

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

LUẬN VĂN THẠC SĨ KHOA HỌC

LOGIC MÔ TẢ VÀ ỨNG DỤNG
TRONG CƠ SỞ DỮ LIỆU

NGÀNH: CÔNG NGHỆ THÔNG TIN

MÃ SỐ:.....

ĐẶNG VĂN HUỆ

Người hướng dẫn khoa học: TS. TRẦN ĐÌNH KHANG

HÀ NỘI 2006

LỜI CAM ĐOAN

Các kết quả nghiên cứu trong luận văn, ngoài những vấn đề mang tính phổ biến mà tác giả đề cập tới dưới dạng các định nghĩa và khái niệm là hoàn toàn mới, những vấn đề tham khảo được trích dẫn cụ thể. Các hình, minh hoạ, ví dụ và kết quả do chính tác giả thực hiện. Nội dung của đề tài chưa công bố trên các công trình nghiên cứu khác. Tác giả xin chịu hoàn toàn trách nhiệm về nội dung của luận văn này.

Tác giả

Đặng Văn Huệ

LỜI CẢM ƠN

Dưới sự dẫn dắt của các thầy, các cô giáo trường Đại học Bách khoa Hà Nội đến nay em đã hoàn thành luận văn tốt nghiệp này.

Em xin chân thành cảm ơn các thầy, các cô trường Đại học Bách Khoa Hà Nội nói chung và Khoa Công nghệ Thông tin nói riêng đã tận tình chỉ bảo, hướng dẫn cho em trong những năm qua.

Em xin bày tỏ lòng biết ơn đến thầy giáo Trần Đình Khang, người trực tiếp hướng dẫn em làm luận văn. Nếu không có sự truyền đạt kiến thức quý báu và hướng dẫn tận tình của thầy giáo chắc chắn rằng luận văn của em sẽ rất khó được hoàn thành.

Tôi cũng xin chân thành cảm ơn bạn bè đã động viên, giúp đỡ tôi trong thời gian học tập tại Trường, cũng như quá trình hoàn thành luận văn.

Mặc dù đã rất cố gắng, song chắc chắn luận văn không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và những ý kiến đóng góp tận tình của các thầy, cô giáo và các bạn cũng như những ai quan tâm tới lĩnh vực trong luận văn này.

Hà Nội, ngày 31 tháng 10 năm 2006

Tác giả

Đặng Văn Huệ

MỤC LỤC

Nội dung	Trang
LỜI CAM ĐOAN	1
LỜI CẢM ƠN	2
MỤC LỤC	3
DANH SÁCH CÁC BẢNG	6
DANH SÁCH CÁC HÌNH	6
LỜI GIỚI THIỆU	7
Chương 1. LOGIC MÔ TẢ	10
1.1. GIỚI THIỆU	10
1.2. NGÔN NGỮ THUỘC TÍNH AI	11
1.2.1. Ngôn ngữ mô tả cơ bản AI	11
1.2.2. Ngữ nghĩa của các khái niệm AI	12
1.2.3. Họ ngôn ngữ logic mô tả AI	13
1.2.4. Ngôn ngữ mô tả là tập con của logic vị từ bậc nhất	15
1.3. HỆ CƠ SỞ TRI THỨC	15
1.3.1. Kiến trúc hệ logic mô tả	15
1.3.2. Bộ thuật ngữ (TBox)	16
1.3.2.1. Tiên đề thuật ngữ	16
1.3.2.2. Định nghĩa khái niệm	17
1.3.2.3. Mở rộng bộ thuật ngữ	20
1.3.2.4. Đệ quy	22
1.3.2.5. Thuật ngữ với các tiên đề bao hàm	22
1.3.3. Bộ khẳng định (ABox)	23
1.3.4. Cá thể	25

1.3.5. Suy luận.....	26
1.3.5.1. Lập luận đối với khái niệm.....	26
1.3.5.2 Loại trừ TBox	28
1.3.5.3. Lập luận đối với ABox	29
1.3.5.4. Ngữ nghĩa “đóng”, ngữ nghĩa “mở”	30
1.4. CÁC THUẬT TOÁN SUY LUẬN	33
1.4.1. Thuật toán bao hàm cấu trúc	33
1.4.2. Thuật toán tableau	35
1.5. MỞ RỘNG NGÔN NGỮ MÔ TẢ	41
1.5.1. Các constructor vai trò	41
1.5.2. Biểu diễn các giới hạn số	42
1.6. NGÔN NGỮ DATALOG	42
1.6.1. Các khái niệm và thành phần của Datalog.....	43
1.6.2. Cú pháp của chương trình Datalog	44
1.7. TỔNG KẾT CHƯƠNG	46
Chương 2. SƠ LƯỢC VỀ CƠ SỞ DỮ LIỆU	48
2.1. MÔ HÌNH THỰC THỂ - QUAN HỆ.....	48
2.2. MÔ HÌNH HƯỚNG ĐỐI TƯỢNG.....	52
2.3. TỔNG KẾT CHƯƠNG	56
Chương 3. CHUYỂN ĐỔI CƠ SỞ DỮ LIỆU THÀNH CƠ SỞ TRI	
THỨC CỦA LOGIC MÔ TẢ.....	57
3.1. MÔ HÌNH HOÁ LƯỢC ĐỒ THỰC THỂ - QUAN HỆ	
BẢNG LOGIC MÔ TẢ	57
3.2. MỞ RỘNG KHẢ NĂNG BIỂU DIỄN CỦA NGÔN	
NGỮ MÔ HÌNH HOÁ	63
3.2.1. Tổng quát hoá thực thể.....	63
3.2.2. Lọc các tính chất thuộc một cấu trúc IS-A.....	64

3.3. BIỂU DIỄN MÔ HÌNH DỮ LIỆU HƯỚNG ĐỐI TƯỢNG BẰNG LOGIC MÔ TẢ.....	64
3.4. CHUYỂN DỮ LIỆU TỪ CƠ SỞ DỮ LIỆU VÀO ABOX CỦA LOGIC MÔ TẢ.....	66
3.5 TỔNG KẾT CHƯƠNG	72
Chương 4. TRUY VẤN	73
4.1. NGUYÊN TỬ TRUY VẤN, ĐỐI TƯỢNG, CÁ THỂ VÀ BIẾN	73
4.1.1. Nguyên tử truy vấn khái niệm.....	73
4.1.2. Nguyên tử truy vấn vai trò	74
4.2. TRUY VẤN PHỨC HỢP	75
4.3. HỖ TRỢ MÔ TẢ - ĐỊNH NGHĨA VÀ THUẬT TOÁN.....	76
4.4. TỔNG KẾT CHƯƠNG	78
KẾT LUẬN	79
CÁC THUẬT NGỮ	80
TÀI LIỆU THAM KHẢO.....	82

DANH SÁCH CÁC BẢNG

1.1	Cú pháp của ngôn ngữ $\mathcal{AL}$	trang 12
1.2	Ngữ nghĩa của logic mô tả	trang 13
3.1	Bảng thực thể Professor	trang 67
3.2	Bảng thực thể Student	trang 68
3.3	Bảng thực thể Course	trang 68
3.4	Bảng thực thể AdvCourse	trang 69
3.5	Bảng quan hệ Teaching	trang 69
3.6	Bảng thực thể GradStudent	trang 69
3.7	Bảng quan hệ Enrolling	trang 69

DANH SÁCH CÁC HÌNH

1.1	Kiến trúc hệ logic mô tả	trang 16
1.2	TBox với các khái niệm về quan hệ gia đình	trang 18
1.3	Khai triển TBox quan hệ gia đình trong Hình 1.2	trang 20
1.4	Bộ khẳng định (ABox)	trang 23
1.5	ABox A_{oe} về câu truyện <i>Oedipus</i>	trang 30
1.6	Luật biến đổi của thuật toán tableau giải bài toán thỏa	trang 37
1.7	Ví dụ chứng minh Mother $\vee$ Parent	trang 39
2.1	Lược đồ ER	trang 49
2.2	Một mô hình hướng đối tượng	trang 52
3.1	TBox chuyển đổi từ lược đồ ER trong Hình 2.1	trang 59
3.2	Cơ sở tri thức $\mathcal{ALCQI}$ tương ứng với lược đồ trong Hình 2.2	trang 65
3.3	Thủ tục chuyển dữ liệu từ bảng vào ABox	trang 67
3.3	ABox nhận được từ việc chuyển đổi dữ liệu của các thực thể	trang 71
4.1	Thủ tục hỗ trợ mô tả	trang 76

LỜI GIỚI THIỆU

Nghiên cứu trong lĩnh vực biểu diễn tri thức và suy diễn thường tập trung vào các phương pháp có khả năng mô tả “thế giới” ở mức cao. Trong những năm gần đây, người ta thường nhắc tới “logic mô tả” (Description logic) như là một phương pháp biểu diễn tri thức hiệu quả. Trong những ứng dụng cụ thể có sử dụng logic mô tả, tri thức của miền ứng dụng được đặc tả bằng các khái niệm và các mối quan hệ.

Lĩnh vực ứng dụng của logic mô tả cũng rất đa dạng, ngay từ ngày đầu, logic mô tả đã được xem như là những ngôn ngữ với mục đích biểu diễn tri thức và suy diễn, vì thế nó phù hợp cho nhiều ứng dụng. Tuy nhiên những ứng dụng mang tính thương mại đến nay vẫn chưa thực sự phổ biến.

Các ứng dụng của logic mô tả có thể kể đến như công nghệ phần mềm, thiết lập cấu hình, y học, các hệ thống thư viện điện tử, hệ thống thông tin web ngữ nghĩa, xử lý ngôn ngữ tự nhiên, quản trị cơ sở dữ liệu...

Mối quan hệ giữa logic mô tả và cơ sở dữ liệu khá khăng khít. Thực tế, nhu cầu xây dựng các hệ thống mà vừa có khả năng biểu diễn tri thức logic mô tả và quản trị cơ sở dữ liệu là cần thiết. Các hệ quản trị cơ sở dữ liệu giải quyết vấn đề toàn vẹn dữ liệu và quản trị một số lượng lớn dữ liệu, trong khi đó hệ biểu diễn cơ sở tri thức logic mô tả quản lý tri thức nội hàm. Hơn nữa, logic mô tả cung cấp một khung chuẩn mà được xem như rất gần gũi với các ngôn ngữ được dùng để mô hình hoá dữ liệu, như mô hình thực thể - quan hệ. Logic mô tả tương đương với các công cụ lập luận. Chẳng hạn, bằng việc sử dụng tính nhất quán khái niệm ta có thể xác nhận một thực thể có ít nhất một thể hiện ngay tại thời điểm thiết kế.

Một yếu tố nữa tăng cường cho hệ quản trị cơ sở dữ liệu bằng logic mô tả là ngôn ngữ truy vấn. Bằng việc biểu diễn truy vấn cơ sở dữ liệu trong logic

mô tả người ta có khả năng phân loại chúng, vì thế xử lý kết quả như thực hiện và tối ưu hoá truy vấn. Hơn nữa, logic mô tả có thể được dùng để biểu diễn các ràng buộc và các câu trả lời nội hàm.

Trong thời gian qua em đã có điều kiện được tiếp xúc, nghiên cứu về logic mô tả. Từ những nghiên cứu này, nên trong luận văn em sẽ trình bày theo hướng nêu lên các vấn đề cơ bản của logic mô tả, sơ lược về các mô hình cơ sở dữ liệu phổ biến, mối quan hệ giữa cơ sở dữ liệu và logic mô tả. Do vậy, các nội dung của luận văn này sẽ được trình bày như sau:

- *Chương 1. Logic mô tả:* Đây là chương giới thiệu về những nội dung cơ bản của logic mô tả như khái lược về logic mô tả, các ngôn ngữ của logic mô tả, kiến trúc của một hệ cơ sở tri thức dựa trên logic mô tả, các bài toán quyết định. Đồng thời giới thiệu một ngôn ngữ lập trình logic Datalog.
- *Chương 2. Sơ lược về cơ sở dữ liệu:* Trong chương này em xin đề cập một cách khái lược nhất về hai mô hình cơ sở dữ liệu đó là mô hình dữ liệu thực thể - quan hệ và mô hình dữ liệu hướng đối tượng.
- *Chương 3. Chuyển đổi cơ sở dữ liệu thành cơ sở tri thức của logic mô tả:* Chương này sẽ giới thiệu phương pháp để biến đổi các lược đồ của mô hình dữ liệu thực thể - liên kết cũng như mô hình hướng đối tượng thành bộ thuật ngữ (TBox) của logic mô tả, đồng thời thảo luận về việc chuyển đổi dữ liệu của cơ sở dữ liệu vào bộ khẳng định (ABox) của logic mô tả.

Chương 4. Truy vấn: Chương này thảo luận về truy vấn cơ sở tri thức, từ các thành phần cơ bản của truy vấn như truy vấn nguyên tử khái niệm, truy vấn nguyên tử vai trò đến các truy vấn phức hợp bằng biểu thức hội các thành phần khái niệm và vai trò cơ sở. Đồng thời cũng đưa ra thuật toán nhằm

chuyển đổi các câu truy vấn xây dựng theo cách thể hiện của ngôn ngữ lập trình logic Datalog sang biểu diễn mô tả khái niệm trong logic mô tả.

Trên đây là những phần chính sẽ được trình bày trong luận văn. Trên thực tế vẫn còn nhiều vấn đề mở trong lý thuyết về logic mô tả và ứng dụng của nó. Em hy vọng mình sẽ có điều kiện để tiếp tục đi sâu hơn vào việc nghiên cứu ứng dụng của logic mô tả trong thời gian tới.

Cuối cùng, em xin được gửi lời cảm ơn của mình tới thầy giáo hướng dẫn Tiến sỹ Trần Đình Khang đã dìu dắt, hỗ trợ và giúp đỡ em hoàn thành đề tài này. Phần trình bày của em chắc chắn còn nhiều thiếu sót, em rất mong được sự góp ý của thầy để có thể hoàn thiện tốt hơn đề tài.

Chương 1. LOGIC MÔ TẢ

1.1. GIỚI THIỆU

“Logic mô tả” là thuật ngữ mới nhất trong họ biểu diễn tri thức (KR), trước khi cụm từ “logic mô tả” phổ biến như hiện nay, người ta nói đến logic mô tả dưới những cụm từ như “ngôn ngữ biểu diễn tri thức thuật ngữ” hay “ngôn ngữ khái niệm”. Logic mô tả được ứng dụng rất hiệu quả trong các hệ thống trí tuệ nhân tạo, hệ thống biểu diễn tri thức ngữ nghĩa. Các hệ thống này hoạt động dựa vào khả năng suy luận theo cách của con người thường làm. Đó là phân lớp các khái niệm và các cá thể. Việc phân lớp các khái niệm xác định mối quan hệ (mà người ta gọi là quan hệ bao hàm) giữa các khái niệm của các thuật ngữ cho trước, và như vậy cho phép người ta xây dựng thuật ngữ theo dạng cấu trúc. Cấu trúc này cung cấp những thông tin hữu ích trong kết nối giữa các khái niệm khác nhau và nó có thể được dùng để tăng tốc các dịch vụ lập luận khác. Việc phân lớp các cá thể thực chất là xác định cá thể cho trước có luôn luôn là một thể hiện (instance) của một khái niệm nào đó hay không. Vì vậy nó cung cấp những thông tin hữu ích về tính chất của cá thể.

Để biểu diễn tri thức bằng logic mô tả công việc trước tiên ta phải làm đó là xây dựng các khái niệm từ các khái niệm nguyên tố, các vai trò nguyên tố và bằng các luật khái niệm. Hệ thống khái niệm mà ta có được gọi là bộ thuật ngữ (TBox). Đây là một trong hai thành phần chính của hệ cơ sở tri thức dựa vào logic mô tả. Còn một thành phần chính khác của hệ cơ sở tri thức nêu trên là bộ khẳng định (ABox). Bộ này là tập hợp các khẳng định thể hiện mối quan hệ giữa khái niệm với cá thể hay giữa hai cá thể với nhau. Bên cạnh việc biểu diễn tri thức phần quan trọng khác của hệ logic mô tả là cung cấp các dịch vụ suy luận dựa trên tri thức đã được biểu diễn. Phần lớn các thủ tục suy

luận bằng logic mô tả là các thủ tục quyết định với các câu trả lời “đúng” hoặc “sai”. Để xây dựng một hệ thống cơ sở tri thức dựa trên logic mô tả người ta đã đúc rút thành ba bước quan trọng là:

- Xác định các khái niệm nguyên tố, các vai trò nguyên tố và các cá thể ban đầu;
- Sử dụng một ngôn ngữ khái niệm để xây dựng lên các khái niệm phức hợp;
- Sử dụng các thủ tục suy luận để rút ra những tri thức đúng đắn về các khái niệm và các cá thể nếu có thể.

Để chi tiết hơn, ta sẽ lần lượt tìm hiểu từng vấn đề trong logic mô tả. Trước hết là các ngôn ngữ định nghĩa khái niệm, tiếp theo là về cơ sở tri thức được xây dựng bằng logic mô tả và các thủ tục suy diễn cho các bài toán quyết định.

1.2. NGÔN NGỮ THUỘC TÍNH AL

Những khái niệm phức tạp trong logic mô tả được xây dựng bằng ngôn ngữ thuộc tính AL (Attributive Language) hoặc các ngôn ngữ mở rộng của AL. Ta gọi các ngôn ngữ này là các “ngôn ngữ mô tả”. Xuất phát từ những “mô tả cơ sở” bằng các luật xây dựng khái niệm mà ngôn ngữ mô tả hỗ trợ ta hình thành nên các khái niệm mới.

Thành phần cơ bản của ngôn ngữ mô tả AL là các khái niệm và các vai trò nguyên tố. Các mô tả phức tạp được xây dựng bằng việc kết hợp các thành phần cơ bản đó thông qua các bộ tạo (constructor). Người ta thường dùng ký tự A và B để biểu diễn các khái niệm nguyên tố, ký tự R và P để biểu diễn các vai trò, ký tự C và D để biểu diễn các khái niệm phức hợp.

1.2.1. Ngôn ngữ mô tả cơ bản AL

AL là ngôn ngữ có luật cú pháp đơn giản nhất. Những luật cú pháp của ngôn ngữ mô tả AL thể hiện như sau:

$C, D \mid A$		(Khái niệm nguyên tố)
$>$		(Khái niệm đỉnh)
$?$		(Khái niệm đáy)
$:$		(Phủ định khái niệm)
$C \sqcup D$		(Giao khái niệm)
$\forall R.C$		(Lượng từ với mọi)
$\exists R.T$		(Lượng từ tồn tại)

Bảng 1.1: Cú pháp của ngôn ngữ AL

Ví dụ: Giả sử ta có các khái niệm nguyên tố PERSON và MALE thì

$\text{PERSON} \sqcup \text{MALE}$ và $\text{PERSON} \sqcup :\text{MALE}$

là các mô tả khái niệm. Ta thấy rằng các mô tả trên là “Người đàn ông” và “Người không phải là đàn ông”.

Giả sử ta có một vai trò nguyên tố hasChild biểu thị rằng một cá thể có con. Ta có thể tạo ra các mô tả khái niệm:

$\text{PERSON} \sqcup \exists \text{hasChild}.>$

để biểu diễn người có con

và $\text{PERSON} \sqcup \forall \text{hasChild}.: \text{MALE}$

để biểu diễn người có toàn con gái.

Sử dụng khái niệm đáy ta có thể biểu diễn người không có con như sau:

$\text{PERSON} \sqcup \forall \text{hasChild}.?$

1.2.2. Ngữ nghĩa của các khái niệm AL

Bên cạnh việc xây dựng các khái niệm, ta cũng cần phải hiểu từng khái niệm được tạo ra. Ngữ nghĩa của các khái niệm trong logic mô tả được thể hiện thông qua phép diễn dịch.

Định nghĩa 1 [8]: Mỗi diễn dịch, ký hiệu là I , là một cặp $(4^I, .^I)$. Trong đó, miền diễn dịch 4^I là một tập không rỗng, còn $.^I$ là một hàm dịch. Hàm dịch $.^I$ chuyển mỗi khái niệm A thành một tập $A^I \subseteq 4^I$, chuyển mỗi vai trò R thành một quan hệ $R^I \subseteq 4^I \times 4^I$.

