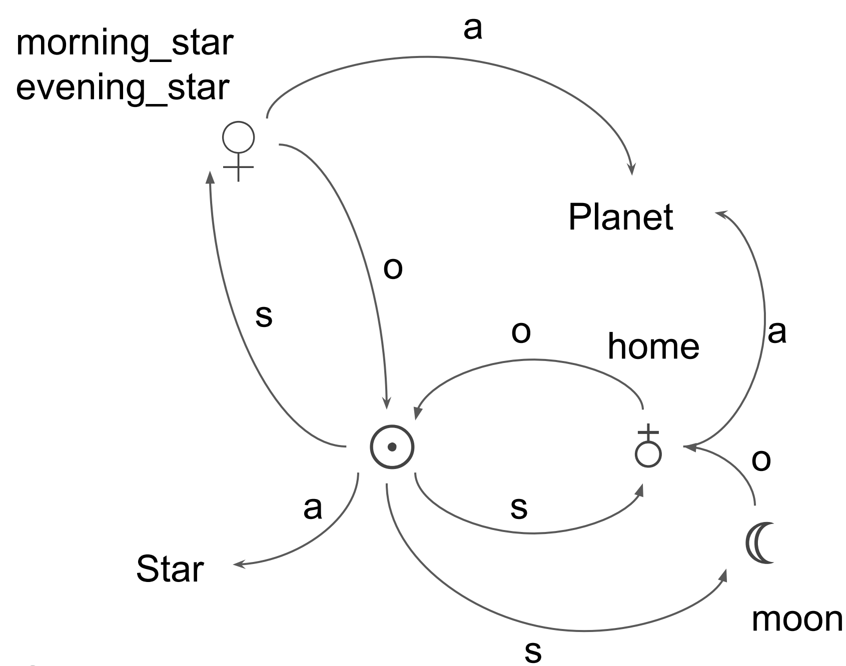
Ví dụ 6.3. Các biểu diễn

S_I = {sun, moring_star, evening_star, moon, home}.
S_C = {Planet, Star}
S_R = {orbitAround, shinesOn}

$\Delta^{\mathcal{I}}$ = {☉, ☿, ♀, ♁, ☾, ♂, ♃, ♄, ♅, ♆}

sun$^{\mathcal{I}}$ = ☉
morning_star$^{\mathcal{I}}$ = ♀
evening_star$^{\mathcal{I}}$ = ♀
moon$^{\mathcal{I}}$ = ☾
home$^{\mathcal{I}}$ = ♁

Planet$^{\mathcal{I}}$ = {☿, ♀, ♁, ♂, ♃, ♄, ♅, ♆}
Star$^{\mathcal{I}}$ = {☉}
orbitsAround$^{\mathcal{I}}$ = {<☿, ☉>, <♀, ☉>,<♁, ☉>, <♂, ☉>,
<♃, ☉>, <♄, ☉>, <♅, ☉>, <♆, ☉>, <☾, ♁>}
shinesOn$^{\mathcal{I}}$ = {<☉, ☿>, <☉, ♀>,<☉, ♁>, <☉, ♂>,
<☉, ♃>, <☉, ♄>, <☉, ♅>, <☉, ♆>,<☉, ☾>, <☉, ♁> }



Biểu diễn các phần tử

Giả thuyết tên duy nhất (UNA-Unique Name Assumption)

Với các định danh $c_1 \neq c_2$, thì $c^{\mathcal{I}}_1 \neq c^{\mathcal{I}}_2$

**Lưu ý:* Nếu UNA đúng thì các kiểm tra tương đương và khác biệt không còn cần thiết. Giả thuyết UNA có thể bị bác bỏ trong DLs.

Giả thuyết tên tiêu chuẩn (SNA-Standard Name Assumption)

Giả thuyết UNA đúng, và hơn nữa các phần tử được biểu diễn theo 1 cách trong tất cả các biểu diễn. Như vậy, chúng ta có thể giả sử $\Delta^{\mathcal{I}}$ chứa một tập các phần tử, và với mỗi biểu diễn $\mathcal{I}$, chúng ta có $c^{\mathcal{I}} = c$ (khi đó c được gọi là tên tiêu chuẩn).

