

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM



TIẾT MINH CHÍ

**BÀI TOÁN THÀNH VIÊN VÀ BÀI TOÁN
TÌM KHÓA TRONG LƯỢC ĐỒ QUAN HỆ
VỚI PHỤ THUỘC BOOLE**

LUẬN VĂN THẠC SĨ

Chuyên ngành: CÔNG NGHỆ THÔNG TIN

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, THÁNG 10 NĂM 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM



TIẾT MINH CHÍ

**BÀI TOÁN THÀNH VIÊN VÀ BÀI TOÁN TÌM
KHÓA TRONG LƯỢC ĐỒ QUAN HỆ
VỚI PHỤ THUỘC BOOLE**

LUẬN VĂN THẠC SĨ

Chuyên ngành: CÔNG NGHỆ THÔNG TIN

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: PGS TSKH NGUYỄN XUÂN HUY

TP. HỒ CHÍ MINH, THÁNG 10 NĂM 2015

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM
ngày ... tháng ... năm ...

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ Luận văn Thạc sĩ)

TT	Họ và tên	Chức danh Hội đồng
1		Chủ tịch
2		Phản biện 1
3		Phản biện 2
4		Ủy viên
5		Ủy viên, Thư ký

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được
sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

PHÒNG QLKH – ĐTSĐH

Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày..... tháng..... năm 20.....

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên:Giới tính:

Ngày, tháng, năm sinh:Nơi sinh:

Chuyên ngành:MSHV:

I- Tên đề tài:

.....
.....
.....

II- Nhiệm vụ và nội dung:

.....
.....
.....
.....

III- Ngày giao nhiệm vụ: (Ngày bắt đầu thực hiện LV ghi trong QĐ giao đề tài)

IV- Ngày hoàn thành nhiệm vụ:

V- Cán bộ hướng dẫn: (Ghi rõ học hàm, học vị, họ, tên)

.....
.....

CÁN BỘ HƯỚNG DẪN

KHOA QUẢN LÝ CHUYÊN NGÀNH

(Họ tên và chữ ký)

(Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

(Ký và ghi rõ họ tên)

TIẾT MINH CHÍ

LỜI CẢM ƠN

Trước tiên, tôi xin được gửi lời cảm ơn đến tất cả quý thầy cô, gia đình, bạn bè và đồng nghiệp. Bởi vì, trong suốt quá trình học tập và thực hiện luận văn này tôi đã nhận được sự hướng dẫn, giúp đỡ quý báu của quý thầy cô, gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc tôi xin được bày tỏ lời cảm ơn chân thành tới:

*Thầy hướng dẫn **Phó giáo sư - Tiến sĩ Khoa học Nguyễn Xuân Huy**, thầy đã truyền đạt những kiến thức rất bổ ích cho tôi trong quá trình học tập. Đồng thời, thầy đã dành nhiều thời gian để hướng dẫn cho tôi hết sức tận tình trong thời gian tôi thực hiện luận văn.*

Khoa Công Nghệ Thông Tin, Phòng quản lý khoa học - Đào tạo sau đại học trường Đại Học Công Nghệ Thành Phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi cho tôi hoàn thành chương trình học tập đúng thời gian quy định và giúp đỡ tôi có được định hướng đúng đắn trong quá trình tôi thực hiện luận văn này.

Toàn thể quý thầy cô đã nhiệt tình giảng dạy và truyền đạt những kiến thức bổ ích cho tôi trong suốt khóa học vừa qua.

Cuối cùng xin cảm ơn đến tất cả những người thân trong gia đình, bạn bè và đồng nghiệp đã giúp đỡ và động viên tôi trong suốt quá trình học tập và thực hiện luận văn này.

TIẾT MINH CHÍ

TÓM TẮT

Nội dung chính của luận văn là trình bày quá trình phát triển của phụ thuộc hàm. Từ đó chỉ ra sự cần thiết phải mở rộng phụ thuộc hàm sang phụ thuộc Boole dương trong cơ sở dữ liệu quan hệ. Các lớp phụ thuộc Logic nói chung và phụ thuộc Boole dương nói riêng là khái niệm dùng để mô tả chi tiết ngữ nghĩa của dữ liệu trong cơ sở dữ liệu. Phần đầu của luận văn mô tả và thiết lập sự tương quan giữa phụ thuộc hàm và phụ thuộc Boole dương. Phần thứ hai phát biểu một số bài toán liên quan đến lớp phụ thuộc Boole dương. Phần cuối vận dụng kết quả của bài toán thành viên để xây dựng một vài ứng dụng như tính bao đóng của tập thuộc tính và tìm khoá của lược đồ quan hệ với phụ Boole dương.

ABSTRACT

The main content of the thesis is to present the development of the dependent function. Subsequently, the thesis will point out the need to expand the dependent function into positive dependent Boole in a relevant database. The general Logic layer and positive dependent Logic Boole in particular, are concepts detailing the semantics of the data in the database. The first part of the thesis describes and establishes a correlation between dependent function and positive dependent Boole. The second part shows some exercises related to positive dependent Boole. The final part applies exercises' results to build applications such as the closure of attributes set and finding key attributes of relational schema with positive dependent Boole.