Hàm diễn dịch được mở rộng cho khái niệm phức hợp như sau:

$\top$	$=$	4^I
$\perp$	$=$	$\emptyset$
$(\neg C)^I$	$=$	$4^I \setminus C^I$
$(C \sqcup D)^I$	$=$	$C^I \cup D^I$
$(C \sqcap D)^I$	$=$	$C^I \cap D^I$
$(\forall R.C)^I$	$=$	$\{a \in 4^I \mid \forall b.(a,b) \in R^I \Rightarrow b \in C^I\}$
$(\exists R.\top)^I$	$=$	$\{a \in 4^I \mid \exists b.(a,b) \in R^I\}$

Bảng 1.2: Ngữ nghĩa của logic mô tả

Ta nói rằng hai khái niệm C và D là tương đương nhau, viết là $C \equiv D$ nếu $C^I = D^I$ với mọi diễn dịch I .

Ví dụ: Quay trở lại định nghĩa ngữ nghĩa của các khái niệm, ta dễ dàng thấy rằng hai mô tả khái niệm:

$\forall \text{hasChild.Male} \sqcup \forall \text{hasChild.Student}$ và $\forall \text{hasChild.}(\text{Male} \sqcup \text{Student})$ là tương đương nhau.

1.2.3. Họ ngôn ngữ logic mô tả AL

Khi ta thêm các bộ tạo (constructor) vào ngôn ngữ AL cơ bản ta được một ngôn ngữ AL mở rộng có khả năng biểu diễn linh hoạt hơn. Các constructor đó bao gồm:

* Hợp khái niệm (ký hiệu bằng chữ $\sqcup$) được viết là $C \sqcup D$, và được diễn dịch là:

$$(C \sqcup D)^I = C^I \cup D^I.$$

Ví dụ: mô tả nhạc công là nhạc sỹ hoặc nghệ sỹ:

Composer $\sqcup$ Performer

* Lượng tử tồn tại (ký hiệu bằng chữ $\exists$) viết là $\exists R.C$, và được dịch là:

$$(\exists R.C)^I = \{a \in 4^I \mid \exists b.(a,b) \in R^I \wedge b \in C^I\}$$

* Giới hạn số lượng (ký hiệu bằng chữ N) được viết là $\geq nR$ (giới hạn nhỏ nhất) và là $\leq nR$ (giới hạn lớn nhất), n là một số nguyên không âm. Nó được diễn dịch như sau:

$$(\geq nR)^I = \{a \in 4^I \mid |\{b \mid (a,b) \in R^I\}| \geq n\}$$

$$\text{và } (\leq nR)^I = \{a \in 4^I \mid |\{b \mid (a,b) \in R^I\}| \leq n\}$$

Ví dụ: **Person u (≤ 1 hasChild t ≥ 3 hasChild)**

* Phủ định khái niệm (ký hiệu bằng chữ $\neg$) viết là $\neg C$, diễn dịch là:

$$(\neg C)^I = 4^I \setminus C^I$$

Ví dụ: ta có Female là bù của Male: $\neg \text{Male}$

Ngôn ngữ AL mở rộng là ngôn ngữ AL khi ta thêm vào một hoặc vài constructor vừa nêu. Ta đặt tên cho từng ngôn ngữ mở rộng bằng cách thêm các ký tự:

$$AL[U][E][N][C]$$

Ví dụ về mô tả khái niệm bằng AL mở rộng như sau:

Person u ($\neg 1$ hasChild t (≥ 3 hasChild u $\exists$ hasChild.Male))

Ví dụ nêu trên mô tả khái niệm người có nhiều nhất 1 con hoặc ít nhất 3 con đồng thời có con gái.

Ngôn ngữ mở rộng ALL có thể biểu diễn bằng ALL thông qua dạng phủ định vì:

$C \text{ t } D$ và $:(D \text{ u } : D)$ là tương đương nhau

hoặc ALE cũng có thể biểu diễn bằng ALL vì:

$\exists R.C$ và $:\forall R.:C$ là tương đương nhau.

Vì vậy ta có thể viết ALL thay vì viết ALUE và viết ALCN thay vì viết ALUEN.

1.2.4. Ngôn ngữ mô tả là tập con của logic vị từ bậc nhất

Ngữ nghĩa của các khái niệm xác định ngôn ngữ mô tả là phân đoạn của logic vị từ bậc nhất. Khi diễn dịch I lần lượt áp vào tất cả các khái niệm và vai trò nguyên tố ta được các vị từ không ngôi từ các khái niệm và vị từ hai ngôi từ các vai trò. Khái niệm C bất kỳ được diễn dịch vào công thức logic vị từ $\phi C(x)$ bằng một biến x:

Một khái niệm nguyên tố A được chuyển thành công thức $A(x)$, các phép giao, hợp, phủ định được diễn dịch vào $\phi C(x)$ và R là một vai trò nguyên tố thì các lượng từ tồn tại, với mọi được chuyển theo dạng công thức:

$$\phi \exists R.C(y) = \exists x.R(y,x) \wedge \phi C(x)$$

$$\phi \forall R.C(y) = \forall x.R(y,x) \rightarrow \phi C(x)$$

ở đây y là một biến mới; giới hạn số lượng được biểu diễn theo công thức:

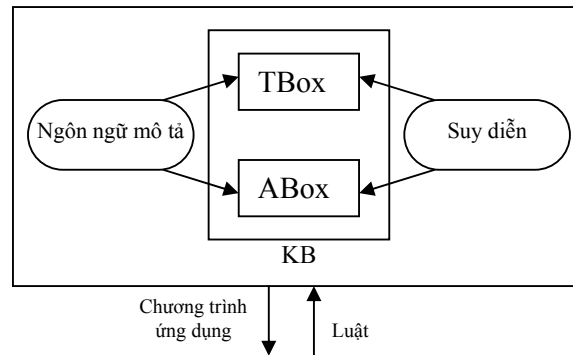
$$\begin{aligned} \phi_n R(x) &= \exists y_1, \dots, y_n. R(x, y_1) \wedge \dots \wedge R(x, y_n) \wedge \bigwedge_{i < j} y_i \neq y_j \\ \phi_n R(x) &= \forall y_1, \dots, y_{n+1}. R(x, y_1) \wedge \dots \wedge R(x, y_n) \rightarrow y_n = y_{n+1} \end{aligned}$$

1.3. HỆ CƠ SỞ TRI THỨC

1.3.1. Kiến trúc hệ logic mô tả

Như ta biết rằng hệ logic mô tả là các hệ thống thông tin có sử dụng logic mô tả để biểu diễn tri thức. Các hệ này sử dụng khả năng biểu diễn mạnh mẽ của logic mô tả, kết hợp với các thủ tục suy diễn để tạo nên tính linh hoạt của chúng. Nhờ vào logic mô tả người ta thiết lập lên những hệ thống khái niệm của lĩnh vực ứng dụng.

Một hệ cơ sở tri thức được biểu diễn bằng logic mô tả chứa đựng hai thành phần chính. Thành phần thứ nhất đó là “bộ thuật ngữ” (TBox), nơi chứa đựng các khái niệm được xây dựng bằng ngôn ngữ mô tả; thành phần thứ hai đó là “bộ khẳng định” (ABox) là nơi chứa các khẳng định hay nói cụ thể hơn là các mô tả về thế giới. Bên cạnh đó, bằng các dịch vụ suy luận ta có thể nhận được những tri thức đúng đắn. Hình 1.1 minh họa kiến trúc chung của hệ logic mô tả.



Hình 1.1: Kiến trúc hệ logic mô tả

1.3.2. Bộ thuật ngữ (TBox)

Bộ thuật ngữ (TBox) được sử dụng để lưu các thuật ngữ. Đó là các khái niệm phức được định nghĩa thông qua các khái niệm và các vai trò nguyên tố dựa trên các constructor của ngôn ngữ AL mà hệ thống sử dụng.

1.3.2.1. Tiên đề thuật ngữ

Trường hợp thông dụng nhất tiên đề thuật ngữ có dạng:

$$C \sqcup D (R \sqcup S) \text{ hoặc } C \sqcap D (R \sqcap S).$$

Trong đó C, D là các khái niệm; R, S là các vai trò. Tiên đề thứ nhất ($C \vee D$ ($R \vee S$)) được gọi là bao hàm; tiên đề thứ hai ($C \wedge D$ ($R \wedge S$)) được gọi là tương đương.

Ngữ nghĩa của các tiên đề được xác định như sau: Một diễn dịch I thoả mãn một bao hàm $C \vee D$ nếu $C^I \subseteq D^I$, và nó thoả mãn một tương đương $C \wedge D$ nếu $C^I = D^I$. Nếu T là một tập các tiêu đề thì I thoả mãn T khi và chỉ khi I thoả mãn từng thành phần của T . Nếu I thoả mãn một tiên đề thì ta nói rằng nó là mô hình của tiên đề này. Hai tiên đề hoặc hai tập tiên đề là tương đương nếu chúng có cùng mô hình.

1.3.2.2. Định nghĩa khái niệm

Một tương đương mà về bên trái là một khái niệm nguyên tố, là một định nghĩa khái niệm. Định nghĩa khái niệm dùng một tên tượng trưng để mô tả một khái niệm phức tạp.

Ví dụ:

$$\text{Mother} \wedge \text{Woman} \sqsubseteq \exists \text{hasChild. Person}$$

$$\text{Father} \wedge \text{Man} \sqsubseteq \exists \text{hasChild. Person}$$

Ta ngụ ý mô tả ở về bên phải người đàn bà (đàn ông) có con bằng tên là Mother và Father.

Các tên tượng trưng có thể được coi như là sự rút gọn trong các mô tả khác.

Từ hai ví dụ trên ta có thể định nghĩa Parent là:

$$\text{Parent} \wedge \text{Mother} \sqsubseteq \text{Father}.$$

Tập các định nghĩa phải rõ ràng. Ta gọi tập hữu hạn các định nghĩa T là TBox nếu không có tên tượng trưng nào được định nghĩa nhiều hơn một lần.

Ví dụ:

$\text{Woman} \wedge \text{Person} \sqsubseteq \text{Female}$

Man	'	Person u : Woman
Mother	'	Woman u $\exists$ hasChild.Person
Father	'	Man u $\exists$ hasChild.Person
Parent	'	Father t Mother
Grandmother	'	Mother u $\exists$ hasChild.Parent
MotherWithManyChildren	'	Mother u ≥ 3 Children
MotherWithoutDaughter	'	Mother u $\forall$ hasChild.:Woman
Wife	'	Woman u $\exists$ hasHusband.Man.

Hình 1.2: TBox với các khái niệm về quan hệ gia đình

Giả sử $\mathcal{T}$ là một thuật ngữ. Ta chia các khái niệm nguyên tố xuất hiện trong $\mathcal{T}$ thành hai tập: tập tên $N_{\mathcal{T}}$ xuất hiện bên trái của các tiên đề, và tập cơ sở $B_{\mathcal{T}}$ xuất hiện bên phải của các tiên đề. Tập tên $N_{\mathcal{T}}$ được gọi là các khái niệm được định nghĩa, tập cơ sở $B_{\mathcal{T}}$ gọi là các khái niệm nguyên thủy.

Một diễn dịch cơ sở đối với $\mathcal{T}$ là một diễn dịch chỉ dịch các khái niệm cơ sở. Cho $\mathcal{J}$ là một diễn dịch cơ sở, một diễn dịch $\mathcal{I}$ mà dịch cả các khái niệm được định nghĩa là một mở rộng của $\mathcal{J}$ nếu nó có cùng miền là $\mathcal{J}$, có nghĩa là $\mathcal{I}^{\mathcal{J}} = \mathcal{J}^{\mathcal{J}}$.

Ta nói rằng $\mathcal{J}$ không nhập nhằng nếu tất cả các diễn dịch cơ sở có chính xác một mở rộng là mô hình của $\mathcal{J}$. Nói cách khác, nếu ta biết tập cơ sở nói về cái gì và $\mathcal{T}$ không nhập nhằng thì nghĩa của khái niệm được định nghĩa (tên) hoàn toàn xác định. Rõ ràng, nếu một thuật ngữ là không nhập nhằng, thì cả các thuật ngữ tương đương cũng không nhập nhằng.

Câu hỏi đặt ra là một thuật ngữ có nhập nhằng hay không? ví dụ thuật ngữ chứa tiên đề sau:

$$\text{Human}' \quad ' \quad \text{Animal } u \quad \forall \text{hasParent.Human}'$$

Ví dụ trên bao gồm một chu trình. Thông thường, chúng ta định nghĩa trong thuật ngữ Γ như sau: Cho A, B là các khái niệm nguyên tố xuất hiện trong Γ . Ta nói rằng A sử dụng trực tiếp B trong Γ nếu B xuất hiện bên phải thuật ngữ của A , Γ chứa chu trình khi và chỉ khi tồn tại một khái niệm nguyên tố trong Γ mà sử dụng lại chính khái niệm đó. Ngược lại ta gọi thuật ngữ đó là không có chu trình.

1.3.2.3. Mở rộng bộ thuật ngữ

Nếu một thuật ngữ T không có chu trình thì nó xác định. Ta có thể mở rộng các định nghĩa trong T thông qua việc thay thế các khái niệm được định nghĩa xuất hiện ở bên phải của các tiên đề bằng các mô tả tạo ra chúng. Mục đích của việc mở rộng là nhằm đạt được bộ thuật ngữ T với hai tính chất sau:

- Mọi thuật ngữ đều ở dạng định nghĩa khái niệm;
- Vế trái của mọi thuật ngữ là một tên tượng trưng, còn vế phải chỉ chứa các khái niệm nguyên thủy.

Ví dụ:

TBox ở hình 1.2 là không có chu trình. Ta có thể mở rộng như sau:

Woman	´	Person $\sqcup$ Female
Man	´	Person $\sqcup$: (Person $\sqcup$ Female)
Mother	´	(Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person
Father	´	(Person $\sqcup$: (Person $\sqcup$ Female)) $\sqcup$ $\exists$ hasChild.Person
Parent	´	((Person $\sqcup$: (Person $\sqcup$ Female)) $\sqcup$ $\exists$ hasChild.Person) $\sqcap$ ((Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person)
Grandmother	´	((Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person) $\sqcup$ $\exists$ hasChild.(((Person $\sqcup$: (Person $\sqcup$ Female)) $\sqcup$ $\exists$ hasChild.Person) $\sqcap$ ((Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person)
MotherWithManyChildren	´	((Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person) $\sqcup$ ≥ 3 Children
MotherWithoutDaughter	´	((Person $\sqcup$ Female) $\sqcup$ $\exists$ hasChild.Person) $\sqcup$ $\forall$ hasChild.:(Person $\sqcup$ Female)

$\text{Wife} \quad \text{'} \quad (\text{Person} \sqcup \text{Female})$ $\sqcup \exists \text{hasHusband.}(\text{Person} \sqcup :(\text{Person} \sqcup \text{Female}))$

Hình 1.3: Khai triển TBox quan hệ gia đình trong Hình 1.2

Mệnh đề 1.1. [8] Gọi $\mathcal{T}$ là một bộ thuật ngữ không chứa chu trình và $\mathcal{T}'$ là bộ thuật ngữ mở rộng của nó, khi đó:

* $\mathcal{T}$ và $\mathcal{T}'$ có cùng các tên định nghĩa (symbol tên) và khái niệm cơ sở (Symbol cơ sở);

* $\mathcal{T}$ và $\mathcal{T}'$ tương đương nhau;

* Cả $\mathcal{T}$ và $\mathcal{T}'$ đều xác định.

Chứng minh: Cho $\mathcal{T}_1$ là một thuật ngữ. Giả sử $A \text{ ' } C$ và $B \text{ ' } D$ là các định nghĩa trong $\mathcal{T}_1$ để B xuất hiện trong C . Cho C' là một khái niệm thu được bằng việc thay thế các sự kiện của B trong C bằng D , cho $\mathcal{T}_2$ là thuật ngữ thu được bằng việc thay thế định nghĩa $C \text{ ' } D$ với $A \text{ ' } C'$ trong $\mathcal{T}_1$, khi đó cả hai thuật ngữ có cùng symbol tên và symbol cơ sở. Hơn nữa thu được $\mathcal{J}_2$ bằng việc thay thế tương đương $\mathcal{J}_1$, vậy cả hai thuật ngữ có cùng mô hình, khi đó ta thu được $\mathcal{T}'$ từ $\mathcal{T}$ thông qua việc thay thế nói trên.

Giả sử $\mathcal{J}$ là một diễn dịch của các symbol cơ sở. Ta mở rộng $\mathcal{J}$ thành một diễn dịch $\mathcal{I}$ tác động lên symbol tên theo cách thiết lập $A^{\mathcal{I}} \text{ ' } C'^{\mathcal{I}}$, nếu $A \text{ ' } C'$ là định nghĩa của A trong $\mathcal{T}'$. Rõ ràng, $\mathcal{I}$ là một mô hình của $\mathcal{T}'$ đồng thời cũng là mở rộng của $\mathcal{J}$. Điều đó có nghĩa là $\mathcal{T}$ được xác định. Hơn nữa, $\mathcal{T}$ hoàn toàn xác định khi đó $\mathcal{T}$ tương đương $\mathcal{T}'$.

Tất nhiên cũng có các thuật ngữ có chu trình mà vẫn xác định. Ví dụ:

$$A \text{ ' } \forall R.B \text{ t } \exists R.(A \sqcup A)$$

Tuy nhiên, $\exists R.(A \sqcup A)$ tương đương với khái niệm đáy nên ví dụ trên tương đương với tiên đề không có chu trình:

$$A \text{ ' } \forall R.B$$

1.3.2.4. Định quy

Giả sử ta muốn mô tả khái niệm về người đàn ông chỉ có con cháu trai (Man who has only male offspring) viết ngắn gọn là MOMO. Trường hợp đặc biệt ông ta chỉ có con trai (Man who has only son) viết tắt là MOS. MOS được định nghĩa không có chu trình như sau:

$$\text{MOS} = \text{Man} \wedge \forall \text{hasChild}.\text{Man}$$

Còn đây là định nghĩa đệ quy khái niệm người đàn ông chỉ có con cháu trai:

$$\text{MOMO} \hat{=} \text{Man} \wedge \forall \text{hasChild}.\text{MOMO}$$

Chu trình xuất hiện khi ta muốn mô hình hoá cấu trúc đệ quy như trên.

Ta xem xét tiếp ví dụ đệ quy trong biểu diễn cây nhị phân: Giả sử có tập các đối tượng là các cây (Tree) và các quan hệ nhị phân có cây con (hasBranch) giữa các đối tượng liên quan giữa một cây và cây con ta có:

$$\text{BinaryTree} \hat{=} \text{Tree} \wedge \neg \text{hasBranch} \wedge \forall \text{hasBranch}.\text{BinaryTree}.$$

1.3.2.5. Thuật ngữ với các tiên đề bao hàm

Có nhiều khái niệm mà chúng ta không thể định nghĩa chúng một cách hoàn thiện. Trong trường hợp như vậy, ta vẫn có thể biểu diễn các trạng thái cần thiết bằng việc sử dụng bao hàm. Ta còn gọi bao hàm là tổng quát hoá.

Ví dụ: Nếu một người xây dựng tri thức nghĩ rằng định nghĩa “Woman” trong ví dụ ở Hình 1.2 là không thoả đáng ($\text{Woman} \hat{=} \text{Person} \wedge \text{Female}$), nhưng anh ta cũng cảm thấy rằng không thể định nghĩa khái niệm “Woman” một cách chi tiết hơn, anh ta có thể quy định rằng tất cả phụ nữ trên đời này là người bằng sự tổng quát hoá:

$$\text{Woman} \vee \text{Person}$$

Một tập tiên đề Γ là một thuật ngữ được tổng quát hoá nếu về bên trái của nó là một khái niệm nguyên tố và với tất cả các khái niệm nguyên tố thì có nhiều nhất một tiên đề xuất hiện ở bên trái.