Biểu diễn các biểu thức khái niệm

Tên	**Cú pháp**	**Ngữ nghĩa**
đỉnh - top	$\top$	$\Delta^{\mathcal{I}}$
đáy - bottom	$\bot$	$\varnothing$
phần bù - negation	$\neg C$	$\Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
giao - conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
hợp - disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
định lượng tất cả - universal quantifier	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}$ và $y \in C^{\mathcal{I}}\}$
định lượng tồn tại - existential quantifier	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid$ có $y \in \Delta^{\mathcal{I}}$ thỏa mãn $(x, y) \in r^{\mathcal{I}}$ và $y \in C^{\mathcal{I}}\}$

Biểu diễn các biểu thức khái niệm$_{(2)}$

Cấu trúc	**Cú pháp**	**Ví dụ**	**Ngữ nghĩa**
Giới hạn số lượng	$(\geq k\ r)$ $(\leq k\ r)$	≥ 2 hasSister ≤ 3 hasBrother	$\{o \mid \#\{o' \mid (o, o') \in r^{\mathcal{I}}\} \geq k\}$ $\{o \mid \#\{o' \mid (o, o') \in r^{\mathcal{I}}\} \leq k\}$
Giới hạn số lượng định kiểu	$(\geq k\ r.C)$ $(\leq k\ r.C)$	≥ 2 hasSibling.F ≤ 3 hasSibling.M	$\{o \mid \#\{o' \mid (o, o') \in r^{\mathcal{I}} \wedge o' \in C^{\mathcal{I}}\} \geq k$ $\{o \mid \#\{o' \mid (o, o') \in r^{\mathcal{I}} \wedge o' \in C^{\mathcal{I}}\} \leq k$

Nhiều cấu trúc DL và các tổ hợp của chúng đã được nghiên cứu. Mỗi tổ hợp tương ứng với 1 ngôn ngữ DL.

Biểu diễn các biểu thức vai trò

Cấu trúc	**Cú pháp**	**Ví dụ**	**Ngữ nghĩa**
Vai trò hạt nhân	r	hasChild	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Phần bù vai trò	$\neg r$	¬hasSister	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus \{(o, o') \in r^{\mathcal{I}}\}$
Vai trò ngược	r^{-}	hasParent⁻	$\{(o, o') \mid (o', o) \in r^{\mathcal{I}}\}$
Chuỗi vai trò	$r \circ r'$	hasChild ∘ hasParent	$\{(o, o') \mid (o, o'') \in r^{\mathcal{I}}, (o'', o') \in r'^{\mathcal{I}}\}$

Ngữ nghĩa của các mệnh đề

Kiểm tra biểu diễn $\mathcal{I}$ có phải là mô hình của mệnh đề α hay không (ký hiệu là: $\mathcal{I} \vDash \alpha$)

Tên	**Cú pháp**	**Ngữ nghĩa**	**Ký hiệu**
bao gồm	$C \sqsubseteq D$	đúng nếu $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	$\mathcal{I} \vDash C \sqsubseteq D$
tương đương	$C \equiv D$	đúng nếu $C^{\mathcal{I}} = D^{\mathcal{I}}$	$\mathcal{I} \vDash C \equiv D$
kiểm tra khái niệm	$C(a)$	đúng nếu $a^{\mathcal{I}} \in C^{\mathcal{I}}$	$\mathcal{I} \vDash C(a)$
khẳng định vai trò	$r(a, b)$	đúng nếu $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$	$\mathcal{I} \vDash r(a, b)$
phủ định vai trò	$\neg r(a, b)$	đúng nếu $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$	$\mathcal{I} \vDash r(a, b)$

Ngữ nghĩa dựa trên FOL

- Ngữ nghĩa của biểu thức DL có thể được xác định dựa trên biểu thức tương ứng trong FOL.
- Đặt τ là hàm biến đổi biểu thức α trong lô-gic mô tả thành biểu thức tương ứng $\tau(\alpha)$ trong lô-gic bậc nhất.
- Đối với K là CSTT trong DL, $\tau(K) = \bigcup_{\alpha \in K} \tau(\alpha)$

Biến đổi DL về FOL

Các mô tả tài nguyên

Biểu thức DL	**Biểu thức FOL**
C(α)	C(α)
r(a, b)	r(a, b)
¬r(a, b)	¬r(a, b)