Ta sẽ chuyển thuật ngữ bị tổng quát hoá $\bar{T}$ sang thuật ngữ thường $\overline{\bar{T}}$ chỉ chứa các định nghĩa để $\overline{\bar{T}}$ tương đương với $\bar{T}$ theo nghĩa sau: Ta thu được $\overline{\bar{T}}$ từ $\bar{T}$ bằng cách lựa chọn symbol cơ sở $\overline{\bar{A}}$ cho tất cả các phép chuyên biệt hoá $A \vee C$ trong $\bar{T}$, và bằng việc thay thế chuyên biệt hoá $A \vee C$ bằng định nghĩa $A \vee \overline{\bar{A}} \vee C$. Thuật ngữ $\overline{\bar{T}}$ là chuẩn hoá của $\bar{T}$.

Nếu một TBox chứa chuyên biệt hoá như ví dụ: **Woman $\vee$ Person**, thì chuẩn hoá chứa định nghĩa là **Woman $\vee \overline{\bar{\text{Woman}}} \vee \text{Person}$** .

Mệnh đề 1.2. [8] Cho $\bar{T}$ là một thuật ngữ khái quát hoá và $\overline{\bar{T}}$ là chuẩn hoá của $\bar{T}$, khi đó:

- * Tất cả mô hình của $\overline{\bar{T}}$ là mô hình của $\bar{T}$
- * Đối với tất cả mô hình I của $\bar{T}$ có một mô hình $\overline{\bar{I}}$ của $\overline{\bar{T}}$ mà có cùng miền như I , khớp với I về các khái niệm và vai trò nguyên tố trong $\bar{T}$.

1.3.3. Bộ khẳng định (ABox)

Ngoài bộ thuật ngữ TBox vừa trình bày, thành phần thứ hai của cơ sở tri thức là bộ khẳng định ABox. Bằng bộ khẳng định người ta biểu diễn các cá thể. Ta ký hiệu các cá thể là những ký tự a, b, c . Dùng các khái niệm C, D và vai trò R ta có thể tạo ra các khẳng định theo hai loại ABox là:

$$C(a), \quad R(b,c)$$

Loại thứ nhất $C(a)$ được gọi là khẳng định khái niệm; loại thứ hai $R(b,c)$ được gọi là khẳng định vai trò.

Khẳng định khái niệm cho biết một cá thể thuộc vào khái niệm nào, còn khẳng định vai trò thể hiện mối quan hệ giữa hai cá thể (c gọi là Filler của vai trò R đối với b).

Ví dụ: Nếu ta có PETER, PAUL, MARY, HARRY là tên các cá thể và ta có ABox sau:

MotherWithoutDaughter (MARY)

Father (PETER)
hasChild (MARY, PETER)
hasChild (PETER, HARRY)
hasChild (MARY, PAUL)

Hình 1.4: Bộ khẳng định (ABox)

ABox trong hình 4 biểu diễn rằng: MARY là một người mẹ không có con gái, PETER là một người cha, MARY có các con là PETER và PAUL, PETER có con là HARRY.

Xem xét một cách đơn giản ta thấy, ABox có thể xem như một minh dụ (instance) của cơ sở dữ liệu quan hệ với các quan hệ chỉ là một ngôi hoặc hai ngôi.

Ta xác định nghĩa của ABox bằng việc mở rộng các diễn dịch đến tên cá thể. từ giờ trở đi, một diễn dịch $I = (I^I, \cdot^I)$ không chỉ ánh xạ các khái niệm và vai trò nguyên tố, mà còn ánh xạ từng tên cá thể a vào phần tử $a^I \in I^I$. Giả sử rằng tên các cá thể là khác nhau và biểu diễn các đối tượng khác nhau, thì phép ánh xạ này ánh xạ giả định tên duy nhất (UNA). Nếu a, b là tên khác nhau thì $a^I \neq b^I$. Diễn dịch I thoả mãn khẳng định khái niệm $C(a)$ nếu $a^I \in C^I$, và thoả mãn khẳng định vai trò $R(a,b)$ nếu $(a^I, b^I) \in R^I$.

Một diễn dịch thoả mãn ABox A nếu nó thoả mãn từng khái niệm trong A . Trong trường hợp như vậy ta nói rằng I là mô hình của bộ khẳng định ABox.

Khi I thoả mãn một khẳng định / hoặc một ABox A đối với TBox T , nếu nó là mô hình của / hay của A thì nó là mô hình của T .

Như vậy, một mô hình của A và T là một trừu tượng của thế giới cụ thể, ở đó các khái niệm được diễn dịch thành các tập con của miền xác định bởi TBox.

1.3.4. Cá thể

Đôi khi, các cá thể không những được dùng trong ABox mà còn trong ngôn ngữ mô tả, vì vậy người ta đưa ra các bộ tạo khái niệm (constructor) dùng các cá thể xuất hiện trong hệ thống. Một trong những constructor cơ bản đó là “tập hợp”, viết là:

$$\{a_1, \dots, a_n\}$$

trong đó $a_1, \dots, a_n$ là tên các cá thể. Tập cá thể được diễn dịch là:

$$\{a_1, \dots, a_n\}^I = \{a_1^I, \dots, a_n^I\}.$$

Ví dụ, bằng tập hợp trong ngôn ngữ mô tả ta có thể định nghĩa khái niệm các thành viên thường trực hội đồng bảo an liên hiệp quốc là $\{\text{CHINA}, \text{FRANCE}, \text{RUSSIA}, \text{UK}, \text{USA}\}$.

Từ diễn dịch trên ta thu được các khái niệm $\{a_1, \dots, a_n\}$ và $\{a_1\} \sqsubseteq \dots \sqsubseteq \{a_n\}$ là tương đương nhau.

Một constructor khác xử lý tên cá thể đó là constructor “fill” cho các vai trò, viết là:

$$R : a,$$

Ngữ nghĩa của constructor này được định nghĩa là:

$$(R : a)^I = \{d \in 4^I \mid (d, a^I) \in R^I\},$$

nghĩa là $R : a$ đại diện cho tập các đối tượng mà a là Filler của vai trò R .

Từ công thức trên ta thấy rằng, nếu với tập hợp đơn thì $R : a$ và $\exists R. \{a\}$ là tương đương.

Lưu ý rằng “fill” cho phép ta biểu diễn các khẳng định vai trò thông qua các khẳng định khái niệm: một diễn dịch thoả mãn $R(a,b)$ khi và chỉ khi nó thoả mãn $(\exists R. \{b\})(a)$.

1.3.5. Suy luận

Hệ biểu diễn tri thức dựa trên logic mô tả có thể thực hiện các dạng suy luận đặc biệt. Như đã trình bày, hệ cơ sở tri thức bao gồm TBox và ABox có ngữ nghĩa tương đương với tập hợp các tiên đề trong logic vị từ bậc nhất. Như vậy, giống như bất kỳ tập hợp tiên đề nào khác, nó cũng chứa tri thức tiềm ẩn mà bằng suy luận có thể làm cho nó rõ ràng. Ví dụ, từ TBox trong Hình 1.2 và ABox trong hình 1.4 người ta có thể kết luận rằng Mary là người bà, mặc dù tri thức này không được biểu diễn rõ ràng như một khẳng định.

Dạng suy luận khác được thực hiện bằng hệ thống logic mô tả được định nghĩa như các lập luận logic. Sau đây ta sẽ thảo luận các lập luận này, trước hết lập luận đối với các khái niệm, sau đó đối với TBox và ABox, cuối cùng lập luận đồng thời trên TBox và ABox.

1.3.5.1. Lập luận đối với khái niệm

Khi mô hình một miền ứng dụng, ta xây dựng TBox gọi là $\mathcal{T}$ bằng cách định nghĩa các khái niệm mới, kiểm tra “bài toán thoả” của các khái niệm đó được coi là suy luận mấu chốt. Một số suy luận quan trọng khác có thể rút gọn về “bài toán thoả”. Chẳng hạn để xác định mô hình là đúng hoặc để tối ưu hoá câu hỏi được thiết lập là những khái niệm, ta cần biết khái niệm nào bao quát hơn khái niệm nào, ta gọi đó là “bài toán bao hàm”. Một khái niệm C được bao hàm bởi khái niệm D nếu trong tất cả các mô hình của $\mathcal{T}$ có tập ký hiệu bởi C là một tập con của tập ký hiệu bởi D . Tiếp đến ta quan tâm đến mối quan hệ giữa các khái niệm là “bài toán tương đương” và “bài toán không giao”. Các bài toán này được định nghĩa một cách hình thức như sau:

Cho $\mathcal{T}$ là một TBox:

- **Bài toán thoả:** Một khái niệm C là thoả mãn theo $\mathcal{T}$ nếu như tồn tại một mô hình $\mathcal{I}$ của $\mathcal{T}$ mà $C^{\mathcal{I}} \neq \perp$. Ta cũng nói rằng khi đó $\mathcal{I}$ là mô hình của C .

- **Bài toán bao hàm:** Một khái niệm C bị bao hàm bởi khái niệm D theo $\mathcal{T}$, nếu $C^I \subseteq D^I$ với mọi mô hình I của $\mathcal{T}$. Khi đó ta ký hiệu là $C \sqsubseteq_{\mathcal{T}} D$ hoặc $\mathcal{T} \models C \sqsubseteq D$.
- **Bài toán tương đương:** Hai khái niệm C và D là tương đương theo $\mathcal{T}$ nếu $C^I = D^I$ với mọi mô hình I của $\mathcal{T}$. Khi đó ta ký hiệu là $C \equiv_{\mathcal{T}} D$ hoặc $\mathcal{T} \models C \equiv D$.
- **Bài toán không giao:** Hai khái niệm C và D là không giao nhau theo $\mathcal{T}$ nếu như $C^I \cap D^I = \emptyset$ với mọi mô hình I của $\mathcal{T}$.

Xét ví dụ trong Hình 1.2, Person bao hàm Woman, cả Woman và Parent bao hàm Mother, Mother bao hàm Grandmother. Hơn nữa, các cặp Woman và Man, Father và Mother là không giao nhau.

Mệnh đề 1.3. [8] Chuyển về bài toán bao hàm

Xét hai khái niệm C và D:

- C không thoả mãn, C bị bao hàm bởi ?.
- C và D tương đương, C bị bao hàm bởi D đồng thời D bị bao hàm bởi C.
- C và D không giao nhau, $C \sqcup D$ bị bao hàm bởi ?.

Mệnh đề 1.4. [8] Chuyển về bài toán không thoả

Xét hai khái niệm C và D:

- C bị bao hàm bởi D, $C \sqcup D$ là không thoả mãn.
- C và D là tương đương, cả $(C \sqcup D)$ và $(\neg C \sqcup D)$ là không thoả mãn.
- C và D không giao nhau, $C \sqcup D$ là không thoả mãn.

Mệnh đề 1.5. [8] Xét hai khái niệm C và D, các trường hợp sau đây là tương đương nhau:

- C không thoả mãn;
- C bị bao hàm bởi ?;

- C và $?$ là tương đương;
- C và $>$ không giao nhau.

1.3.5.2 Loại trừ TBox

Vấn đề tiếp theo trong suy luận là loại trừ TBox, vì sự có mặt của bộ thuật ngữ trong các thủ tục suy diễn chỉ làm phức tạp thêm cho các thủ tục này. Người ta loại bỏ ảnh hưởng của TBox trong các bài toán quyết định bằng cách sử dụng TBox mở rộng. Vì như ta đã biết, mở rộng của TBox chỉ chứa các tiên đề khái niệm với vế trái là các khái niệm mới (các symbol tên), còn vế phải là các khái niệm nguyên thủy và/hoặc vai trò nguyên thủy (các symbol cơ sở). Như vậy, với một khái niệm C cho trước, thông qua mở rộng TBox, ta có được một biểu thức khái niệm đầy đủ của C chỉ chứa các khái niệm và vai trò nguyên thủy. Xét ví dụ trong Hàng 1.2, khái niệm mở rộng của **Father** sẽ là:

Person u : (Person u Female) u $\exists$ hasChild.Person

Giả sử C' là mở rộng của C , ta có thể dễ dàng rút ra một số lập luận như sau:

- $C \sqsubseteq_T C'$;
- C là thỏa mãn theo T khi và chỉ khi C' thỏa mãn;
- Nếu D là một khái niệm khác thì ta cũng có $D \sqsubseteq_T D'$. Như vậy, $C \sqcup_T D$ khi và chỉ khi $C' \sqcup_T D'$, và $C \sqcap_T D$ khi và chỉ khi $C' \sqcap_T D'$. Khi mà C' và D' chỉ chứa các symbol cơ sở thì:
 - o $\top \sqsubseteq C \sqcup D$ khi và chỉ khi $\top \sqsubseteq C' \sqcup D'$;
 - o $\top \sqsubseteq C \sqcap D$ khi và chỉ khi $\top \sqsubseteq C' \sqcap D'$.
- tương tự, nếu C và D không giao nhau khi và chỉ khi C' và D' không giao nhau.

Tóm lại, mở rộng khái niệm đối với một TBox không có chu trình cho phép ta loại trừ TBox trong vấn đề suy luận.

Ta xét ví dụ:

Để xác nhận rằng Man và Woman không giao nhau theo TBox Family, thì ta phải kiểm tra $\text{Man} \sqcup \text{Woman}$ là không thoả mãn. Bằng việc mở rộng khái niệm ta có:

$$\text{Person} \sqcup \text{Female} \sqcup \text{Person} \sqcup :(\text{Person} \sqcup \text{Female})$$

và ta dễ dàng thấy rằng khái niệm trên là không thoả mãn. Vì vậy Man và Woman không giao nhau theo TBox Family.

1.3.5.3. Lập luận đối với ABox

Ta nói rằng ABox chứa hai dạng khẳng định: khẳng định khái niệm có dạng $C(a)$ và khẳng định vai trò $R(b,c)$. Tuy nhiên biểu diễn tri thức phải hợp lệ, bởi vì nếu không thì từ quan điểm logic người ta có thể vẽ ra kết quả bất kỳ. Chẳng hạn, ABox chứa các khẳng định $\text{Mother}(\text{MARY})$ và $\text{Father}(\text{MARY})$, hệ thống có thể cho rằng trong TBox quan hệ gia đình, những câu lệnh trên là hợp lệ.

Theo ngữ nghĩa lý thuyết của mô hình chúng ta dễ dàng đưa ra định nghĩa hình thức về sự hợp lệ. Một ABox A là hợp lệ đối với TBox $\mathcal{T}$, nếu có một diễn dịch là mô hình của cả A và $\mathcal{T}$. Chúng ta nói một cách đơn giản rằng A là hợp lệ nếu nó hợp lệ đối với TBox rỗng.

Ví dụ, tập các khẳng định $\{\text{Mother}(\text{MARY}), \text{Father}(\text{MARY})\}$ là hợp lệ đối với TBox rỗng, bởi vì không có ràng buộc nào trên diễn dịch Mother và Father, hai khái niệm có thể được diễn dịch trong cùng cách là chúng có cùng thành phần chung. Tuy nhiên, các khẳng định là không hợp lệ đối với TBox quan hệ gia đình, khi mà trong mô hình của nó khái niệm Mother và Father được diễn dịch là không giao nhau.

Tương tự như đối với khái niệm, việc kiểm tra tính hợp lệ của ABox đối với TBox không có chu trình có thể quy về việc kiểm tra một ABox mở

rộng. Ta xác định mở rộng của A đối với $\mathcal{I}$ là một ABox A' thu được bằng việc thay thế các khẳng định khái niệm $C(a)$ trong A bằng các khái niệm $C'(a)$, với C' là mở rộng của C đối với $\mathcal{I}$. Trong tất cả mô hình của $\mathcal{I}$, một khái niệm C và mở rộng C' của nó được diễn dịch theo cùng cách thức. Cho nên, A' hợp lệ khi và chỉ khi A hợp lệ. Tuy nhiên, khi A' không chứa tên được định nghĩa trong $\mathcal{I}$, thì nó hợp lệ đối với $\mathcal{I}$ khi và chỉ khi bản thân nó hợp lệ.

Tóm lại ta có kết luận:

- A hợp lệ đối với $\mathcal{I}$ khi và chỉ khi mở rộng A' của nó hợp lệ.
- $A \models C(a)$ khi và chỉ khi $A \not\models \{C(a)\}$ không hợp lệ.
- C thỏa mãn khi và chỉ khi $\{C(a)\}$ hợp lệ.

1.3.5.4. Ngữ nghĩa “đóng”, ngữ nghĩa “mở”

Giữa cơ sở dữ liệu và cơ sở tri thức theo logic mô tả có sự tương đồng. Lượng đồ của cơ sở dữ liệu có thể so sánh với TBox trong logic mô tả, còn các minh dụ (instance) của cơ sở dữ liệu thực có thể được so sánh với ABox. Tuy nhiên, ngữ nghĩa của ABox khác với ngữ nghĩa thông thường của các minh dụ (instance) cơ sở dữ liệu. Các instance cơ sở dữ liệu biểu diễn chính xác một diễn dịch, đó là các lớp và các quan hệ trong lược đồ được biểu diễn bởi các đối tượng và các bộ (bản ghi) trong instance, còn một ABox biểu diễn nhiều diễn dịch khác nhau, đó là tất cả các mô hình của nó. Khi diễn dịch những thông tin vắng mặt của instance cơ sở dữ liệu ta được kết quả phủ định, còn khi diễn dịch thông tin vắng mặt của ABox cho ta biết sự thiếu vắng tri thức.

Ví dụ: Nếu chỉ có khẳng định về PETER là $\text{hasChild}(\text{PETER}, \text{HARRY})$, thì trong cơ sở dữ liệu được hiểu là một biểu diễn thực tế PETER chỉ có một người con là HARRY; trong ABox khẳng định trên lại được hiểu là HARRY là con của PETER.

Tuy nhiên, ABox có nhiều mô hình, một trong những mô hình là HARRY là người con duy nhất, mô hình khác thì HARRY có anh chị em. Để biểu diễn trong ABox HARRY là con duy nhất ta thêm vào khẳng định (≤ 1 hasChild(PETER)); nghĩa là PETER chỉ có nhiều nhất một người con.

Từ phần thảo luận vừa rồi, ta thấy nghĩa của ABox có đặc điểm là "mở" còn nghĩa truyền thống của cơ sở dữ liệu có đặc điểm là "đóng".

Để nhìn nhận rõ hơn ngữ tính mở của ngữ nghĩa trong ABox ta xét một ví dụ dựa trên câu truyện thần thoại Hy Lạp, Oedipus. Trong một ngôi làng nhỏ, câu chuyện kể lại Oedipus đã giết cha và lấy người mẹ tên là Iokaste và có con tên là Polyneikes với bà ta ra sao. Cuối cùng Polyneikes cũng có con tên là Thersandros.

Ta giả sử rằng ABox A_{oe} được biểu diễn trong Hình 1.5 biểu diễn sơ bộ sự thực này. Trong ABox khẳng định rằng Oedipus là kẻ giết cha còn Thersandros không giết cha, nó được biểu diễn bằng khái niệm nguyên tố Patricide.

hasChild(IOKASTE, OEDIPUS)
hasChild(OEDIPUS, POLYNEIKES)
hasChild(IOKASTE, POLYNEIKES)
hasChild(POLYNEIKES, THERSANDROS)
Patricide(OEDIPUS)
:Patricide(THERSANDROS)

Hình 1.5: ABox A_{oe} về câu truyện Oedipus

Giả sử ta muốn biết xem Iokaste có con là kẻ giết cha và bản thân người con này không phải là kẻ giết cha. Ta có thể biểu diễn vấn đề đó như sau:

$$A_{oe} \models (\exists \text{hasChild.}(\text{Patricide} \sqcup \exists \text{hasChild.}:\text{Patricide}))(IOKASTE)?$$

Ai đó có thể gợi ý suy luận như sau: Iokaste có hai người con trong ABox. Một người là Oedipus là kẻ giết cha. Ông ta có một người con là Polyneikes. Nhưng không có gì cho ta biết rằng Polyneikes không phải là kẻ giết cha. Như vậy, Oedipus không phải là người con mà ta đang tìm. Người thứ hai là Polyneikes, nhưng không có gì cho ta biết Polyneikes là kẻ giết cha. Như vậy Polyneikes cũng không phải là người con mà ta đang tìm. Dựa vào lập luận này người ta cho là khẳng định về Iokaste không được kế thừa.