Biến đổi DL về FOL: Các biểu thức khái niệm

Biểu thức DL	Biểu thức FOL
C	$(\forall x)C(x)$
$\neg C$	$(\forall x)(\neg C(x))$
$C \sqcap D$	$(\forall x)(C(x) \wedge D(x))$
$C \sqcup D$	$(\forall x)(C(x) \vee D(x))$
$D \equiv \forall r.C$	$(\forall x)(D(x) \Leftrightarrow (\forall y)(r(x, y) \rightarrow C(y)))$
$D \equiv \exists r.C$	$(\forall x)(D(x) \Leftrightarrow (\exists y)(r(x, y) \wedge C(y)))$
$C \sqsubseteq D$	$(\forall x)(C(x) \rightarrow D(x))$
$\{a\}$	$x = a$
$\exists s.Self$	$s(x, x)$
$\geq n\ s.C$	$(\exists x_1) \ldots (\exists x_n)(\wedge_{i \neq j}(x_i \neq x_j) \wedge \wedge_i(s(x, x_i) \wedge C(x_i)))$
$\leq n\ s.C$	$\neg(\exists x_1) \ldots (\exists x_n)(\wedge_{i \neq j}(x_i \neq x_j) \wedge \wedge_i(s(x, x_i) \wedge C(x_i)))$

Biến đổi DL về FOL: Các biểu thức vai trò

Biểu thức DL	Biểu thức FOL
$s = r^{-}$	$(\forall x)(\forall y)(r(x, y) \rightarrow s(y, x))$
$r \sqsubseteq s$	$(\forall x)(\forall y)(r(x, y) \rightarrow s(x, y))$
Ref(r)	$(\forall x)(r(x, x))$
Asy(r)	$(\forall x)(\forall y)(r(x, y) \rightarrow \neg r(y, x))$
Dis(r, s)	$\neg(\exists x)(\exists y)(r(x, y) \land r(y, x))$
$r_1 \circ \cdots \circ r_n \sqsubseteq s$	$(\forall x)(\forall y)(\exists x_1) \ldots (\exists x_{n-1})(r_1(x, x_1) \land \ldots \land r_n(x_n, y) \rightarrow s(x, y))$

Các khái niệm tương đương

$C \equiv D$, các biểu thức khái niệm C và D là tương đương nếu với tất cả các biểu diễn $\mathcal{I}$ thì $C^{\mathcal{I}} = D^{\mathcal{I}}$.

- Các tính chất giao hoán, kết hợp, lũy đẳng

  $C \sqcap D \equiv D \sqcap C$ $\quad$ $C \sqcup D \equiv D \sqcup C$

  $(C \sqcap D) \sqcap E \equiv C \sqcap (D \sqcap E)$ $\quad$ $(C \sqcup D) \sqcup E \equiv C \sqcup (D \sqcup E)$

  $C \sqcap C \equiv C$ $\quad$ $C \sqcup C \equiv C$

- Phủ định kép

  $\neg\neg C \equiv C$

- Phần bù và luật De Morgan

  $\neg\top \equiv \bot$ $\quad$ $\neg\bot \equiv \top$

  $C \sqcap \neg C \equiv \bot$ $\quad$ $C \sqcup \neg C \equiv \top$

  $\neg(C \sqcap D) \equiv \neg C \sqcup \neg D$ $\quad$ $\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$

Các khái niệm tương đương$_{(2)}$

- Phân phối, sát nhập

  $(C \sqcup D) \sqcap E \equiv (C \sqcap E) \sqcup (D \sqcap E)$ $(C \sqcup D) \sqcap C \equiv C$

  $(C \sqcap D) \sqcup E \equiv (C \sqcup E) \sqcap (D \sqcup E)$ $(C \sqcap D) \sqcup C \equiv C$

- Định lượng và cơ số, với n > 0:

  $\neg(\leq n\ s) \equiv\ > (n + 1)\ s$ $\neg(\geq 0\ s) \equiv \bot$

  $\neg(> n\ s) \equiv\ \leq (n - 1)\ s$ $\geq 1\ s \equiv \exists s.\top$

  $\leq 0\ s \equiv \forall s.\bot$

  $\top \sqsubseteq\ \leq 1s$ đồng nghĩa với Func(s)

  $\neg \exists r.C \equiv \forall r.\neg C$ $\geq 0\ r.C \equiv \top$

  $\neg \forall r.C \equiv \exists r.\neg C$ $\geq 1\ r.C \equiv \exists r.C$

  $\leq 0\ r.C \equiv \forall r.\neg C$

  $\neg \leq n\ r.C \equiv\ > (n + 1) r.C$ $< n\ r.C \equiv \neg \geq (n + 1) r.C$

Tương đương mệnh đề và CSTT

- Biến đổi Lloyd-Topor

  $A \sqcup B \sqsubseteq C \Leftrightarrow \{A \sqsubseteq C, B \sqsubseteq C\}$

  $A \sqsubseteq B \sqcap C \Leftrightarrow \{A \sqsubseteq B, A \sqsubseteq C\}$

- Hộp dữ kiện (ABox) và Hộp từ vựng

  $C(a) \Leftrightarrow \{a\} \sqsubseteq C$

  $r(a, b) \Leftrightarrow \{a\} \sqsubseteq \exists\ r.\{b\}$

  $\neg r(a, b) \Leftrightarrow \{a\} \sqsubseteq \neg \exists\ r.\{b\}$

  $a = b \Leftrightarrow \{a\} \equiv \{b\}$

  $a \neq b \Leftrightarrow \{a\} \sqcap \{b\} \equiv \bot$

Bao hàm khái niệm và các hệ quả

- $C \sqsubseteq D \Leftrightarrow \top \equiv \neg C \sqcup D$
- $C \sqsubseteq D \Leftrightarrow D \equiv C \sqcup D$
- $C \equiv D \Leftrightarrow \{C \sqsubseteq D, D \sqsubseteq C\}$
- $C \sqsubseteq D \land D \sqsubseteq E \Rightarrow \{C \sqsubseteq E\}$
- $C \sqsubseteq D \Leftrightarrow \neg D \sqsubseteq \neg C$
- $C \sqsubseteq D \Rightarrow C \sqcap E \sqsubseteq D$
- $C \equiv D \Rightarrow C \sqcap E \equiv D \sqcap E$

Ví dụ 6.4. CSTT

$Human \sqsubseteq Animal \sqcap Biped$

$Man \equiv Human \sqcap Male$

$Male \sqsubseteq \neg Female$

$\{President\ Obama\} \equiv \{Barack\_Obama\}$

$\{john\} \sqsubseteq \neg\{peter\}$

$hasDaughter \sqsubseteq hasChild$

$hasChild \equiv hasParent^{-}$

$cost \equiv price$

$Trans(ancestor)$

$Func(hasMother)$

$Func(hasSSN^{-})$

Nội dung

Các OWL và CSTT DL

- Tiếp nối các thành tựu lô-gic, OWL cung cấp thành phần mô tả ontology

OWL	**DL**
Lớp-class	khái niệm-concept
thuộc tính-property	vai trò-role
thuộc tính đối tượng	vai trò trừu tượng
object property	abstract role
thuộc tính dữ liệu	vai trò cụ thể
data property	concrete role
lớp đóng-oneOf	đơn vị-nominal
CSTT(KB-Knowledge Base)	CSTT

Các biểu thức khái niệm DL vs. lớp OWL

- ⊤ tương đương với owl:Thing
- ⊥ tương đương với owl:Nothing
- ⊓ tương đương với owl:intersectionOf
- ⊔ tương đương với owl:unionOf
- ¬ tương đương với owl:complementOf
- ∀ tương đương với owl:allValuesFrom
- ∃ tương đương với owl:someValuesFrom

DL và OWL: Lớp và khái niệm

C = C rdf:type owl:Class
$C \sqsubseteq D$ = C rdfs:subClassOf D
$\exists r.C$ = [a owl:Restriction;owl:onProperty r; owl:someValuesFrom C]
$\forall r.C$ = [a owl:Restriction;owl:onProperty r; owl:allValuesFrom C]
$\geq n\ r.C$ = [a owl:Restriction; owl:minQualifiedCardinality n; owl:onProperty r; owl:onClass C].
$\leq n\ r.C$ = [a owl:Restriction; owl:maxQualifiedCardinality n; owl:onProperty r; owl:onClass C].
$\exists r.T \sqsubseteq C$ = r rdfs:domain C.
$T \sqsubseteq \forall r.C$ = r rdfs:range C.
$\{a_1, \ldots, a_n\}$ = [a owl:Class; owl:oneOf($a_1\ a_2\ \ldots\ a_n$)]
$\neg C$ = [owl:complementOf C].
$C_1 \sqcap \ldots \sqcap C_n$ = [owl:intersectionOf($C_1 \ldots C_n$)]
$C_1 \sqcup \ldots \sqcup C_n$ = [owl:unionOf($C_1 \ldots C_n$)]