Tuy nhiên, lập luận đúng thì khác. Tất cả các mô hình của A_{oe} có thể được chia làm hai lớp, một lớp trong đó nói Polyneikes là kẻ giết cha, và lớp còn lại nói Polyneikes không giết cha. Trong mô hình của dạng thứ nhất, Polyneikes, con của Iokaste, là kẻ giết cha và có con không phải là kẻ giết cha tên là Thersandros. Trong mô hình của dạng thứ hai, Oedipus, con của Iokaste là kẻ giết cha và có con không phải là kẻ giết cha tên là Polyneikes. Như vậy, trong tất cả các mô hình Iokaste có một người con là kẻ giết cha và bản thân kẻ này có một người con không phải là kẻ giết cha. Điều đó có nghĩa rằng khẳng định:

$(\exists \text{hasChild.}(\text{Patricide} \sqcup \exists \text{hasChild.}:\text{Patricide}))(\text{IOKASTE})$ thực sự được kế thừa bởi A_{oe}.

Ví dụ trên cũng nói rằng, lập luận "mở" có thể cần phải phân tích các trường hợp.

1.4. CÁC THUẬT TOÁN SUY LUẬN

1.4.1. Thuật toán bao hàm cấu trúc

Thuật toán bao hàm cấu trúc xuất phát trong hai pha. Trước hết, mô tả được kiểm tra để bao hàm được chuẩn hoá, sau đó cấu trúc cú pháp của dạng chuẩn được so sánh. Để đơn giản, ta giải thích ý tưởng trên bằng cách tiếp cận ngôn ngữ FL_0 (ngôn ngữ chỉ cho phép thực hiện phép hội ($C \cup D$) và lượng từ với mọi ($\forall R.C$)). Tiếp đến ta xử lý đến các khái niệm đầy và phép phủ định khái niệm.

Rõ ràng, FL_0 và mở rộng bằng khái niệm đầy và phủ định khái niệm là ngôn ngữ con của AL , một khái niệm mô tả FL_0 là dạng chuẩn khi và chỉ khi nó có dạng:

$$A_1 \cup \dots \cup A_m \cup R_1.C_1 \cup \dots \cup R_n.C_n$$

Trong đó $A_1, \dots, A_m$ là các khái niệm khác nhau, $R_1, \dots, R_n$ là các vai trò khác nhau, còn $C_1, \dots, C_n$ là các mô tả khái niệm FL_0 ở dạng chuẩn. Ta dễ dàng thấy rằng mô tả bất kỳ có thể chuyển được về một mô tả ở dạng chuẩn. Thực tế, mô tả $\forall R.(C \cup D)$ và $(\forall R.C) \cup (\forall R.D)$ là tương đương nhau.

Mệnh đề 1.6. [8] Cho dạng chuẩn của mô tả khái niệm FL_0 :

$$A_1 \cup \dots \cup A_m \cup \forall R_1.C_1 \cup \dots \cup \forall R_n.C_n$$

và dạng chuẩn của mô tả khái niệm $FL_0(D)$:

$$B_1 \cup \dots \cup B_k \cup \forall S_1.D_1 \cup \dots \cup \forall S_l.D_l$$

thì $C \cup D$ khi và chỉ khi phù hợp hai điều kiện sau:

- 1) Đối với mọi i , $1 \leq i \leq k$, tồn tại j , $1 \leq j \leq m$ để $B_i = A_j$
- 2) Đối với mọi i , $1 \leq i \leq l$, tồn tại j , $1 \leq j \leq n$ để $S_i = R_j$ và $C_j \cup D_i$

D_i

Ta thấy rằng tính chất trên của bao hàm là đúng đắn và đầy đủ.

Nếu ta mở rộng FL_0 bằng các constructor có thể biểu diễn các khái niệm không thoả, thì một mặt ta phải thay đổi định nghĩa dạng chuẩn, mặt khác, khi so sánh cấu trúc của các dạng chuẩn ta phải lưu ý rằng một khái niệm không thoả mãi được bao hàm bởi tất cả các khái niệm. Ngôn ngữ mở rộng đơn giản nhất của FL_0 đó là mở rộng khái niệm đầy (?), ký hiệu là $FL_?$.

Một mô tả khái niệm bằng $FL_?$ là một dạng chuẩn khi và chỉ khi là ? hoặc có dạng:

$$A_1 \sqcup \dots \sqcup A_m \sqcup \forall R_1.C_1 \sqcup \dots \sqcup \forall R_n.C_n.$$

Trong đó $A_1, \dots, A_m$ là các khái niệm khác với ?, $R_1, \dots, R_n$ là các vai trò, còn $C_1, \dots, C_n$ là các mô tả khái niệm $FL_?$ ở dạng chuẩn.

Về nguyên tắc, ta có thể tính dạng chuẩn FL_0 của mô tả (?) được xử lý như một khái niệm bình thường): $B_1 \sqcup \dots \sqcup B_k \sqcup \forall R_1.D_1 \sqcup \dots \sqcup \forall R_n.D_n$. Nếu một trong số B_i là khái niệm đầy ?, thì thay toàn bộ mô tả này bằng ?. Mặt khác, áp dụng cùng thủ tục đối với D_j . Ví dụ dạng chuẩn FL_0 của $\forall R.\forall R.B \sqcup A \sqcup \forall R.(A \sqcup \forall R.?)$ là

$$A \sqcup \forall R.(A \sqcup \forall R.(B \sqcup ?))$$

thu được dạng chuẩn $FL_?$:

$$A \sqcup \forall R.(A \sqcup \forall R.?).$$

Thuật toán bao hàm cấu trúc đối với $FL_?$ làm việc giống như đối với FL_0 , chỉ khác là khái niệm đầy ? bị bao hàm bởi mô tả bất kỳ. Ví dụ:

$$\forall R.\forall R.B \sqcup A \sqcup \forall R.(A \sqcup \forall R.?) \vee \forall R.\forall R.A \sqcup A \sqcup \forall R.A$$

khi so sánh đệ quy các dạng chuẩn $FL_?$:

$A \sqcup \forall R.(A \sqcup \forall R.?)$ và $A \sqcup \forall R.(A \sqcup \forall R.A)$ cuối cùng dẫn đến sự so sánh ? và A.

Mở rộng FL_0 bằng phủ định khái niệm có thể được xử lý tương tự. Trong khi tính dạng chuẩn, các khái niệm bị phủ định được xử lý giống như các khái

niệm thường. Tuy nhiên, nếu một khái niệm và phủ định của nó xuất hiện ở cùng mức của dạng chuẩn, thì ta thêm vào ?. Ví dụ:

$$\forall R. :A \sqcup A \sqcup \forall R. (A \sqcup \forall R. B)$$

Trước hết ta biến đổi thành

$$A \sqcup \forall R. (A \sqcup :A \sqcup \forall R. B)$$

Cuối cùng ta được

$$A \sqcup \forall R. ?$$

Đối với các mô tả phức tạp hơn, thuật toán bao hàm cấu trúc thường không đáp ứng được. Đặc biệt, thuật toán này không xử lý phép hợp, phép phủ định hoàn toàn, và lượng từ tồn tại. Để khắc phục những điểm yếu của thuật toán này, người ta đưa ra một thuật toán khá hữu dụng đó là thuật toán tableau.

1.4.2. Thuật toán tableau

Ngoài việc trực tiếp kiểm tra bao hàm mô tả khái niệm, thuật toán tableau còn dùng phép phủ định để đưa bài toán bao hàm về bài toán thoả (không thoả). Như ta đã biết trong Mệnh đề 1.4: $C \sqsubseteq D$ khi và chỉ khi $C \sqcup :D$ không thoả.

Trước khi mô tả chi tiết thuật toán tableau đối với ngôn ngữ ALCN, ta lược qua các ý tưởng bằng hai ví dụ đơn giản.

Cho A và B là các khái niệm, R là vai trò.

Ví dụ thứ nhất: giả sử ta muốn biết $\exists R.A \sqcup (\exists R.B)$ có bị bao hàm bởi $\exists R.(A \sqcup B)$ hay không, nghĩa là ta phải kiểm tra xem mô tả khái niệm $C = (\exists R.A) \sqcup (\exists R.B) \sqcup :(\exists R.(A \sqcup B))$ có thoả hay không.

Trước hết ta dùng luật de Morgan và luật lượng từ để biến đổi, ta được kết quả mô tả như sau:

$$C_o = (\exists R.A) \sqcup (\exists R.B) \sqcup \forall R. (:A \sqcap :B)$$

Đây là dạng chuẩn phủ định, nghĩa là phủ định chỉ xuất hiện trước tên khái niệm. Sau đó ta xây dựng một diễn dịch hữu hạn l để $C_0^l \neq \perp$. Nghĩa là, phải tồn tại một cá thể trong 4^l là phần tử của C_0^l .

Thuật toán tạo ra một cá thể gọi là b , và chịu sự ràng buộc $b \in C_0^l$. Khi mà C_0 là hội của ba mô tả khái niệm, điều đó nghĩa là b phải thoả mãn ba ràng buộc: $b \in (\exists R.A)^l$, $b \in (\exists R.B)^l$ và $b \in (\forall R.(:A \sqcap :B))^l$.

Từ $b \in (\exists R.A)^l$, ta có thể kết luận rằng cần phải có mặt một cá thể c để $(b, c) \in R^l$ và $c \in A^l$. Tương tự $b \in (\exists R.B)^l$ phải tồn tại một cá thể d để $(b, d) \in R^l$ và $d \in B^l$. Ta không nên giả sử $c = d$ vì có khả năng phải chịu quá nhiều ràng buộc lên các cá thể mới được đưa vào để thoả mãn lượng từ tồn tại. Như vậy, với mọi lượng từ tồn tại, thuật toán đưa vào một cá thể mới làm *filler*¹ của vai trò, và cá thể này phải thoả mãn các ràng buộc biểu diễn bởi lượng từ tồn tại đó.

Khi đó b cũng phải thoả lượng từ với mọi $\forall R.(:A \sqcap :B)$ và c, d đã được đưa vào làm Filler vai trò của R , ta thu được ràng buộc $c \in (:A \sqcap :B)^l$ và $d \in (:A \sqcap :B)^l$. Như vậy, thuật toán sử dụng các lượng từ trong tương tác với các quan hệ đã được định nghĩa để chịu sự ràng buộc mới lên các cá thể.

$c \in (:A \sqcap :B)^l$ nghĩa là $c \in (A)^l$ hoặc $c \in (B)^l$, và ta phải lựa chọn một trong các khả năng có thể. Nếu ta cho rằng $c \in (A)^l$, điều này mâu thuẫn với ràng buộc khác là $c \in A^l$, nghĩa là dẫn đến một sự mâu thuẫn hiển nhiên. Vì vậy ta phải lựa chọn $c \in (B)^l$. Tương tự, ta phải chọn $d \in (A)^l$ để thoả mãn ràng buộc $d \in (:A \sqcap :B)^l$ mà không tạo ra sự mâu thuẫn với $d \in B^l$.

Ở ví dụ trên, ta đã thoả mãn tất cả các ràng buộc mà không bắt gặp một mâu thuẫn nào. Điều đó chứng tỏ C_0 là thoả mãn và như vậy $(\exists R.A) \sqcup (\exists R.B)$ được bao hàm bởi $\exists R.(A \sqcup B)$. Thuật toán đã tạo ra một diễn dịch minh chứng rằng:

¹ Đối với kiểu ABox $R(b, c)$ người ta nói rằng c là filler của vai trò R đối với b

$4^l = \{a, b, c\}$; $R^l = \{(b,c), (b,d)\}$; $A^l = \{c\}$ và $B^l = \{d\}$. Với mọi diễn dịch, $b \in C_0^l$. Nghĩa là $b \in ((\exists R.A) \sqcup (\exists R.B))^l$, nhưng $b \notin (\exists R.(A \sqcup B))^l$.

Ví dụ thứ hai: ta thêm giới hạn số lượng vào khái niệm thứ nhất của ví dụ trên, nghĩa là, ta muốn biết $(\exists R.A) \sqcup (\exists R.B) \sqcup \leq 1R$ có được bao hàm bởi $\exists R.(A \sqcup B)$ hay không. Bằng trực giác, câu trả lời là "có" khi đó $\leq 1R$ trong khái niệm thứ nhất đảm bảo R-filler trong A khớp với R-filler trong B, như vậy có một R-filler trong $A \sqcup B$. Thuật toán tableau giải bài toán thoả có thêm ràng buộc $b \in (\leq 1R)^l$. Để thoả mãn ràng buộc này thì hai R-filler c, d của b đồng nhất với nhau. Như vậy, nếu "giới hạn số lượng lớn nhất" bị vi phạm thì thuật toán phải đồng nhất hoá các *filler* vai trò khác nhau.

Trong ví dụ này, cá thể $c = d$ phải thuộc về cả A_i và B_i , mà khi đi cùng với $c = d \in (: A \sqcup B)^l$ luôn dẫn đến xung đột. Thuật toán quyết định cho ví dụ này là:

$$(\exists R.A) \sqcup (\exists R.B) \sqcup \leq 1R \vee \exists R.(A \sqcup B)$$

Các luật biến đổi của thuật toán tableau giải bài toán thoả:

Luật $\rightarrow \sqcup$ -

Điều kiện: A chứa $(C_1 \sqcup C_2)(x)$, nhưng nó không chứa cả $C_1(x)$ và $C_2(x)$.

Biến đổi: $A' = A [\{ C_1(x), C_2(x) \}]$.

Luật $\rightarrow \sqcup$ -

Điều kiện: A chứa $(C_1 \sqcup C_2)(x)$, nhưng không chứa $C_1(x)$ hoặc $C_2(x)$.

Biến đổi: $A' = A [\{ C_1(x) \}]$, $A'' = A [\{ C_2(x) \}]$.

Luật $\rightarrow \exists$ -

Điều kiện: A chứa $(\exists R.C)(x)$, nhưng không có cá thể z mà làm cho $C(z)$ và $R(x,z)$ trong A.

Biến đổi:	$A' = A [\{ C(y), R(x,y) \}$ với y là một cá thể không nằm xuất hiện trong A .
Luật $\rightarrow \forall$-	
Điều kiện:	A chứa $(\forall R.C)(x)$ và $R(x,y)$, nhưng không chứa $C(y)$.
Biến đổi:	$A' = A [\{ C(y) \}$.
Luật $\rightarrow \geq$-	
Điều kiện:	A chứa $(\geq nR)(x)$, và không có các cá thể $z_1, \dots, z_n$ mà làm cho $R(x, z_i)$ ($1 \leq i \leq n$) và $z_i \neq z_j$ ($1 \leq i < j \leq n$) nằm trong A .
Biến đổi:	$A' = A [\{ R(x, y_i) \mid 1 \leq i \leq n \} [\{ y_i \neq y_j \mid 1 \leq i < j \leq n \}$ với $y_1, \dots, y_n$ là các cá thể tách biệt không xuất hiện trong A .
Luật $\rightarrow \leq$-	
Điều kiện:	A chứa các cá thể tách biệt $y_1, \dots, y_{n+1}$ mà làm cho $(\leq nR)(x)$ và $R(x, y_1), \dots, R(x, y_{n+1})$ trong A , và $y_i \neq y_j$ không có trong A với vài $i \neq j$.
Biến đổi:	Với mỗi cặp y_i, y_j mà $i > j$ và $y_i \neq y_j$ không trong có trong A , thu được ABox $A_{i,j} = [y_i/y_j]A$ từ A bằng việc thay thế từng sự kiện của y_i bằng y_j .

Hình 1.6: Luật biến đổi của thuật toán tableau giải bài toán thoả

Luật trong Hình 1.6 được áp dụng cho một tập hữu hạn các ABox $\mathcal{S}$ như sau: Lấy một phần tử A của $\mathcal{S}$ và thay thế nó bằng một ABox A' , bằng hai ABox A', A'' hoặc bằng nhiều ABox $A_{i,j}$.

Từ các luật biến đổi ta có nhận xét:

Giả sử ta thu được $\mathcal{S}'$ từ tập hữu hạn các ABox $\mathcal{S}$ bằng cách áp dụng một luật biến đổi, thì $\mathcal{S}$ là hợp lệ khi và chỉ khi $\mathcal{S}'$ hợp lệ.

Giả sử C_o là một mô tả khái niệm ALCN ở dạng chuẩn phủ định. Sẽ không tồn tại một dãy vô hạn việc áp dụng luật $\{(C_o(x_o))\} \rightarrow S_1 \rightarrow S_2 \rightarrow \dots$

Giả sử A là một ABox thuộc S_i với $i \geq 1$, thì:

- Với mọi cá thể $x \neq x_o$ xuất hiện trong A , ta có một dãy duy nhất các vai trò $R_1, \dots, R_\ell$ ($\ell \geq 1$) và một dãy duy nhất các cá thể $x_1, \dots, x_{\ell-1}$ mà $\{R_1(x_o, x_1), R_2(x_1, x_2), \dots, R_\ell(x_{\ell-1}, x)\} \in A$. Trong trường hợp này ta nói rằng x xuất hiện ở mức ℓ trong A .
- Nếu $C(x) \in A$ đối với cá thể x ở mức ℓ , thì độ sâu vai trò cực đại của C bị bao bởi độ sâu vai trò cực đại của C_o trừ đi ℓ . Tương tự, mức của cá thể bất kỳ trong A được bao bởi độ sâu vai trò cực đại của C_o .
- Nếu $C(x)$ thuộc A , thì C là một mô tả con của C_o . Tương tự, số lượng các khẳng định khái niệm khác nhau trên x được bao bởi kích thước của C_o .
- Số các vai trò kế tiếp khác nhau của x trong A (nghĩa là các cá thể y mà $R(x, y) \in A$) được bao bởi tổng số lần xuất hiện các giới hạn nhỏ nhất trong C_o cộng với số lượng các lượng từ tồn tại khác nhau trong C_o .

Bắt đầu bằng $\{(C_o(x_o))\}$, ta thu được tập ABox S' mà không còn áp dụng luật biến đổi được nữa sau khi ta đã có một số lần hữu hạn áp dụng luật biến đổi. Một ABox A được gọi là hoàn thiện khi và chỉ khi không còn luật biến đổi nào áp dụng được nữa. Tính hợp lệ của tập ABox hoàn chỉnh có thể được quyết định bằng việc tìm các *mâu thuẫn*. ABox A chứa mâu thuẫn khi và chỉ khi một trong ba tình huống sau xuất hiện:

- (1) $\{?(x)\} \in A$ với một số cá thể x ;
- (2) $\{A(x), :A(x)\} \in A$ với một số cá thể x và một số khái niệm A ;
- (3) $\{(\neg R)(x)\} \cup \{R(x, y_i) \mid 1 \leq i \leq n+1\} \cup \{y_i \neq y_j \mid 1 \leq i < j \leq n+1\} \in A$.

Hiển nhiên, một ABox mà chứa mâu thuẫn không thể hợp lệ. Do đó, nếu tất cả ABox trong $\mathcal{S}'$ chứa mâu thuẫn, thì $\mathcal{S}'$ là không hợp lệ, và như vậy $\{C_o(x_o)\}$ cũng không hợp lệ. C_o không thoả. Tuy nhiên, nếu một trong các ABox hoàn chỉnh trong $\mathcal{S}'$ không có mâu thuẫn thì $\mathcal{S}'$ là hợp lệ. Điều đó dẫn đến $\{C_o(x_o)\}$ hợp lệ và như vậy thì C_o thoả mãn.

Một ABox A hoàn chỉnh và không xung đột là một mô hình. Các luật sẽ được áp dụng lên trên A cho đến khi không còn luật nào có thể áp dụng nữa. Ta gọi đây là quá trình mở rộng A . Việc thực hiện theo thuật toán này cho phép ta thu được một bộ khẳng định đầy đủ của A là $\tilde{A}$. Khi ta phát hiện ra mâu thuẫn trong $\tilde{A}$ có nghĩa là A không thoả mãn hay nói cách khác ta đã đưa ra được câu trả lời cho câu hỏi rằng $C \vee D$ là đúng hay sai.

Để kết thúc phần này, ta xét một ví dụ đơn giản, trong đó có sử dụng các khái niệm được đưa ra trong ví dụ ở Hình 1.2.

Ví dụ: Chứng minh rằng

Mother $\vee$ Parent

Hay **Mother $\vee$ (Mother $\wedge$ Father)**

Lúc này ta chỉ xét một TBox đơn giản là

$$\mathcal{T} = \{\text{Parent} \sqsubseteq \text{Father} \sqcap \text{Mother}\}$$

Áp dụng luật de Morgan và luật $\rightarrow \sqcup$ - ta có dãy biến đổi sau:

A	$h(\text{Mother } u : (\text{Father } \sqcap \text{Mother}))(x)i$
	$h\text{Mother}(x)i, h:(\text{Father } \sqcap \text{Mother})(x)i$
	$h\text{Mother}(x)i, h:\text{Father } u : \text{Mother}(x)i$
$\tilde{A}$	$h\text{Mother}(x)i, h:\text{Father}(x)i, h:\text{Mother}(x)i$

Hình 1.7: Ví dụ chứng minh Mother $\vee$ Parent

Có thể nhận thấy rằng mâu thuẫn đã xuất hiện giữa hai khẳng định $\text{hMother}(x)$ và $\text{h:Mother}(x)$ trong $\tilde{A}$. Điều đó chứng tỏ rằng $\text{Mother} \vee \text{Parent}$ là đúng.

1.5. MỞ RỘNG NGÔN NGỮ MÔ TẢ

Ở trên ta đã được biết đến ngôn ngữ ALCN là một ngôn ngữ logic mô tả mẫu. Đối với nhiều ứng dụng, khả năng biểu diễn của ALCN là không đáp ứng được. Vì vậy, nhiều constructor để mở rộng ngôn ngữ logic đã được đưa ra. Trong phần này ta xem xét các mở rộng quan trọng của logic mô tả. Đó là các constructor mới được dùng để xây dựng các vai trò phức tạp, đồng thời ta cũng thảo luận về khả năng biểu diễn của các giới hạn số lượng.

1.5.1. Các constructor vai trò

Khi các vai trò được diễn dịch ngữ nghĩa là các quan hệ nhị phân, ta có thể dùng các toán tử trên các quan hệ nhị phân (như toán tử bool, hợp thành, đảo vai trò) làm các constructor để thiết lập các vai trò. Cú pháp và ngữ nghĩa của các constructor này có thể được định nghĩa như sau:

Tất cả các tên vai trò là một mô tả vai trò (vai trò nguyên tố), và nếu R , S là các mô tả vai trò, thì $R \sqcup S$ (phép giao), $R \sqcap S$ (phép hợp), $\neg R$ (phép phủ định), $R \circ S$ (phép hợp thành), R^- (đảo vai trò) cũng là các mô tả vai trò.

Một diễn dịch I cho trước được mở rộng cho các mô tả vai trò phức tạp như sau:

- (1) $(R \sqcup S)^I = R^I \cup S^I$, $(R \sqcap S)^I = R^I \cap S^I$, $(\neg R)^I = 4^I \setminus R^I$;
- (2) $(R \circ S)^I = \{(a, c) \in 4^I \mid \exists b. (a, b) \in R^I \wedge (b, c) \in S^I\}$;
- (3) $(R^-)^I = \{(b, a) \in 4^I \mid (a, b) \in R^I\}$.

Chẳng hạn, phép hợp của các vai trò **hasSon** và **hasDaughter** có thể được dùng để định nghĩa vai trò **hasChild**, đảo vai trò của **hasChild** đem đến vai trò **hasParent**.

1.5.2. Biểu diễn các giới hạn số

Trước hết, ta có thể xem xét các giới hạn số lượng (qualified number restrictions) có liên quan đến R-filler thuộc về một khái niệm cụ thể. Ví dụ, cho vai trò `hasChild`, ta có thể biểu diễn rằng số lượng của toàn bộ các đứa trẻ bị giới hạn trong khoảng nhất định, như trong khái niệm $\geq 2 \text{ hasChild } u \leq 5 \text{ hasChild}$. Giới hạn số lượng cũng được dùng để biểu diễn rằng có ít nhất 2 con trai và nhiều nhất 5 con gái như sau:

$$\geq 2 \text{ hasChild.Male } u \leq 5 \text{ hasChild.Female.}$$

Ngoài ra, ta có thể thay thế các chữ số tường minh trong giới hạn số bằng các biến đại diện cho một số nguyên bất kỳ không âm. Chẳng hạn, để định nghĩa khái niệm tất cả những người có ít nhất số con gái bằng số con trai, mà không nói tường minh người này có bao nhiêu con trai và bao nhiêu con gái:

$$\text{Person } u \geq \alpha \text{ hasDaughter } u \leq \alpha \text{ hasSon}$$

1.6. NGÔN NGỮ DATALOG

Cho đến nay đã có một số ngôn ngữ dùng để xây dựng các ứng dụng dựa trên logic mô tả như RACER, AL-Log, Datalog... Sau đây chúng ta sẽ xem xét một ngôn ngữ tiêu biểu đó là ngôn ngữ Datalog. Datalog là một ngôn ngữ truy vấn cho các cơ sở dữ liệu suy diễn. Nó là một ngôn ngữ con của Prolog. Thuật ngữ Datalog được một nhóm nghiên cứu về lý thuyết cơ sở dữ liệu đưa ra vào khoảng giữa những năm 1980.

Đánh giá truy vấn bằng Datalog là hợp lệ và hoàn chỉnh, thậm chí Datalog có thể được dùng để truy vấn các cơ sở dữ liệu lớn. Việc đánh giá truy vấn thường dùng chiến lược bottom up. Tuy nhiên Datalog chỉ mới chỉ được dùng trong việc nghiên cứu cơ sở dữ liệu, nhưng chưa thật phổ biến trong các hệ thống cơ sở dữ liệu thương mại.

1.6.1. Các khái niệm và thành phần của Datalog

Vì Datalog là một ngôn ngữ con của Prolog nên các thành phần chủ yếu của Datalog tương tự với Prolog. Bây giờ ta sẽ xem xét cụ thể các thành phần của Datalog.

- Vị từ (predicate) là một hàm với một hoặc nhiều tham số. Tham số của vị từ là phối hợp của miền vị từ.

- Biến (variable) là một tham số của vị từ. Trong Datalog có hai loại biến, biến được đặt tên và biến vô danh. Biến vô danh được ký hiệu bằng dấu "_". Trình biên dịch sẽ tự gán các định danh duy nhất cho từng biến vô danh.

- Hạng thức (term) là biến hoặc hằng. Hạng thức được gọi là phức nếu nó chứa các biểu thức (chẳng hạn biểu thức số học), ngược lại được gọi là đơn giản. Trong Datalog chỉ chứa các hạng thức đơn giản.

- Nguyên tử (atom) là vị từ trong chương trình. Nguyên tử bao gồm tên vị từ và danh sách hạng thức. Ví dụ, $p(X,Y)$. Các nguyên tử có cùng tên liên quan đến cùng vị từ. Nguyên tử được gọi là cơ sở (ground) nếu nó chỉ chứa các hằng.

Mỗi nguyên tử phải được xác lập bằng một vị từ của nó. Trong phạm vi của Datalog điều đó có nghĩa là số ngôi của vị từ và số ngôi của nguyên tử có tên của vị từ này phải trùng nhau.

- *Literal* là nguyên tử - $p(X_1, \dots, X_n)$ hoặc phủ định của nguyên tử - $\text{not } p(X_1, \dots, X_n)$. Nguyên tử được gọi là literal khẳng định, sự phủ định nguyên tử được gọi là literal phủ định.

- Sự kiện (Fact) là literal khẳng định. Sự kiện cơ sở không chứa các biến.

- Luật là tập các literal mà có nhiều nhất một literal khẳng định. Đôi khi luật được coi như tập có thứ tự các literal.

- Đích (goal) là tập các literal phủ định. Đích được gọi là đơn giản nếu nó có một literal.

- Vị từ mở rộng (Extensional predicate) là vị từ, mà miền của nó được lập trực tiếp bằng sự trợ giúp của các sự kiện cơ sở. Tập các vị từ mở rộng tạo lên cơ sở dữ liệu mở rộng (EDB).

- Vị từ tăng cường (Intensional predicate) được xác định không rõ ràng bằng sự có mặt của vị từ luật mà được tính toán khi chương trình thực hiện. Tập các vị từ tăng cường tạo nên cơ sở dữ liệu tăng cường (IDB).

- Đầu luật (vị từ đầu) là literal khẳng định của luật. Đầu của luật không thể mở rộng.

- Thân luật là tập tất cả các literal phủ định của luật.

Tất cả các biến được đề cập ở đầu luật cũng được đề cập trong các vị từ của thân luật (là các tham số).

- Mô hình là kết quả của việc thực hiện chương trình. Mô hình của chương trình logic bao gồm các sự kiện mở rộng và các sự kiện tăng cường được tính toán.

1.6.2. Cú pháp của chương trình Datalog

Chương trình Datalog có cú pháp tương tự chương trình Prolog.

Luật Datalog có cú pháp như sau:

"luật" ::= "đầu luật" :- "thân luật"

"đầu luật" ::= "vị từ"

"thân luật" ::= "vị từ" [, "thân luật"]

"vị từ" ::= "TÊN VỊ TỪ" ("danh sách tham số")

"danh sách tham số" ::= "tham số" [, "danh sách tham số"]

"tham số" ::= "HẰNG" | "biến"

"biến" ::= "TÊN BIẾN" | "biến vô danh"

"biến vô danh" ::= _

Sự kiện của Datalog là tập như sau:

"sự kiện" ::= "TÊN VỊ TỪ" ("danh sách hằng")

"danh sách hằng" ::= "HẰNG" [, "danh sách hằng"]

Đích của Datalog:

"đích" ::= ? "danh sách vị từ"

"danh sách vị từ" ::= "vị từ" [, "danh sách vị từ"]

Các thành phần của chương trình (sự kiện, luật, đích) kết thúc bằng dấu chấm.

Để giải thích rõ ràng hơn ta xét ví dụ về quan hệ gia đình.

parent(peter, mary).

parent(mary, paul).

ancestor(X, Y) :- parent(X, Y).

ancestor(X,Y) :- parent(X, Z), ancestor(Z, Y).

? ancestor(peter, X).

Trong chương trình Datalog trên chứa hai vị từ nhị phân: vị từ mở rộng parent và vị từ tăng cường ancestor. Hai sự kiện được thiết lập vị từ parent:

parent(peter, mary).

parent(mary, paul).

Vị từ ancestor được mô tả bằng hai luật:

ancestor(X, Y) :- parent(X, Y).

ancestor(X,Y) :- parent(X, Z), ancestor(Z, Y).

Bên trái dấu ":-" là đầu luật, còn bên phải dấu ":-" là thân luật. EDB bao gồm một vị từ parent, IDB bao gồm một vị từ ancestor.

Trong ví dụ có chứa một đích đơn giản:

? ancestor(peter, X).

Chương trình gồm có 3 hằng: peter, mary, paul. Chỉ các biến tham số được dùng trong các luật, trong khi đích có cả biến (X) và hằng (peter).

Mỗi nguyên tử của chương trình được thiết lập bằng một vị từ. Vì vậy, ancestor(peter, X), ancestor(X, Y), ancestor(Z, Y) và tất cả mọi đề cập đến vị từ parent chứa hai tham số (số ngôi của chúng bằng hai).

Ta thấy rằng để biểu diễn cơ sở tri thức logic mô tả bằng ngôn ngữ Datalog là việc chuyển đổi tương đối dễ dàng. Dữ liệu ABox được biểu diễn bằng các sự kiện, còn TBox ta có thể biểu diễn bằng các luật trong Datalog (Tính tương đương chuyển đổi luật của Datalog sang logic mô tả sẽ được thảo luận ở chương 4 trong thủ tục *DescriptiveSupport*).

Ngoài ra, trong Datalog ở các phiên bản mới đây còn tăng cường thêm các phép toán, nhằm hỗ trợ cho việc biểu diễn logic mô tả được thuận lợi hơn, như các phép toán phủ định, đảo vai trò (*not*, *inv*) và các phép giới hạn số lượng (*atLeastOf*, *atMostOf*).

1.7. TỔNG KẾT CHƯƠNG

Trong Chương 1 ta đã thảo luận về những khái niệm cơ bản của logic mô tả và ngôn ngữ truy vấn cơ sở tri thức Datalog. Cụ thể là:

- Ngôn ngữ ALC là ngôn ngữ cho phép ta xây dựng những khái niệm phức hợp từ những khái niệm và vai trò nguyên thủy. ALC là ngôn ngữ chuẩn, các mở rộng của ALC cung cấp cho ngôn ngữ có khả năng biểu diễn linh hoạt hơn. Các constructor được dùng để mở rộng ALC là lượng từ tồn tại ($\exists R$), lượng từ với mọi ($\forall R$), toán tử phủ định ($:$), toán tử đảo vai trò (R^-) và các lượng từ giới hạn (giới hạn nhỏ nhất ($\geq n$), giới hạn lớn nhất ($\leq m$)).
- Cùng với biểu diễn cơ sở tri thức bằng ALC thông qua các TBox và ABox, Chương này cũng đã thảo luận phép diễn dịch $\models$ được dùng để xây dựng ngữ nghĩa cho logic mô tả.
- Chương 1 cũng cung cấp các dịch vụ để giải quyết các bài toán cơ bản trên logic mô tả đó là bài toán thoả, bài toán tương đương và bài toán giao.
- Cuối Chương ta đã giới thiệu một ngôn ngữ cho phép xây dựng lên các ứng dụng dựa vào logic mô tả, đó là ngôn ngữ Datalog, một ngôn ngữ

con của Prolog. Datalog có khả năng cho phép ta xây dựng các luật (chủ yếu dựa vào biểu thức hội các hạng thức) để truy vấn các hệ cơ sở dữ liệu suy diễn.

Nội dung của Chương 1 là cơ sở lý thuyết cơ bản, đồng thời cũng đã nêu được những ưu điểm (khả năng suy diễn đệ quy, biểu diễn ngữ nghĩa mở) để ta tiếp tục hướng tới bài toán ứng dụng logic mô tả để mở rộng năng lực biểu diễn trong cơ sở dữ liệu. Ở các chương tiếp theo ta sẽ hướng tới việc ứng dụng logic mô tả vào cơ sở dữ liệu.

Chương 2. SƠ LƯỢC VỀ CƠ SỞ DỮ LIỆU

Một cơ sở dữ liệu có thể nói là một tập hợp nhất quán các dữ liệu liên quan. Cơ sở dữ liệu gần tương tự như cơ sở tri thức. Sự khác nhau chủ yếu giữa cơ sở dữ liệu và cơ sở tri thức là: đối với cơ sở dữ liệu thì người tạo tập trung vào điều tác các mô dữ liệu lớn và ổn định nhưng các dữ liệu có quan hệ đơn giản; còn cơ sở tri thức đòi hỏi phải hỗ trợ nhiều trong việc tìm kiếm câu trả lời mà không thể nói rõ được.

Trong cơ sở dữ liệu, cùng với thời gian người ta đã đưa ra nhiều mô hình dữ liệu khác nhau, mỗi mô hình đều có những ưu điểm, nhược điểm nhất định. Một trong những mô hình được người ta biết đến và sử dụng nhiều nhất tại thời điểm hiện nay là mô hình dữ liệu thực thể - quan hệ (ER).

Ở chương này ta sẽ thảo luận sơ lược về mô hình dữ liệu thực thể - quan hệ và mô hình dữ liệu hướng đối tượng, là một mô hình có triển vọng đang được tiếp tục nghiên cứu và ứng dụng (tuy nhiên nó vẫn chưa được các hãng phát triển hệ quản trị cơ sở dữ liệu tập trung vào nhiều).

2.1. MÔ HÌNH THỰC THỂ - QUAN HỆ

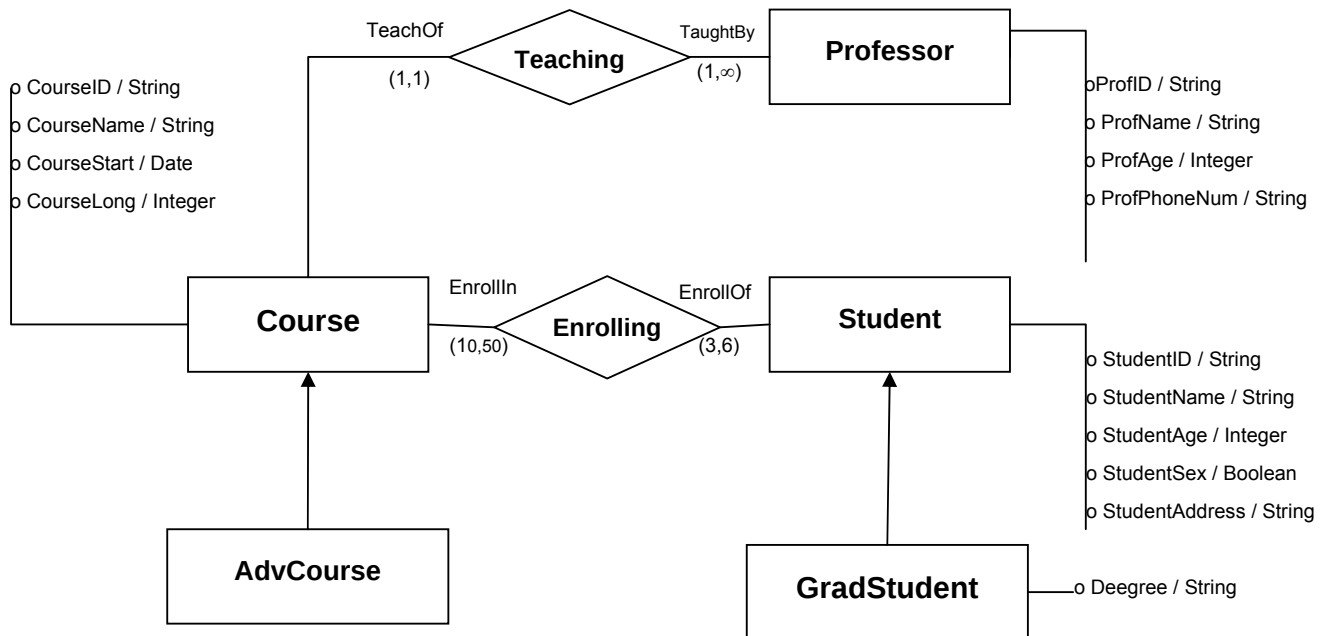
Mô hình thực thể - quan hệ do Chen giới thiệu vào năm 1976, đến nay nó đã thành chuẩn và rất phổ biến trong mô hình hoá và thiết kế cơ sở dữ liệu.

Các thành phần cơ bản của mô hình ER là các thực thể, các quan hệ và các tính chất. Một tập thực thể (hoặc một thực thể đơn) biểu thị một tập các đối tượng, được gọi là các trường hợp, mà chúng có các thuộc tính chung. Các tính chất cơ bản được mô hình thông qua các thuộc tính có các giá trị thuộc vào một trong các miền định sẵn như: Integer, String hoặc Boolean... Các tính chất mà thuộc vào các quan hệ với các thực thể khác được mô hình hoá thông qua việc tham gia vào thực thể trong quan hệ. Tập quan hệ (hoặc

một quan hệ đơn) biểu thị một tập các bộ, mỗi quan hệ biểu diễn một liên kết giữa các thành phần khác nhau của các trường hợp trong các thực thể tham gia vào quan hệ. Khi mỗi thực thể có thể tham gia vào quan hệ nhiều hơn một lần, thì ý niệm vai trò ER được đưa vào, nó miêu tả sự tham gia và một tên duy nhất được xác lập cho nó. Số ngôi của quan hệ là số lượng các vai trò của nó. Các ràng buộc chủ yếu có thể được gắn vào vai trò ER để giới hạn số lần mỗi trường hợp của thực thể được phép tham gia thông qua vai trò ER của quan hệ. Các ràng buộc như vậy có thể được dùng để đặc tả cả sự độc lập tồn tại và chức năng của quan hệ. Chúng chỉ được dùng trong dạng giới hạn, ở đó ràng buộc chủ yếu cực tiểu hoặc là 0 hay 1 và ràng buộc cực đại hoặc là 1 hoặc ∞ . Thêm vào nữa, quan hệ có tên là IS-A được dùng để diễn tả các khẳng định bao hàm giữa các thực thể, vì thế sự thừa kế các thuộc tính từ một thực thể tổng quát hơn với thực thể chuyên biệt hơn.