DL và OWL: Thuộc tính và vai trò

r = r rdf:type rdf:Property
r^{-} = [owl:inverseOf r]
u = owl:topObjectProperty
$s \sqsubseteq r$ = s rdfs:subPropertyOf r
$r_1 \circ \ldots \circ r_n \sqsubseteq r$ = r owl:propertyChainAxiom(r_1, …, r_n)
Dis(r, s) = r owl:propertyDisjointWith s.

DL và OWL: Các dữ kiện

C(x) = x a C.
r(a, b) = a r b.
r^{-}(a, b) = b r a.
a ≈ b = a owl:sameAs b.
a ≉ b = a owl:differentFrom b.

Ví dụ 6.5. Các giới hạn trong OWL

- owl:hasValue bắt buộc sử dụng 1 phần tử cụ thể

  :Woman owl:equivalentClass [

      a owl:Restriction;

      owl:onProperty :hasGender;

      owl:hasValue :female

  ] .
- Trong lô-gic mô tả:

$$\text{Woman} \equiv \exists\, \text{hasGender}.\{\text{female}\}$$

Ví dụ 6.6. CSTT DL & OWL

RBox $\mathcal{R}$

own $\sqsubseteq$ careFor

"Nếu ai đó sở hữu cái gì thì họ quan tâm về nó"

TBox $\mathcal{T}$

Healthy $\sqsubseteq \neg$Dead

Cat $\sqsubseteq$ Dead $\sqcup$ Alive

HappyCatOwner $\sqsubseteq \exists$owns.Cat $\sqcap \forall$caresFor.Healthy

"Sống khỏe mạnh là không chết"; " Mèo có thể đã chết hoặc còn sống"; "Chủ nhân hạnh phúc của mèo là người sở hữu mèo và tất cả mèo mà người đó quan tâm đều khỏe mạnh."