Một lược đồ thực thể - quan hệ $\mathcal{S}$ được xây dựng bắt đầu từ các tập ký hiệu thực thể, các ký hiệu quan hệ (mỗi ký hiệu quan hệ với một ngôi), các ký hiệu vai trò, các ký hiệu thuộc tính và các ký hiệu miền tách rời nhau từng đôi một. Mỗi ký hiệu miền $\mathcal{D}$ có một miền cơ sở định nghĩa trước $\mathcal{D}^{\text{BD}}$, chúng ta giả sử rằng miền cơ sở là miền không giao nhau. Mỗi ký hiệu thực thể là một tập các ký hiệu thuộc tính được định nghĩa, và với mỗi thuộc tính như vậy thì có một ký hiệu miền duy nhất được liên kết. Mỗi ký hiệu quan hệ của ngôi k có k liên kết với các ký hiệu vai trò, mỗi ký hiệu vai trò có một ký hiệu thực thể được liên kết và định nghĩa một quan hệ giữa các thực thể này. Những ràng buộc chủ yếu được diễn đạt bằng hai hàm $cmin_s$, từ ký hiệu vai trò đến một số nguyên không âm, $cmax_s$, từ ký hiệu vai trò đến một số nguyên dương hợp với ký hiệu đặc biệt ∞ . Quan hệ Là một (IS-A) giữa các thực thể được mô hình bằng quan hệ nhị phân v_s .

Thông thường, biểu diễn đồ hoạ mô hình thực thể - quan hệ thì các thực thể là các hình chữ nhật, các quan hệ được biểu diễn bằng các hình thoi. Thuộc tính được thể hiện bằng vòng tròn gắn với thực thể định nghĩa nó. Các vai trò được miêu tả bằng việc kết nối quan hệ với các thực thể tham gia vào quan hệ và gắn nhãn trên đường liên kết bằng các ràng buộc liên kết chính. Quan hệ IS-A giữa hai thực thể được diễn đạt bằng một mũi tên từ thực thể chuyên biệt hơn đến thực thể khái quát hơn.



Hình 2.1 Lược đồ ER

Hình 2.1 biểu diễn một lược đồ ER đơn giản mô hình hoá một trường hợp tại một trường đại học. Mô tả một phần lược đồ như sau: Thực thể Course biểu diễn các khoá học mà các sinh viên đăng ký và được các giảng viên giảng dạy. Ràng buộc được dùng để giới hạn số lượng sinh viên đăng ký vào một khoá là từ 10 đến 50, số lượng khoá học mà mỗi sinh viên có thể tham

gia trong khoảng từ 3 đến 6. Hai thực thể AdvCourse và GradStudent là chuyên biệt hoá của Course và Student.

Ngữ nghĩa của lược đồ ER có thể được đưa ra bằng việc đặc tả trạng thái cơ sở dữ liệu đặc trưng với cấu trúc thông tin được biểu diễn bởi lược đồ. Một trạng thái cơ sở dữ liệu B tương đương với một lược đồ ER S được tạo thành bởi một tập giới hạn không rỗng Δ^B , giả sử tách biệt với tất cả miền cơ sở khác, và một hàm B mà ánh xạ:

- Tất cả ký hiệu miền D vào miền cơ sở D^{B0} tương ứng,
- Tất cả thực thể E vào tập con E^B của Δ^B ,
- Tất cả thuộc tính A vào một hàm cục bộ A^B từ Δ^B đến hợp nhất, đối với tất cả các miền D, của D^{B0} ,
- Tất cả các quan hệ R vào một tập R^B của các bộ gán nhãn trên Δ^B .

Một bộ gán nhãn trên miền Δ^B là một hàm từ một tập của vai trò ER đến Δ^B .

Một bộ gán nhãn T mà ánh xạ vai trò ER U_i thành o_i , $i \in \{1, \dots, k\}$, được biểu thị là $[U_1:o_1, \dots, U_k:o_k]$. Chúng ta cũng có thể viết $T[U_i]$ để biểu thị o_i , và gọi nó là thành phần U_i của T. Các phần tử của E^B , A^B và R^B được gọi là các trường hợp của E, A, R tương ứng.

Một trạng thái cơ sở dữ liệu được xem xét là chấp nhận được nếu nó thoả tất cả các ràng buộc toàn vẹn. Một trạng thái cơ sở dữ liệu B là hợp lệ đối với lược đồ ER S, nếu nó thoả các điều kiện sau:

- Mỗi cặp thực thể E_1, E_2 với $E_1 \vee_S E_2$, thì $E_1^B \subseteq E_2^B$.
- Mỗi thực thể E, nếu E có thuộc tính A với miền D, thì mỗi trường hợp $e \in E^B$, $A^B(e)$ được định nghĩa và trong D^{B0} .

- Mỗi quan hệ R k ngôi giữa các thực thể $E_1, \dots, E_k$, mà R liên kết với cùng bằng các vai trò ER $U_1, \dots, U_k$ tương ứng, thì tất cả các trường hợp của R có dạng $[U_1:e_1, \dots, U_k:e_k]$, với $e_i \in E_i^B$, $i \in \{1, \dots, k\}$.
- Mỗi vai trò ER U được mà liên kết với quan hệ R và thực thể E, và mỗi trường hợp e của E thì

$$cmin_s(U) \leq \#\{r \in R^B \mid r[U] = e\} \leq cmax_s(U).$$

2.2. MÔ HÌNH HƯỚNG ĐỐI TƯỢNG

Mô hình dữ liệu hướng đối tượng được giới thiệu với mục đích nhằm thừa kế chuẩn cơ sở dữ liệu mà có thể tích hợp với các hệ lập trình hướng đối tượng. Nó là chủ đề trong việc nghiên cứu lĩnh vực cơ sở dữ liệu, nó dựa trên các tính năng sau:

- Nó dựa vào ý niệm nhận diện các đối tượng tại mức mở rộng và vào ý niệm của các lớp ở mức tăng cường.
- Cấu trúc của các lớp được đặc tả bằng kiểu và kế thừa.

Chúng ta định nghĩa một ngôn ngữ hướng đối tượng đơn giản theo kiểu mô hình thông thường nhất có các đối tượng phức tạp và định danh đối tượng. Mô hình hướng đối tượng được thể hiện bằng 1 ví dụ trong hình dưới đây:

```

Class Professor type-is
    Record
        ProfID: String
        ProfName: String
        ProfAge: Integer
        ProfPhoneNum: String
    End
Class Student type-is

```

```
Record  
    StudentID: String  
    StudentName: String  
    StudentAge: Integer  
    StudentSex: Boolean  
    StudentAddress: String  
End  
Class Teacher type-is  
    Union Professor, GradStudent  
End  
Class GradStudent is-a Student type-is  
    Record  
        degree:String  
    End  
Class Course type-is  
    Record  
        enrolls: Set-of Student  
        taughtby: Teacher  
        CourseStart: Date  
        CourseLong: Integer  
    End
```

Hình 2.2 Một mô hình hướng đối tượng

Một lược đồ hướng đối tượng được định nghĩa trên một tập hữu hạn các tên lớp, biểu diễn bởi ký tự C và tập hữu hạn tên các thuộc tính, biểu diễn bởi ký tự A . Một lược đồ hướng đối tượng $\mathcal{S}$ là một tập hữu hạn các định nghĩa lớp có dạng:

Class C is-a $C_1, \dots, C_k$ type-is T ,

với chính xác một định nghĩa đối với mỗi lớp C , ở đây T biểu diễn cho biểu thức kiểu được xây dựng theo cú pháp sau:

$T \mid C \mid$
 $\text{Union } T_1, \dots, T_k \text{ End} \mid$
 $\text{Set-of } T \mid$
 $\text{Record } A_1: T_1, \dots, A_k: T_k \text{ End}.$

Hình 2.2 biểu diễn một phần lược đồ hướng đối tượng tương ứng với lược đồ thực thể quan hệ trong Hình 2.1.

Mỗi định nghĩa lớp chứa các ràng buộc về các trường hợp của lớp mà nó liên quan. Phần is-a của định nghĩa lớp cho phép người ta đặc tả sự bao hàm giữa các tập trường hợp của lớp liên quan, phần type-is đặc tả kiểu cấu trúc được ấn định cho các đối tượng của lớp.

Nghĩa của lược đồ hướng đối tượng được đưa ra bởi việc đặc tả tính chất của các thể hiện trong lược đồ.

Trước hết ta định rõ đặc điểm tập các giá trị được xây dựng từ một tập các ký hiệu, gọi là các định danh đối tượng. Cho Ω là một tập hữu hạn các ký hiệu, tập V_Ω các giá trị trên Ω được định nghĩa một cách quy nạp như sau:

- μV_Ω .
- Nếu $v_1, \dots, v_k \in V_\Omega$ thì $\{|v_1, \dots, v_k|\} \in V_\Omega$.
- Nếu $v_1, \dots, v_k$ thuộc V_Ω thì $\{A_1:v_1, \dots, A_k:v_k\}$ thuộc V_Ω .
- Còn lại V_Ω rỗng.

Một *thể hiện cơ sở dữ liệu* $\mathcal{J}$ của lược đồ $\mathcal{S}$ được tạo thành bởi:

- một tập hữu hạn $\Omega^\mathcal{J}$ các định danh đối tượng;
- một phép ánh xạ $\pi^\mathcal{J}$ ấn định cho mỗi tên lớp một tập con của $\Omega^\mathcal{J}$;
- một ánh xạ $\rho^\mathcal{J}$ ấn định một giá trị trong $V_\Omega^\mathcal{J}$ cho mỗi đối tượng trong $\Omega^\mathcal{J}$.

Cho dù tập giá trị $V_{\mathbb{J}}$ mà có thể được xây dựng từ một tập $\mathbb{J}^{\downarrow}$ các định danh đối tượng là vô hạn, thì đối với một thể hiện cơ sở dữ liệu người ta chỉ cần quan tâm đến một tập con hữu hạn của $V_{\mathbb{J}}$, vì các cấu trúc hữu hạn mới có thể được lưu trong một cơ sở dữ liệu. Đối với một lược đồ hướng đối tượng $\mathbb{S}$ và một thể hiện $\mathbb{J}$ của $\mathbb{S}$, tập hữu hạn này được gọi là tập $V_{\mathbb{J}}$ các giá trị tích cực đối với $\mathbb{J}$, và được tạo bởi hợp của:

- tập các định danh đối tượng $\mathbb{J}^{\downarrow}$ và
- tập các giá trị được ấn định cho các phần tử của $\mathbb{J}^{\downarrow}$ bởi $\rho^{\downarrow}$, bao gồm các giá trị mà không liên kết rõ ràng với các định danh đối tượng, nhưng được dùng để tạo các giá trị khác.

Diễn dịch của các biểu thức kiểu trong $\mathbb{J}$ được định nghĩa thông qua một hàm diễn dịch $^{\downarrow}$. Hàm này ấn định cho mỗi biểu thức kiểu một tập con của $V_{\mathbb{J}}$ mà các điều kiện dưới đây được thoả:

$$\begin{aligned} C^{\downarrow} &= \pi^{\downarrow}(C) \\ (\text{Union } T_1, \dots, T_k \text{ End})^{\downarrow} &= T_1^{\downarrow} \cup \dots \cup T_k^{\downarrow} \\ (\text{Set-of } T)^{\downarrow} &= \{ \{v_1, \dots, v_k\} \mid k \geq 0, v_i \text{ thuộc } T^{\downarrow}, \\ &\quad \text{với } i \text{ thuộc } \{1, \dots, k\} \} \\ (\text{Record } A_1:T_1, \dots, A_k:T_k \text{ End})^{\downarrow} &= \{ [A_1:v_1, \dots, A_h:v_h] \mid h \geq k, \\ &\quad v_i \in T_i^{\downarrow}, \text{ với } i \in \{1, \dots, k\}, \\ &\quad v_j \in V_{\mathbb{J}}, \text{ với } j \in \{k+1, \dots, h\} \}. \end{aligned}$$

Để đặc tả các mô hình hướng đối tượng ta định nghĩa các thể hiện có thể được chấp nhận đối với một mô hình. Một thể hiện cơ sở dữ liệu $\mathbb{J}$ của một mô hình hướng đối tượng $\mathbb{S}$ được cho là hợp lệ nếu mỗi định nghĩa

$$\text{Class } C \text{ is-a } C_1, \dots, C_n \text{ type-is } T$$

trong $\mathbb{S}$, nó có $C^{\downarrow} \subseteq C_i^{\downarrow}$ với $i \in \{1, \dots, n\}$, và có $(C^{\downarrow}) \subseteq T^{\downarrow}$. Bởi vậy, đối với một thể hiện cơ sở dữ liệu hợp lệ, biểu thức kiểu mà xuất hiện trong lược đồ

xác định tập hữu hạn các giá trị tích cực, phải được xem xét. Việc xây dựng giá trị như vậy bị giới hạn bởi độ sâu của biểu thức kiểu.

2.3. TỔNG KẾT CHƯƠNG

Trong chương này ta đã lược qua hai mô hình dữ liệu phổ biến là mô hình thực thể - quan hệ và mô hình dữ liệu hướng đối tượng.

Trong chương tiếp theo, kết hợp các kiến thức được thảo luận ở Chương 1 và Chương 2, ta sẽ thảo luận việc mô hình hoá cơ sở dữ liệu bằng logic mô tả; đồng thời đưa ra các thuật toán biến đổi một cơ sở dữ liệu thành một cơ sở tri thức dựa trên logic mô tả.

Course $\vee \forall \text{CourseName.String } u \vee \text{CourseStart.Date } u$
 $\vee \text{CourseLong.Integer } u = 1 \text{ CourseName } u = 1$
 $\text{CourseStart } u = 1 \text{ CourseLong}$

Student $\vee \forall \text{StudentName.String } u \vee \text{StudentAge.Date } u$
 $\vee \text{StudentSex.Boolean } u \vee \text{StudentAddress.String } u$
 $= 1 \text{ StudentName } u = 1 \text{ StudentAge } u = 1 \text{ StudentSex}$
 $u = 1 \text{ StudentAddress}$

- Mỗi quan hệ R với k ngôi giữa các thực thể $E_1, \dots, E_k$ được liên kết bằng các vai trò quan hệ $U_1, \dots, U_k$ tương ứng thì:

$$\phi(R) \vee \forall \phi(U_1). \phi(E_1) \sqcap \dots \sqcap \forall \phi(U_k). \phi(E_k) \sqcap = 1 \phi(U_1) \sqcap \dots \sqcap = 1 \phi(U_k)$$

$$\phi(E_i) \vee \forall (\phi(U_i))^\perp. \phi(R), \text{ với } i \in \{1, \dots, k\}.$$

Mỗi vai trò quan hệ U liên kết với quan hệ R và thực thể E,

Nếu $m = \text{cmin}_s(U) \neq 0$, thì:

$$\phi(E) \vee \geq m(\phi(U))^\perp.$$

Nếu $n = \text{cmax}_s(U) \neq \infty$, thì:

$$\phi(E) \vee \leq n(\phi(U))^\perp.$$

Ví dụ:

TEACHING $\vee \forall \text{TaughtBy. Professor } u = 1 \text{ TaughtBy } u$
 $\vee \text{TeachOf.Course } u = 1 \text{ TeachOf}$

ENROLLING $\vee \forall \text{EnrollIn.Course } u = 1 \text{ EnrollIn } u \vee \text{EnrollOf.Student}$
 $u = 1 \text{ EnrollOf}$

Professor $\vee \text{TaughtBy}^\perp. \text{TEACHING } u \geq 1 \text{ TaughtBy}^\perp u \leq n$
 $\text{TaughtBy}^\perp$

$\text{Course} \vee \forall \text{TeachOf}^-. \text{TEACHING } u \geq 1 \text{ TeachOf}^- u \leq n$
 $\text{TeachOf}^- \vee \text{EnrollIn}^-. \text{ENROLLING } u \geq 10 \text{ EnrollIn}^-$
 $u \leq 50 \text{ EnrollIn}^-$
 $\text{Student} \vee \forall \text{EnrollOf}^-. \text{ENROLLING } u \geq 3 \text{ EnrollOf}^- u \leq 6$
 EnrollOf^-

- Mỗi cặp ký hiệu C_1, C_2 mà C_1 là một ký hiệu quan hệ hoặc một ký hiệu miền, C_2 là một ký hiệu thực thể, một ký hiệu quan hệ hoặc ký hiệu miền và $C_1 \neq C_2$, thì $\phi(C_1) \vee \neg \phi(C_2)$.

Ví dụ: Professor $\vee$: Student

Như vậy bằng việc chuyển đổi lược đồ trong Hình 2.1 ta nhận được một TBox biểu diễn lược đồ quan hệ bằng logic mô tả đầy đủ như sau:

Các khái niệm nguyên thuỷ:

String, Integer, Boolean, Date

Các vai trò nguyên thuỷ (được dùng để biểu diễn cho các thuộc tính của các thực thể):

ProfName, ProfAge, ProfPhone

CourseName, CourseStart, CourseLong

StudentName, StudentAge, StudentSex, StudentAddress

Các vai trò biểu diễn các quan hệ giữa các thực thể:

TaughtBy, TeachOf, EnrollIn, EnrollOf.

TBox biểu diễn lược đồ quan hệ:

$\text{TEACHING} \vee \forall \text{TaughtBy}^-. \text{Professor } u = 1 \text{ TaughtBy } u$
 $\forall \text{TeachOf}^-. \text{Course } u = 1 \text{ TeachOf}$

ENROLLING	v	$\forall \text{EnrollIn.Course } u = 1 \text{ EnrollIn } u \forall \text{Eof.Student } u = 1 \text{ EnrollOf}$
Professor	v	$\forall \text{TaughtBy}^-. \text{TEACHING } u \geq 1 \text{ TaughtBy}^- u \leq n$ $\text{TaughtBy}^- u \forall \text{ProfName.String } u \forall \text{ProfAge.Integer } u \forall \text{ProfPhone.String } u = 1 \text{ ProfName } u = 1 \text{ ProfAge } u = 1 \text{ ProfPhone}$
Course	v	$\forall \text{TeachOf}^-. \text{TEACHING } u \geq 1 \text{ TeachOf}^- u \leq n$ $\text{TeachOf}^- u \forall \text{EnrollIn}^-. \text{ENROLLING } u \geq 10$ $\text{EnrollIn}^- u \leq 50 \text{ EnrollIn}^- u \forall \text{CourseName.String } u \forall \text{CourseStart.Date } u \forall \text{CourseLong.Number } u = 1$ $\text{CourseName } u = 1 \text{ CourseStart } u = 1 \text{ CourseLong}$
Student	v	$\forall \text{EnrollOf}^-. \text{ENROLLING } u \geq 3 \text{ EnrollOf}^- u \leq 6$ $\text{EnrollOf}^- u \forall \text{StudentName.String } u \forall \text{StudentAge.Date } u \forall \text{StudentSex.Boolean } u \forall \text{StudentAddress.String } u = 1 \text{ StudentName } u = 1$ $\text{StudentAge } u = 1 \text{ StudentSex } u = 1 \text{ StudentAddress}$
AdvCourse	v	Course
GradStudent	v	Student $u \forall \text{Degree.String } u = 1 \text{ Degree}$
Professor	v	: Student

Hình 3.1: TBox được chuyển đổi từ lược đồ ER trong Hình 2.1

Đối với phép biến đổi được cung cấp trên đây, chúng ta có các nhận xét sau:

- Vì một lược đồ ER chỉ biểu diễn các trạng thái cần thiết của đối tượng, nên sự chuyển đổi chỉ sử dụng các khẳng định bao hàm, còn các khẳng định tương đương không cần thiết.
- Mỗi quan hệ được cụ thể hoá, được mô hình hoá bằng một khái niệm mà các trường hợp của nó biểu diễn các bộ trong quan hệ. Các khẳng định buộc từng vai trò $\phi(U)$ của quan hệ phải kết nối với chỉ một đối tượng mà biểu diễn thành phần U của bộ tương ứng.
- Các vai trò đảo và giới hạn số cũng được dùng để nắm bắt ngữ nghĩa của lược đồ ER. Chính xác hơn, chức năng của các vai trò liên kết với các vai trò ER được dùng để biểu diễn các sự kiện mà mỗi *instance* của một quan hệ được kết nối thông qua từng vai trò ER đến chính xác một *instance* của thực thể được liên kết, trong khi chúng ta sử dụng các giới hạn số trong các vai trò đảo để biểu diễn các ràng buộc của lược đồ.
- Ngay cả khi lược đồ ER không có chu trình, thì cơ sở tri thức kết quả vẫn bao gồm chu trình.
- Bằng constructor đảo vai trò, một quan hệ nhị phân được xử lý theo cách đơn giản hơn bằng việc lựa chọn một hướng đi và một ánh xạ trực tiếp quan hệ đó tới một vai trò.