ABox $\mathcal{A}$

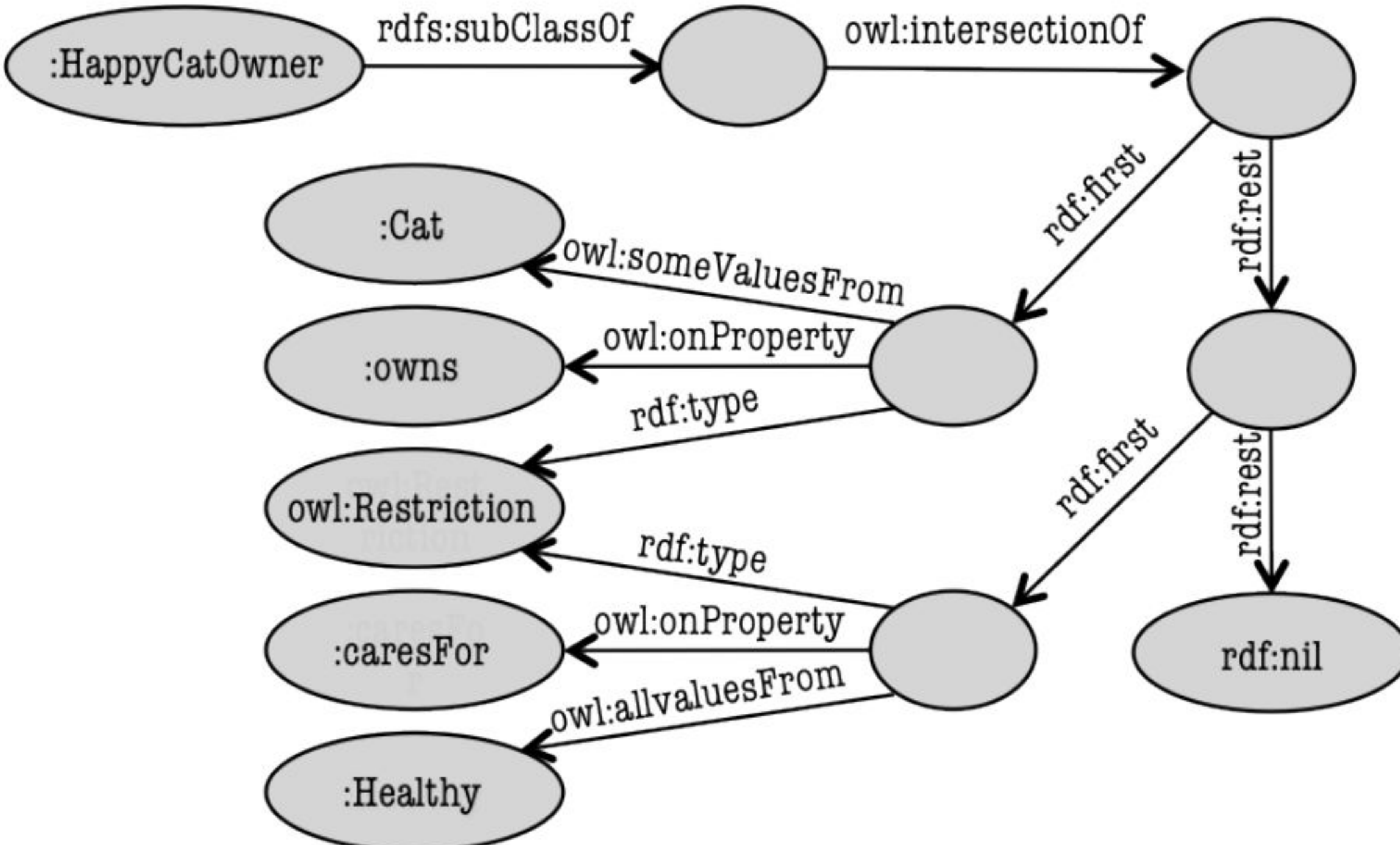
HappyCatOwner(NVA)

"NVA là người sở hữu mèo hạnh phúc."

Ví dụ 6.6. CSTT DL & OWL$_{(2)}$

```
:owns rdfs:SubPropertyOf :caresFor .
:Healthy rdfs:subClassOf [ owl:complementOf :Dead ].
:Cat rdfs:subClassOf [ owl:unionOf (:Dead :Alive) ] .
:HappyCatOwner rdfs:subClassOf
          [ owl:intersectionOf
          ( [ rdf:type owl:restriction ;
              owl:onProperty :owns ;
              owl:someValuesFrom :Cat ]
            [ rdf:type owl:Restriction ;
               owl:onProperty :caresFor ;
               owl:allValuesFrom :Healthy ] ) ] .
:NVA rdf:type :HappyCatOwner .
```

Ví dụ 6.6. CSTT DL & OWL$_{(3)}$



Biến đổi các khái niệm ABox

Nếu các lớp đóng được hỗ trợ thì ABox có thể được chuyển đổi thành (1 phần) TBox

$C(a) = \{a\} \sqsubseteq C$

$r(a, b) = \{a\} \sqsubseteq \exists r.\{b\}$

$\neg r(a, b) = \{a\} \sqsubseteq \forall r.(\neg\{b\})$

$a \approx b = \{a\} \equiv \{b\}$

$a \not\approx b = \{a\} \sqsubseteq \neg\{b\}$

Nội dung

6.1. Tổng quan về vấn đề biểu diễn tri thức

6.2. Lô-gic mệnh đề và lô-gic bậc nhất

6.3. Lô-gic mô tả

6.4. Ngữ nghĩa lô-gic của OWL

6.5. Suy diễn tự động với OWL

Giả thuyết thế giới mở vs thế giới đóng

Giả thuyết thế giới mở - Open World Assumption (OWA)

- Có thể tồn tại các thành phần mới nếu nó không bị bác bỏ tường minh
- OWL sử dụng OWA

Giả thuyết thế giới đóng - Closed World Assumption (CWA)

- Được cho rằng CSTT có tất cả các phần tử và dữ kiện - Phạm vi là những gì đang có.

Ví dụ 6.7. OWA và CWA

	Có phải tất cả con của Bill đều là con trai?	Không rõ nếu chúng ta cho rằng không biết tất cả về Bill	Nếu chúng ta biết tất cả thì tất cả con của Bill đều là con trai
child(bill, bob) Man(bob)	⊨? $(\forall$child.Man)(bill)	Không rõ	Yes
(≤ 1 child)(bill)	⊨? $(\forall$child.Man)(bill)	Yes	Yes

Các vấn đề suy diễn

Suy diễn lô-gic trong CSTT

- Đặt $\mathcal{I}$ là 1 biểu diễn, $\mathcal{T}$ là 1 TBox, $\mathcal{A}$ là 1 ABox và $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ là 1 CSTT
- $\mathcal{I}$ là mô hình của $\mathcal{T}$ nếu $\mathcal{I} \vDash$ ax cho tất cả mệnh đề trong $\mathcal{T}$, viết là $\mathcal{I} \vDash \mathcal{T}$.
- $\mathcal{I}$ là mô hình của $\mathcal{A}$, nếu $\mathcal{I} \vDash$ ax cho tất cả mệnh đề trong $\mathcal{A}$, viết là $\mathcal{I} \vDash \mathcal{A}$.
- $\mathcal{I}$ là mô hình của $\mathcal{K}$, nếu $\mathcal{I} \vDash \mathcal{T}$ và $\mathcal{I} \vDash \mathcal{A}$
- Một mệnh đề ax được suy ra từ $\mathcal{K}$, viết là $\mathcal{K} \vDash$ ax, nếu tất cả mô hình $\mathcal{I}$ của $\mathcal{K}$ cũng là mô hình của ax.

Các vấn đề suy diễn cho CSTT $\mathcal{K}$

- Tính nhất quán toàn cục của CSTT:
  - Giả sử $\mathcal{K} \models^? \text{false}$, $\mathcal{K} \models^? \top \sqsubseteq \bot$?
  - CSTT có hợp lý hay không?
- Tính nhất quán lớp: $\mathcal{K} \models^? C \sqsubseteq \bot$?
  - Lớp C có phải là lớp rỗng ko?
- Bao gồm lớp: $\mathcal{K} \models^? C \sqsubseteq D$?
  - Cấu trúc cây của CSTT
- Lớp tương đương: $\mathcal{K} \models^? C \approx D$?
  - Các lớp có tập phần tử giống nhau hay không?

Các vấn đề suy diễn cho CSTT $\mathcal{K}_{(2)}$

- Lớp không giao nhau: $\mathcal{K} \models^? C \sqcap D \sqsubseteq \bot$?
  - Các lớp có giao nhau hay không?
- Phần tử của lớp: $\mathcal{K} \models^? C(a)$?
  - Phần tử a có nằm trong lớp C hay không?
- Truy xuất phần tử: Tìm tất cả x sao cho $\mathcal{K} \models C(x)$
  - Tìm tất cả các phần tử của lớp C

Các dạng kiểm tra ABox

Một kiểm tra ABox có thể có 1 trong các dạng sau:

- C(a) (kiểm tra khái niệm)
- r(a, b) (kiểm tra vai trò)
- ¬r(a, b) (kiểm tra vai trò phủ định)
- a ≈ b (kiểm tra tương đương)
- a ≉ b (kiểm tra khác biệt)

Tính khả quyết của OWL DL

- OWL DL là 1 phần của FOL, như vậy có thể sử dụng tiến trình suy diễn FOL (phân giải Tableaux)
    - Nhưng có thể không đảm bảo tính dừng
- Vấn đề: Tìm kiếm các giải thuật có đảm bảo tính dừng
    - Không có giải pháp đơn giản cho vấn đề này.

Suy diễn trong OWL 2

- OWL 2 mở rộng các tính năng đã giới thiệu với các cấu trúc mới
- OWL 2 cũng định nghĩa các thành phần đơn giản hơn
- Nhiều công cụ cho phép suy diễn tự động
- Trình biên soạn hỗ trợ tạo ontologies/CSTT

Bài tập 6.1

Xây dựng các Ontology với OWL DL, lô-gic mô tả và lô-gic bậc nhất.