Để chứng tỏ rằng phép biến đổi bảo toàn ngữ nghĩa của mô hình ER, chúng ta định nghĩa một phép ánh xạ giữa các trạng thái cơ sở dữ liệu tương ứng với mô hình ER và các diễn dịch hữu hạn của cơ sở tri thức bắt nguồn từ đó. Do cụ thể hoá các quan hệ, nên phép ánh xạ này đương nhiên không phải một – một và chúng ta cần miêu tả trước các diễn dịch của cơ sở tri thức tương ứng trực tiếp với các trạng thái cơ sở dữ liệu.

Chúng ta nói rằng một diễn dịch I của cơ sở tri thức $\phi(S)$ bắt nguồn từ lược đồ ER S là quan hệ - mô tả, nếu với tất cả các quan hệ R của S , có các vai trò $U_1, \dots, U_k$, với $d, d' \in (\phi R)^I$, chúng ta có:

$$\bigwedge_{1 \leq i \leq k} (\forall d'' \in \Delta^i. (d, d'') \in (\phi(U_i))^i \leftrightarrow (d', d'') \in (\phi(U_i))^i) \rightarrow d = d'.$$

Tính tương ứng của các mô hình ER với biểu diễn Logic mô tả. Thực chất nó không phải là ánh xạ 1 - 1, nhưng nó vẫn bảo toàn về mặt ngữ nghĩa. Tính tương ứng có thể được thiết lập bằng việc định nghĩa hai phép ánh xạ α_S và β_S như sau:

- α_S là phép ánh xạ từ các trạng thái cơ sở dữ liệu tương ứng với S tới các diễn dịch quan hệ - mô tả của $\phi(S)$, bằng các thuộc tính sau (cho β là một trạng thái cơ sở dữ liệu và $l = \alpha_S(\beta)$):
 - + Nếu β là một cơ sở dữ liệu hợp lệ, thì l là một mô hình của $\phi(S)$.
 - + Δ^l là hợp của Δ^β , tập của các bộ (tuples) và tập giá trị miền xuất hiện trong β . Tập giá trị miền xuất hiện trong β là giá trị của một số thuộc tính.
 - + Đối với mỗi E có $(\phi(E))^l = E^\beta$.
- β_S là phép ánh xạ từ các diễn dịch quan hệ - mô tả của $\phi(S)$ tới các trạng thái cơ sở dữ liệu tương ứng với S , bằng các thuộc tính sau (cho l là một diễn dịch của $\phi(S)$ và $\beta = \beta_S(l)$):
 - + Nếu l là một mô hình của $\phi(S)$, thì β là một trạng thái cơ sở dữ liệu hợp lệ.
 - + Δ^β là tập tất cả các đối tượng trong Δ^l mà không có các trường hợp của bất cứ khái niệm nguyên tố nào tương ứng với một quan hệ hoặc một miền cơ sở.
 - + Đối với mỗi thực thể E , $E^\beta = (\phi(E))^l$.

Việc tồn tại các phép ánh xạ α_S và β_S cho phép chúng ta giảm bớt vấn đề kiểm tra các thuộc tính trong việc suy luận trên cơ sở tri thức ALCQI tương ứng.

3.2. MỞ RỘNG KHẢ NĂNG BIỂU DIỄN CỦA NGÔN NGỮ MÔ HÌNH HOÁ

Các mô hình dữ liệu ngữ nghĩa nói chung và mô hình ER nói riêng, không cung cấp nhiều tính năng biểu diễn sự phụ thuộc phức tạp của dữ liệu. Logic mô tả là một trong những công cụ hiệu quả để tăng cường khả năng biểu diễn. Sau đây chúng ta sẽ xem xét các ví dụ mang nhiều ý nghĩa về khả năng tăng cường cho mô hình ER thông qua logic mô tả.

3.2.1. Tổng quát hoá thực thể

Chỉ có quan hệ trực tiếp giữa các thực thể được biểu diễn trong mô hình ER cơ sở mới là quan hệ IS-A. Một mở rộng thông dụng đó là tổng quát hoá (generalization). Để tổng quát hoá ta có thể dùng phép hợp và phép phủ định của logic mô tả.

Ví dụ: Thực tế rằng mỗi một Professor có thể là một Associate Professor hoặc là một Full Professor. Ta có thể biểu diễn thực thể Professor là tổng quát hoá của hai thực thể AssProfessor và FullProfessor. Chúng ta biểu diễn bằng AL^Q thông qua việc thêm các khẳng định sau:

Professor v AssProfessor t FullProfessor

AssProfessor v Professor u :FullProfessor

FullProfessor v Professor

Thông thường, mỗi mức tổng quát hoá biểu diễn rằng thực thể E là tổng quát hoá của các thực thể tách biệt nhau $E_1, \dots, E_n$ có thể được chuyển thành các khẳng định như sau:

$$\phi(E) \quad \vee \quad \phi(E_1) \wedge \dots \wedge \phi(E_n)$$

$$\phi(E_1) \quad \vee \quad \phi(E) \sqcup : \phi(E_2) \sqcup : \phi(E_3) \sqcup \dots \sqcup : \phi(E_n)$$

$$\phi(E_2) \quad \vee \quad \phi(E) \sqcup : \phi(E_3) \sqcup : \phi(E_4) \sqcup \dots \sqcup : \phi(E_n)$$

....

$$\begin{aligned} \phi(E_{n-1}) & \vee \phi(E) \cup \phi(E_n) \\ \phi(E_n) & \vee \phi(E) \end{aligned}$$

3.2.2. Loại các tính chất thuộc một cấu trúc IS-A

Một mở rộng quan trọng khác cần được quan tâm đó là khả năng đặc tả việc loại các thuộc tính của các thực thể thuộc cấu trúc IS-A. Đây là tính năng cốt yếu trong các mô hình hướng đối tượng. Đặc biệt, các ràng buộc có thể được loại bằng việc hạn chế khoảng giá trị và thành phần trong các quan hệ.

Ví dụ, khẳng định sau ràng buộc rằng các khoá học cao cấp (advanced courses) phải có ít nhất 5 sinh viên, nhiều nhất 15 sinh viên và những sinh viên học ở khoá này là những sinh viên đã tốt nghiệp:

$$\text{AdvCourse } v \quad \exists^{\geq 5} \text{EnrollIn}^- \cup \exists^{\leq 15} \text{EnrollIn}^- \cup \forall \text{EnrollIn}^- . \forall \text{EnrollOf.GradStudent}$$

3.3. BIỂU DIỄN MÔ HÌNH DỮ LIỆU HƯỚNG ĐỐI TƯỢNG BẰNG LOGIC MÔ TẢ

Ta thiết lập một quan hệ giữa ngôn ngữ mô tả ALCQI và ngôn ngữ hướng đối tượng đã giới thiệu ở Chương 2. Nghĩa là ta sẽ cung cấp một phép ánh xạ từ lược đồ hướng đối tượng sang cơ sở tri thức ALCQI . Ta sẽ đưa ra khái niệm **AbstractClass** để biểu diễn các lớp, hai khái niệm **RecType** và **SetType** để biểu diễn các kiểu dữ liệu tương ứng, vai trò value để mô hình hoá mối liên quan giữa lớp và kiểu, và vai trò member được dùng để chỉ rõ kiểu của các thành phần trong một tập. Hơn nữa, các khái niệm biểu diễn kiểu được coi là không giao nhau và không giao với các khái niệm biểu diễn lớp. Những ràng buộc này được biểu diễn bởi các khẳng định bao hàm thích hợp trong cơ sở tri thức.

Nói một cách hình thức, cơ sở tri thức $\psi(\mathcal{S})$ tương ứng với một lược đồ hướng đối tượng $\mathcal{S}$ chứa các khái niệm nguyên tố xác định trước là

AbstractClass, **RecType** và **SetType**, và một khái niệm $\psi(C)$ ứng với mỗi lớp C trong $\mathbb{S}$. Đồng thời nó cũng chứa các vai trò nguyên tố xác định trước là **value** và **member**; một vai trò nguyên tố $\psi(A)$ ứng với mỗi tên thuộc tính A trong $\mathbb{S}$.

Trước khi định rõ tập các khẳng định của $\psi(\mathbb{S})$ ta xác định cách mà hàm ψ ánh xạ từng biểu thức kiểu vào biểu thức khái niệm như sau:

- Tất cả lớp C được ánh xạ vào một khái niệm nguyên tố $\psi(C)$.
- Tất cả biểu thức kiểu **Union** $T_1, \dots, T_k$ **End** được ánh xạ vào $\psi(T_1) \sqcup \dots \sqcup \psi(T_k)$.
- Tất cả biểu thức kiểu **Set-of** T được ánh xạ vào **SetType** $\sqcup \forall \text{member}.\psi(T)$.
- Tất cả thuộc tính A được ánh xạ vào một vai trò $\psi(A)$.
- Tất cả biểu thức kiểu **Record** $A_1:T_1, \dots, A_k:T_k$ **End** được ánh xạ vào:

$$\text{Rectype} \sqcup \forall \psi(A_1).\psi(T_1) \sqcup \exists=1 \psi(A_1) \sqcup \dots \sqcup \forall \psi(A_k).\psi(T_k) \sqcup \exists=1 \psi(A_k).$$

Sử dụng ψ ta định nghĩa cơ sở tri thức $\psi(\mathbb{S})$ tương ứng với $\mathbb{S}$ như sau:

$$\begin{aligned} \text{AbstractClass} & \quad \mathbf{v} \quad \exists=1 \text{value} \\ \text{RecType} & \quad \mathbf{v} \quad \forall \text{value.} ? \\ \text{SetType} & \quad \mathbf{v} \quad \forall \text{value.} ? \sqcup : \text{RecType} \end{aligned}$$

và mỗi định nghĩa lớp

$$\text{Class } C \text{ is-a } C_1, \dots, C_n \text{ type-is } T$$

trong $\mathbb{S}$ được thể hiện bằng một bao hàm sau:

$$\psi(C) \mathbf{v} \text{AbstractClass} \sqcup \psi(C_1) \sqcup \dots \sqcup \psi(C_n) \sqcup \forall \text{value}.\psi(T).$$

Ví dụ: Ta sẽ chuyển đổi lược đồ hướng đối tượng trong Hình 2.2 tương ứng với cơ sở tri thức ALCQI sau đây:

Professor	v	AbstractClass $\sqcup \forall \text{value}.$ (RecType $\sqcup$ $\forall \text{ProfID}.\text{String} \sqcup \exists=1 \text{ ProfID} \sqcup$ $\forall \text{ProfName}.\text{String} \sqcup \exists=1 \text{ ProfName} \sqcup$ $\forall \text{ProfAge}.\text{Integer} \sqcup \exists=1 \text{ ProfAge} \sqcup$
------------------	----------	--

		$\forall \text{ProfPhoneNum.String } u \exists=1 \text{ ProfPhoneNum})$
Student	v	$\text{AbstractClass } u \forall \text{value.}(\text{RecType } u$ $\forall \text{StudentID.String } u \exists=1 \text{ StudentID } u$ $\forall \text{StudentName.String } u \exists=1 \text{ StudentName } u$ $\forall \text{StudentAge.Integer } u \exists=1 \text{ StudentAge } u$ $\forall \text{StudentSex.Boolean } u \exists=1 \text{ StudentSex } u$ $\forall \text{StudentAddress.String } u \exists=1 \text{ StudentAddress})$
Teacher	v	$\text{AbstractClass } u \forall \text{value.}(\text{Professor } t$ $\text{GradStudent})$
GradStudent	v	$\text{AbstractClass } u \text{ Student } u \forall \text{value.}(\text{RecType } u$ $\forall \text{Degree.String } u \exists=1 \text{ Degree})$
Course	v	$\text{AbstractClass } u \forall \text{value.}(\text{RecType } u$ $\forall \text{Enrolls.}(\text{SetType } u \forall \text{member.Student}) u \exists=1$ $\text{Enrolls } u \forall \text{taughtBy.Teacher } u \exists=1 \text{ taughtBy})$
AbstractClass	v	$\exists=1 \text{value}$
RecType	v	$\forall \text{value.}?$
SetType	v	$\forall \text{value.} ? u : \text{RecType}$

Hình 3.2 Cơ sở tri thức ALCQI tương ứng với lược đồ hướng đối tượng trong Hình 2.2

3.4. CHUYỂN DỮ LIỆU TỪ CƠ SỞ DỮ LIỆU VÀO ABOX CỦA LOGIC MÔ TẢ

Như ta đã biết, các thực thể được xem là các khái niệm nguyên tố trong DL, các thuộc tính được biểu diễn bằng các vai trò. Như vậy mỗi một bảng dữ liệu có n thuộc tính sẽ được biểu diễn bằng n vai trò. Việc đưa các dữ liệu vào ABox được tiến hành như sau: các đại diện của mỗi bản ghi được đưa vào

khái niệm bảng đã được định nghĩa, chẳng hạn như Professor (P001), Professor(P002), ta có thể coi P001, P002 là duy nhất (cũng là "khóa" trong CSDL quan hệ). Các thuộc tính của mỗi cá thể đưa vào các vai trò thuộc tính tương ứng, ví dụ ProfName(P001, Tên 1); ProfName(P002, tên 2), ProfAge(P001, 1950)...

Ta có thể xây dựng thuật toán chuyển đổi dữ liệu từ các bảng của cơ sở dữ liệu quan hệ vào ABox của cơ sở tri thức logic mô tả thông qua thủ tục **DBtoKBConvert** sau đây:

Procedure DBtoKBConvert()

Begin

{đối với các bảng thực thể, giả sử rằng khóa chính chỉ chứa 1 trường}

For each (tbl là các bảng thực thể) **in** (DB.tabledef) **do**

While not (tbl.Recordset.EOF) **do**

{xác nhận khái niệm trùng với tên của bảng mang giá trị của trường làm khóa chính trong bảng}

AssertConcept (tbl.Name) = tbl.PrimaryKeyField.Value

For each fld **in** (tbl.Fields) **do** *{với mỗi trường của bảng:}*

{Xác nhận giá trị cho các vai trò trùng tên với các thuộc tính của bảng}

AssertRole((fld.Name)=(tbl.PrimaryKeyField, fld.Value)

Next *{với trường tiếp theo}*

tbl.Recordset.MoveNext *{Tới bản ghi tiếp theo của bảng}*

Loop

Next *{Với bảng thực thể tiếp theo}*

{Đối với các bảng quan hệ}

For each (tbl là các bảng quan hệ) **in** (DB.tabledef) **do**

While not (tbl.Recordset.EOF) **do**

```

        {Thêm một giá trị bằng giá trị số thứ tự bản ghi
        cho khái niệm có tên trung tâm của bảng quan hệ}
        AssertConcept(tbl.Name) = tbl.RecordNumber
        For each fld in tbl.Fields do
            {Thêm một giá trị cho vai trò có tên trùng tên
            trường trong bảng}
            AssertRole(fld.Name)= (tbl.RecordNum, fld.Value)
        Next
        tbl.Recordset.MoveNext
    loop
Next
End.

```

Hình 3.3: Thủ tục chuyển dữ liệu từ bảng vào ABox

Để cụ thể hơn ta xét ví dụ về việc chuyển các dữ liệu được đưa ra dưới dạng các bảng quan hệ dưới đây thành cơ sở tri thức ABox tương ứng với mô hình dữ liệu ở Hình 2.1.

Professor:

ProfID	ProfName	ProfAge	ProfPhoneNum
P001	Professor A	1950	9422957
P002	Professor B	1952	8343097
P003	Professor C	1966	5742488
P004	Professor D	1952	7662009
P005	Professor E	1963	8271483

Bảng 3.1: Bảng thực thể Professor

Student:

StudentID	StudentName	StudentAge	StudentSex	StudentAddress
S001	Student 1	1981	Nam	Hà Nội
S002	Student 2	1979	Nam	Nam Định
S003	Student 3	1973	Nữ	Hà Nội
S004	Student 4	1975	Nữ	Hà Nội
S005	Student 5	1979	Nam	Thái Bình
S006	Student 6	1979	Nữ	Bắc Giang

Bảng 3.2: Bảng thực thể Student

Course:

CourseID	CourseName	CourseStart	CourseLong
C01	Course a	15/8/2006	4
C02	Course b	27/8/2006	3
C03	Course c	8/9/2006	5
C04	Course d	20/9/2006	4
C05	Course e	1/10/2006	4
C06	Course f	22/10/2006	4

Bảng 3.3: Bảng thực thể Course

AdvCourse:

AdvCourseID
C04
C05

Bảng 3.4: Bảng thực thể AdvCourse

Teaching:

TaughtBy	TeachOf
P001	C02

P002	C04
P003	C01
P004	C05
P005	C03

Bảng 3.5: Bảng quan hệ Teaching

GradStudent:

GradStudent	Degree
S002	Degree A
S006	Degree A

Bảng 3.6: Bảng thực thể GradStudent

Enrolling:

EnrollIn	EnrollOf
S001	C01
S001	C02
S001	C03
S001	C06
S002	C01
S002	C04
S002	C05
S003	C01
S003	C02
S003	C06
S004	C02
S004	C06
S004	C03

Bảng 3.7: Bảng quan hệ Enrolling

Từ các bảng quan hệ với các dữ liệu trên ta xây dựng được ABox tương ứng với các khái niệm, vai trò trong Hình 3.1 như sau:

Professor (P001), ProfName(P001, Professor A), ProfAge(P001, 1950), ProfPhoneNum(P001, 9422957)
Professor (P002), ProfName(P001, Professor B), ProfAge(P001, 1952), ProfPhoneNum(P001, 8343097)
Professor (P003), ProfName(P001, Professor C), ProfAge(P001, 1966), ProfPhoneNum(P001, 5742488)
Professor (P004), ProfName(P001, Professor D), ProfAge(P001, 1962), ProfPhoneNum(P001, 7662009)
Professor (P005), ProfName(P001, Professor E), ProfAge(P001, 1963), ProfPhoneNum(P001, 8271483)
...
Student(S001), StudentName(S001, Student 1), StudentAge(S001, 1981), StudentSex(S001, Nam), StudentAddress(S001, Hà Nội)
....
Course(C01), CourseName(C01, Course a), CourseStart(C01, 15/8/2006), CourseLong(C01, 4)
....
AdvCourse(C04), AdvCourse(C05)
....
GradStudent(S02), Degree(S02, Degree A)
GradStudent(S06), Degree(S06, Degree A)
....
Teaching(1), TaughtBy(1, P001), TeachOf(1, C02)
Teaching(2), TaughtBy(2, P002), TeachOf(2, C04)
....