- The class `Vegetable` is a subclass of `PizzaTopping`.
- The class `PizzaTopping` does not share any elements with the class `Pizza`.
- The individual `aubergine` is an element of the class `Vegetable`.
- The abstract role `hasTopping` is only used for relationships between elements of the classes `Pizza` and `PizzaTopping`.
- The class `VegPizza` consists of those elements which are in the class `NoMeatPizza` and in the class `NoFishPizza`.
- The role `hasTopping` is a subrole of `hasIngredient`.

Bài tập 6.1 (2)

:Vegetable rdfs:subClassOf :PizzaTopping.

:PizzaTopping owl:disjointWith :Pizza.

:aubergine a :Vegtable.

:hasTopping rdfs:domain :Pizza.

:hasTopping rdfs:range :PizzaTopping.

[owl:intersectionOf(:NoMeatePizza :NoFishPizza)]

rdfs:subClassOf :VegPizza.

:hasTopping rdfs:subPropertyOf :hasIngredient.

Bài tập 6.1$_{(3)}$

$Vegetable \sqsubseteq PizzaTopping$

$PizzaTopping \sqsubseteq \neg Pizza$

Vegetable(aubergine)

$\exists hasTopping.\top \sqsubseteq Pizza$

$\top \sqsubseteq \forall hasTopping.PizzaTopping$

$NoMeatPizza \sqcup NoFishPizza \sqsubseteq VegPizza$

$hasTopping \sqsubseteq hasIngredient$

Bài tập 6.1$_{(4)}$

$(\forall x)(Vegetable(x) \rightarrow PizzaTopping(x))$

$\neg(\exists x)(PizzaTopping(x) \wedge Pizza(x))$

$Vegetable(aubergine)$

$(\forall x)((\exists y)hasTopping(x, y) \rightarrow Pizza(x))$

$(\forall x)(\forall y)(hasTopping(x, y) \rightarrow PizzaTopping(y))$

$(\forall x)(NoMeatPizza(x) \sqcap NoFishPizza(x) \rightarrow VegPizza(x))$

$(\forall x)(\forall y)(hasTopping(x, y) \rightarrow hasIngredient(x, y))$

Bài tập 6.2

Chuyển đổi sang OWL

$$\mathtt{Human} \sqsubseteq \exists \mathtt{hasMother.Human}$$

$$\exists \mathtt{hasMother.}(\exists \mathtt{hasMother.Human}) \sqsubseteq \mathtt{Grandchild}$$

$$\mathtt{Human}(\mathtt{anupriyaAnkolekar})$$

Bài tập 6.2

Thử nghiệm suy diễn:

$$\mathtt{Human} \sqsubseteq \exists \mathtt{hasMother}.\mathtt{Human}$$

$$\exists \mathtt{hasMother}.(\exists \mathtt{hasMother}.\mathtt{Human}) \sqsubseteq \mathtt{Grandchild}$$

$$\mathtt{Human}(\mathtt{anupriyaAnkolekar})$$

:a :hasMother :b.
:b :hasMother :c.
:c a :Woman.
:Woman rdfs:subClassOf :Human.

=> :a a :GrandChild ??

Bài tập 6.3

```
:Person a owl:Class.
:Food a owl:Class.
:eats rdfs:domain :Person;
        rdfs:range :Food.
:Maverick :eats :Steak.
:Vegetarian a owl:Class;
                  rdfs:subClassOf :Person.
:VegetarianFood a owl:Class;
                  rdfs:subClassOf :Food.
:Jen a :Vegetarian;
       :eats :Marzipan.
```

a) Hãy bổ xung các mô tả dựa trên owl để có thể suy ra :Marzipan là :VegetarianFood, nhưng :Steak thì không?

b) Chuyển sang DL.

Bài tập 6.4

1. Tạo Ontology trong phạm vi ngữ nghĩa owl, sử dụng cú pháp Turtle dựa trên các mô tả:

a. Bus Drivers are Drivers who drive buses.

b. Drivers are people who drive at least one vehicle.

c. Bus is a subclass of vehicle

Sử dụng các từ sau cho các lớp: BusDriver, Driver, Bus, Vehicle

2. Chuyển Ontology thu được ở 1 sang DL.

3. Hãy tìm 3 bộ 3 có thể suy diễn được từ các dữ kiện

  :b001 rdf:type :Bus.

  :NVA :drive :b001.

dựa trên Ontology đã tạo trong 1?