Enrolling (1), EnrollIn(1, S001), EnrollOf(1, C01)
Enrolling (2), EnrollIn(2, S001), EnrollOf(1, C02)
...
Enrolling (5), EnrollIn(5, S002), EnrollOf(5, C01)

Hình 3.4 ABox nhận được từ việc chuyển đổi dữ liệu của các thực thể

3.5 TỔNG KẾT CHƯƠNG

Chương 3 đã giới thiệu phương pháp để chuyển lược đồ của mô hình dữ liệu thực thể - quan hệ và mô hình dữ liệu hướng đối tượng thành TBox của cơ sở tri thức DL, đồng thời xây dựng một thuật toán để chuyển đổi dữ liệu từ các bảng trong cơ sở dữ liệu quan hệ vào ABox của cơ sở tri thức DL. Chương tiếp theo ta sẽ thảo luận cách thức thực hiện truy vấn trên cơ sở tri thức đã được chuyển đổi ở Chương này.

Chương 4. TRUY VẤN

4.1. NGUYÊN TỬ TRUY VẤN, ĐỐI TƯỢNG, CÁ THỂ VÀ BIẾN

Các biểu thức cơ bản trong truy vấn bằng DL được gọi là nguyên tử truy vấn (query atoms), hay nói một cách đơn giản là tiên đề. Các nguyên tử có thể là một ngôi hoặc hai ngôi. Nguyên tử một ngôi tham chiếu đến một đối tượng, còn nguyên tử hai ngôi tham chiếu đến hai đối tượng.

Một đối tượng là một cá thể trong ABox hoặc là một biến. Các biến được buộc vào các cá thể trong ABox mà thoả mãn biểu thức truy vấn.

Một truy vấn ở dạng cơ bản là truy vấn hội có dạng $Q(X) := q_1 \wedge \dots \wedge q_n$, với $q_1, \dots, q_n$ là các nguyên tử truy vấn. Mỗi nguyên tử truy vấn có dạng $C(x)$ hoặc $R(x, y)$, ở đó C là khái niệm còn R là vai trò còn x, y là các cá thể hoặc các biến. X là tập các biến riêng biệt trong truy vấn, nó được dùng để trả về các thể hiện mà người dùng quan tâm.

Một truy vấn được chia làm 2 phần, phần đầu và phần thân. Phần thân truy vấn là một biểu thức được thiết lập bởi các tiên đề. Phần đầu truy vấn để trả về danh sách các thể hiện được buộc vào biến.

4.1.1. Nguyên tử truy vấn khái niệm

Nguyên tử khái niệm là loại nguyên tử một ngôi. Một nguyên tử khái niệm được dùng để trả về một cá thể của một khái niệm. Ví dụ ta có truy vấn sử dụng nguyên tử khái niệm **Student(x)** để trả về các thể hiện (instances) của khái niệm Student như sau:

$$ans(x) := Student(x)$$

Giả sử với cơ sở tri thức ta đã xây dựng ở chương 3 gồm bộ TBox trong hình 3.1 và ABox hình 3.2, với truy vấn trên ta được câu trả lời là các sinh viên:

(S001 S002 S003 S004 S005 S006 S007)

Truy vấn với các cá thể: Giả sử ta chỉ muốn biết có sinh viên nào trong tất cả các mô hình ở ABox ở hình 3.2 không. Ta chỉ truy vấn một cách đơn giản là:

$$ans() := student(x)$$

Câu trả lời là đúng (True) hoặc rỗng (NIL). Trong trường hợp này câu trả lời là "True".

Giả sử ta chỉ muốn biết S001 có phải là một sinh viên không ta có thể thực hiện truy vấn:

$$ans() := Student(S001)$$

Và câu trả lời ta nhận được là "True".

4.1.2. Nguyên tử truy vấn vai trò

Kiểu nguyên tử thứ hai là nguyên tử vai trò. Đó là các nguyên tử hai ngôi, tương phản với nguyên tử khái niệm một ngôi. Nguyên tử vai trò được dùng để trả về một cặp filler vai trò từ ABox. Giả sử ta tìm tất cả các cặp sinh viên và giới tính trong cơ sở tri thức trên. Ta có thể tạo truy vấn sau:

$$ans(StudID, StudSex) := StudentSex(StudID, StudSex)$$

Câu trả lời là:

S001 Nam

S002 Nam

S003 Nữ

S004 Nữ

S005 Nam

S006 Nam

trong câu truy vấn trên biểu thức $StudentSex(StudID, StudSex)$ là nguyên tử vai trò.

Nếu ta chỉ muốn quan tâm đến những sinh viên là Nữ, ta có thể thực hiện:

$$ans(X) := StudentSex(X, Nữ)$$

Kết quả trả về là: S003, S004, S006

Ta cũng có thể truy vấn bằng nguyên tử vai trò ở dạng đảo để trả về giới tính của sinh viên 002: $ans(X) := StudentSex^-(002, X)$

Kết quả trả về là: *Nam*

4.2. TRUY VẤN PHỨC HỢP

Sau khi đã thảo luận các nguyên tử truy vấn ta tiếp tục với việc miêu tả phần thân truy vấn. Phần thân truy vấn được định nghĩa bằng các nguyên tử đơn hoặc kết hợp phức tạp các nguyên tử bằng các constructor $^, _, \cdot, \neg, \forall, \exists$.

Dựa vào cơ sở tri thức ở chương 3 ta xét các ví dụ sau:

Ví dụ 1: Cho biết những sinh viên có năm sinh là 1979 và là nam

$$ans(x) := Student(x) \wedge StudentAge(x, 1979) \wedge StudentSex(x, \text{"Nam"})$$

Ta có mô tả khái niệm (áp dụng thuật toán hỗ trợ mô tả ở mục 4.3) như sau:

$$Student \cup \exists \geq 1 StudentAge \cup \forall StudentAge. \{1979\} \cup \exists \geq 1 StudentSex \cup \forall StudentSex. \{\text{Nam}\}$$

Kết quả thu được là: S002, S005

Ví dụ 2: Cho biết những sinh viên nam sống ở Nam Định có cùng tuổi với một trong những sinh viên nữ

$$ans(x) := Student(x) \wedge StudentSex(x, \{\text{Nam}\}) \wedge StudentAddress(x, \{\text{Nam Định}\}) \wedge StudentAge(x, y) \wedge StudentAge(z, y) \wedge StudentSex(z, \{\text{Nữ}\})$$

Có thể chuyển phần thân của truy vấn trên thành mô tả khái niệm (áp dụng thuật toán hỗ trợ mô tả ở mục 4.3):

$$Student \cup \exists \geq 1 StudentSex \cup \forall StudentSex. \{\text{Nam}\} \cup \exists \geq 1 StudentAddress \cup \forall StudentAddress. \{\text{Nam Định}\} \cup \exists \geq 1 StudentAge \cup \forall StudentAge. (\exists \geq 1 StudentAge^- \cup \forall StudentAge^-. (\exists \geq 1 StudentSex \cup \forall StudentSex. \{\text{Nữ}\}))$$

Kết quả thu được là: S002

4.3. HỖ TRỢ MÔ TẢ - ĐỊNH NGHĨA VÀ THUẬT TOÁN

Cho $C_J(X, Y)$ là hội của các nguyên tử khái niệm và nguyên tử vai trò, với X là các biến tách biệt, còn Y là các biến tồn tại. Ta nói rằng biến z' là một kế thừa trực tiếp của biến z trong $C_J(X, Y)$ nếu ở đó tồn tại một nguyên tử $R(z, z')$ trong $C_J(X, Y)$. Quan hệ kế thừa này là bao đóng của quan hệ kế thừa trực tiếp. Với mọi $x \in X$, cho $s(x)$ là một tập các kế thừa của x mà tập kế thừa này là các biến tồn tại, cho $C_J(x)$ là mệnh đề hội các nguyên tử của $C_J(X, Y)$ và gồm có các nguyên tử khái niệm dính lứu đến các biến trong $s(x)$. Ta định nghĩa một đồ thị G giải thích sự liên kết của các biến xuất hiện trong $C_J(X, Y)$ như sau: Các nút của đồ thị là các biến, có một cung từ một biến bất kỳ đến một biến kế thừa trực tiếp của nó. Ta nói rằng một mệnh đề hội $C_J(x)$ nằm trong $C_J(X, Y)$ có cấu trúc cây khi và chỉ khi:

- Với mọi biến tách biệt x' mà $x \neq x'$ thì $s(x) \setminus s(x') = \emptyset$ hoặc $s(x) \subset s(x')$.
- Đồ thị con của G giới hạn các biến $x \in s(x)$ là một cây có gốc là x .
- Mỗi nút trong cây không có kế thừa tách biệt trong G .

Ví dụ $C(x_1) \wedge R_1(x_1, y_1) \wedge R_2(y_1, y_2) \wedge D(y_2)$ là một cấu trúc cây có gốc là x_1 .

Còn $C_1(x_1) \wedge R(x_1, y_1) \wedge R_2(y_1, x_1)$ hoặc $R(y, y)$ không phải là cấu trúc cây.

Biểu thức hội của các nguyên tử khái niệm và các nguyên tử vai trò có cấu trúc cây có thể thay thế bằng một nguyên tử khái niệm đơn giản theo thuật toán diễn đạt bằng thủ tục "**DescriptiveSupport**" [10] dưới đây:

Procedure *DescriptiveSupport* (ST(z))

{ ST(z) là một biểu thức của các nguyên tử khái niệm và vai trò có cấu trúc cây với gốc là z và các biến tồn tại là Y }

{ Thuật toán trả về một khái niệm DS và một nguyên tử khái niệm DS(z) }

If ST(z) có dạng: $C_1(z) \wedge \dots \wedge C_n(z)$, với C_i là các biểu thức khái niệm

Then: $DS = C_1 \cup \dots \cup C_n$

Else If ST(z) có dạng:

$$C_1(z) \wedge \dots \wedge C_n(z) \wedge R_1(z, y_1^1) \wedge \dots \wedge R_1(z, y_1^{n_1})$$

$$\wedge \dots \wedge R_k(z, y_k^1) \wedge \dots \wedge R_k(z, y_k^{n_k}) \wedge Tree(y_1^1) \wedge \dots \wedge tree(y_1^{n_1})$$

$$\wedge \dots \wedge Tree(y_k^1) \wedge \dots \wedge Tree(y_k^{n_k}), \text{ với các } Tree(y_i^j) \text{ là biểu thức hội của các}$$

nguyên tử có cấu trúc cây gốc là } y_i^j

Then:

for $i \in [1..k]$ **do:**

for $j \in [1..n_i]$ **do:**

$DS_j^i := DescriptiveSupport(Tree(y_i^j))$

$DS := C_1 \cup \dots \cup C_n \cup (\geq n_1 R_1) \cup \dots \cup (\geq n_k R_k) \cup \forall R_1. (DS_1^1 \cup \dots \cup DS_{n_1}^1) \cup \dots \cup$
 $\forall R_k. (D_k^1 \cup \dots \cup DS_{n_k}^{n_k})$

return $DS \{ \text{ trả lại một mô tả của } ST(z) \}$

return $DS(z) \{ \text{ trả lại một nguyên tử } DS \text{ của } ST(z) \}$

Hình 4.1: Thủ tục hỗ trợ mô tả

Chẳng hạn, xét biểu thức hội các nguyên tử khái niệm và vai trò dưới đây, với x_1 là một biến tách biệt còn y_1 và y_2 là các biến tồn tại:

$$C(x_1) \wedge R_1(x_1, y_1) \wedge R_2(y_1, y_2) \wedge D(y_2)$$

Biểu thức trên có cấu trúc cây gốc là x_1 . Ta có khái niệm trả về bằng cách áp dụng thuật toán DescriptiveSupport như sau:

$$C \cup (\geq 1 R_1) \cup \forall R_1. ((\geq 1 R_2) \cup \forall R_2. D)$$

hay ta có thể viết lại dưới dạng chuẩn như sau:

$$C \cup (\geq 1 R_1) \cup \forall R_1. (\geq 1 R_2) \cup \forall R_1. \forall R_2. D$$

Ta xét tiếp các truy vấn sau:

Ví dụ 3: Tìm những sinh viên học 3 môn học

$Ans(x) :- Student(x) \wedge \exists=3 EnrollIn(x, _)$

Mô tả logic như sau:

$Student \cup \exists=3 EnrollIn$

Kết quả là: S002, S003, S004

Ví dụ 4: Tìm những sinh viên học giáo sư có tên là “Professor C”

$$\text{Ans}(x) :- \text{Student}(x) \wedge \text{EnrollIn}(y,x) \wedge \text{EnrollOf}(y,z) \wedge \text{teachOf}(z,w) \wedge \\ \text{ProfessorName}(w, \{\text{Professor B}\})$$

Luật truy vấn biểu diễn bằng DL như sau:

$$\text{Student } u \exists \geq 1 \text{EnrollIn}^- u \forall \text{EnrollIn}^-. (\exists \geq 1 \text{EnrollOf } u \\ \forall \text{EnrollOf}. (\exists \geq 1 \text{teachOf } u \forall \text{teachOf}. (\exists \geq 1 \text{ProfessorName } u \\ \forall \text{ProfessorName}. \{\text{Professor C}\}))).$$

Kết quả là: S001, S002, S003

4.4. TỔNG KẾT CHƯƠNG

Ở Chương 4, ta đã thảo luận về truy vấn cơ sở tri thức, từ các thành phần cơ bản của truy vấn như truy vấn nguyên tử khái niệm, truy vấn nguyên tử vai trò đến các truy vấn phức hợp bằng biểu thức hội các thành phần khái niệm và vai trò cơ sở. Chương 4 cũng đưa ra thuật toán nhằm chuyển đổi các câu truy vấn xây dựng theo cách thể hiện của ngôn ngữ lập trình logic Datalog sang biểu diễn mô tả khái niệm trong logic mô tả. Cũng như đã xem xét các ví dụ cụ thể có áp dụng thuật toán hỗ trợ mô tả (*DescriptiveSupport*) đã đưa ra.

KẾT LUẬN

Trong đề tài này em đã giới thiệu một cách tương đối đầy đủ về họ logic mô tả ALC , thảo luận về khả năng biểu diễn cơ sở tri thức của ngôn ngữ AL ; đồng thời giới thiệu một ngôn ngữ lập trình tương thích để biểu diễn cơ sở tri thức bằng logic mô tả, đó là ngôn ngữ Datalog. Đề tài đã trình bày khả năng ứng dụng logic mô tả trong biểu diễn các mô hình cơ sở dữ liệu, như cơ sở dữ liệu quan hệ và cơ sở dữ liệu hướng đối tượng. Đề tài cũng đưa ra các phương pháp biến đổi một cơ sở dữ liệu thành cơ sở tri thức dựa trên logic mô tả, như biểu diễn dữ liệu của các bảng trong cơ sở dữ liệu quan hệ thành các sự kiện trong ABox của cơ sở tri thức. Đề tài cũng thảo luận về khả năng thực hiện truy vấn thông qua cách thức biểu diễn của ngôn ngữ Datalog, biến đổi các luật của Datalog thành các mô tả khái niệm trong logic mô tả.

Với khả năng biểu diễn, truy vấn bằng logic mô tả ta có thể suy rộng ra rằng, cơ sở dữ liệu có thể được biến đổi thành cơ sở tri thức, với ngữ nghĩa truy vấn phong phú hơn.

Tuy nhiên, ngoài những công việc đã làm được, trong đề tài này mới mang tính lý thuyết. Em vẫn chưa xây dựng được chương trình DEMO, nên chắc chắn chưa hoàn toàn thuyết phục. Vì vậy, kế hoạch công việc trong tương lai để hoàn thiện hơn sẽ là đi sâu, ứng dụng một trong những ngôn ngữ suy diễn (chẳng hạn Prolog, Datalog, Racer...) để xây dựng được một cơ sở tri thức cụ thể dựa trên một cơ sở dữ liệu cụ thể.

Trong đề tài chắc chắn còn nhiều những hạn chế, em rất mong các thầy, các cô và các bạn đồng nghiệp đóng góp ý kiến cho em để tiếp tục hoàn thiện đề tài.

CÁC THUẬT NGỮ

Tiếng việt	Tiếng Anh
Bài toán bao hàm	Subsumtion
Bài toán không giao	Disjointness
Bài toán không thoả	Unsatisfiability
Bài toán thoả	Satisfiability
Bài toán tương đương	Equivalence
Bộ khẳng định	Assertional box (ABox)
Bộ thuật ngữ	Terminological box (TBox)
Cơ sở dữ liệu mở rộng	Extensional Database (EDB)
Cơ sở dữ liệu tăng cường	Intensional Database (IDB)
Cơ sở tri thức	Knowledge base
Chuẩn hoá	Normalization
Chuyên biệt hoá khái niệm	Concept specialization
Giới hạn lớn nhất	At most of
Giới hạn nhỏ nhất	At least of
Giới hạn số lượng	Number restriction
Hàm diễn dịch	Interpretation function
Hạng thức	Term
Hội	Conjunction
Khái niệm	Concept
Khái niệm đáy	Bottom concept
Khái niệm đỉnh	Top concept
Khái niệm nguyên thuỷ	Primitive concept
Khái quát hoá	Generalization
Khẳng định	Assertion
Không chứa mâu thuẫn	Clash-free
Logic mô tả	Description logic

Lượng từ tồn tại	Existential quantification
Lượng từ với mọi	Universal quantification
Mâu thuẫn	Clash
Mô hình	Model
Mở rộng	Expansion
Ngôn ngữ biểu diễn tri thức thuật ngữ	Terminological knowledge representation language
Ngôn ngữ khái niệm	Concept language
Ngôn ngữ thuộc tính	Attributed language
Nguyên tử, nguyên tố	Atom
Ngữ nghĩa	Semantics
Phép diễn dịch	Interpretation
Phủ định	Negation
Quan hệ	Relationship
Thể hiện	Instance
Thuật ngữ	Terminology
Thuật toán cấu trúc	Structural algorithm
Tiên đề khái niệm	Terminological axiom
Tuyển	Disjunction
Vai trò	Role

TÀI LIỆU THAM KHẢO

Tiếng việt

- [1] Nguyễn Văn Ba, *Phân tích và thiết kế các hệ thống thông tin*, Nhà xuất bản Đại học Quốc gia.

Tiếng Anh

- [2] Alex Borgida, (1995), *Description logics in data management*, IEEE Trans. on Know. and Data Engineering.
- [3] Alex Borgida and Ronald J. Brachman, *Conceptual modeling with description logics*, In Baader et al.
- [4] Catriel Beeri, Alon Y. Levy and Marie-Christine Rousset (1997), *Rewriting queries using views in description logics*, In Proceedings of the Sixteenth Symposium on Principles of Database Systems, (PODS'97).
- [5] Diego Calvanese, Maurizio Lenzerini, Daniele Nardi, *Description Logics For Conceptual Data Modeling*, Dipartimento di Informatica e Sistemistica, Università di Roma, "La Sapienza", Via Salaria 138. 10098, Roma, Italy.
- [6] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, *Data Complexity of Query Answering in Description Logics*, Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy.

- [7] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini (2000), *Answering queries using views over description logics knowledge bases*, In Proc. of AAAI.
- [8] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider (2003), *The Description Logic Handbook. Theory, Implementation and Applications*, Hardback.
- [9] Francesca A. Lisi and Floriana Esposito, *On Learning in AL-log*, Dipartimento di Informatica, Università degli Studi di Bari, Italy.
- [10] Marie Christine Rousset, *Backward Reasoning in Aboxes for Query Answering*, L.R.I Computer Science Laboratory C.N.R.S, University of Paris, Sud Building Orsay Cedex, France.
- [11] Mathieu Roger, Ana Simonet, Michel Simonet, *Bringing together Description Logics and Databases in an Object Oriented Model*, Laboratory TIMC-IMAG- Osiris Team, La Tronche Cedex.
- [12] Mathieu Roger, Ana Simonet, Michel Simonet, *A Description Logics-like Model for a Knowledge and Data Management System*, TIMC-IMAG Faculté de Médecine de Grenoble, 38706 La Tronche Cedex. France.