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT	VI
DANH MỤC CÁC BẢNG	VII
MỞ ĐẦU	1
CHƯƠNG 1 TỔNG QUAN VỀ LÝ THUYẾT CƠ SỞ DỮ LIỆU QUAN HỆ.....	6
1.2.1 Khái niệm phụ thuộc hàm	8
1.2.2 Một số tính chất của phụ thuộc hàm	9
1.2.3 Quan hệ thỏa phụ thuộc hàm.....	10
1.2.5 Phụ thuộc hàm và Logic mệnh đề	12
1.4 KẾT CHƯƠNG	16
CHƯƠNG 2.....	17
MỘT SỐ KHÁI NIỆM CƠ SỞ TRONG	17
PHỤ THUỘC BOOLE DƯƠNG	17
2.1 Biểu thức Boole.....	17
2.2 Phụ thuộc Boole dương.....	19
2.2.1 Công thức Boole dương	20
2.2.3 Một số tính chất của công thức Boole dương.....	20
2.2.4 Định lý tương đương	22
2.2.5 Quy trình áp dụng phép hợp giải.....	24
2.2.6 Bài toán thành viên.....	25
2.2.7 Ưu điểm của Thuật toán Member so với các thuật toán khác	28
2.2.9 Xác định công thức Boole dương từ một biểu thức Boole cho trước	31
2.10 KẾT CHƯƠNG.....	33
3.4 Xây dựng chương trình	43
3.4.1 Cài đặt chương trình.....	43
3.4.2 Kết quả thực nghiệm	43
3.5 Kết chương.....	43
KẾT LUẬN VÀ KIẾN NGHỊ HƯỚNG PHÁT TRIỂN	44
Kết luận	44
Kiến nghị hướng phát triển tiếp theo	44
TÀI LIỆU THAM KHẢO	45

DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT

CSDL	Cơ sở dữ liệu
CTBD	Công thức Boole dương
CTSD	Công thức suy dẫn
$P(U)$	Tập toàn bộ các công thức dương trên U
PTBD	Phụ thuộc Boole dương
PTBDTQ	Phụ thuộc Boole dương tổng quát
PTH	Phụ thuộc hàm
$R(U)$	Quan hệ R với tập thuộc tính U
$SAT(F)$	Tập toàn thể các quan hệ trên U thỏa tập ràng buộc F
$FD(R)$	Chứa toàn thể các phụ thuộc hàm đúng trong quan hệ R
\Leftrightarrow	Khi và chỉ khi
\nrightarrow	Không suy ra
\nVdash	Không suy dẫn ra
\rightarrow	suy ra, kéo theo
\subseteq	Chứa trong
\vdash	suy dẫn theo quan hệ
\models	suy dẫn logic
\vdash_2	suy dẫn theo quan hệ có không quá 2 bộ

DANH MỤC CÁC BẢNG

Bảng 1.1 Mô tả quan hệ bán hàng	6
Bảng 1.2 Minh họa một phần của quan hệ bán hàng.....	7
Bảng 1.3 Minh họa lược đồ quan hệ.....	9
Bảng 1.4 Minh họa quan hệ thỏa phụ thuộc hàm	9
Bảng 1.5 Minh họa PTH được phát biểu theo Logic mệnh đề	11
Bảng 1.6 Minh họa bảng trị của quan hệ bán hàng	13
Bảng 2.1 Mô tả mối liên hệ giữa PTH và Biểu thức Boole.....	16
Bảng 2.2 Minh họa bảng trị và bảng chân lý của biểu thức Boole.....	17
Bảng 2.3 Minh họa bảng trị, bảng chân lý của công thức Boole.....	21
Bảng 2.4 Một số luật logic thường trong phép biến đổi tương đương	23
Bảng 3.1 Minh họa các bước phủ không dư của thí dụ 3.1.1	34
Bảng 3.1 Minh họa các bước phủ không dư của thí dụ 3.1.2	34
Bảng 3.3 Minh họa các bước phủ không dư của thí dụ 3.1.3	35

DANH MỤC CÁC HÌNH

Hình 3.1 Gọi thủ tục “tv.” để giải bài toán thành viên	47
Hình 3.2 Gọi thủ tục “tk1.” để tìm khóa của LĐQH	48
Hình 3.3 Gọi thủ tục kiểm tra biểu thức logic	49
Hình 3.4 Gọi thủ tục tìm phủ không dư của tập PTBD	50
Hình 3.5 Gọi thủ tục tính bao đóng của tập thuộc tính.....	51

MỞ ĐẦU

Cơ sở dữ liệu là một yếu tố quan trọng trong việc phân tích, thiết kế các hệ thống nói chung. Cùng với sự phát triển không ngừng của Internet, việc trao đổi thông tin và truyền dữ liệu trên mạng là một nhu cầu tất yếu đặt ra. Ngày nay, việc phát triển các ứng dụng với cơ sở dữ liệu phân tán đang trở nên phổ biến. Đặc điểm của các ứng dụng này là: lượng thông tin trao đổi qua mạng lớn, dữ liệu lưu trữ phân tán ở nhiều nơi, các yêu cầu truy xuất thông tin có thể xảy ra ở nhiều điểm khác nhau, việc đảm bảo tính nhất quán, tránh dư thừa dữ liệu khi thêm, xóa thông tin cũng như các yêu cầu liên quan đến tổ chức, xử lý, phục hồi dữ liệu, ... luôn là vấn đề được quan tâm.

Để lưu trữ, quản lý và khai thác dữ liệu người ta có thể dùng nhiều mô hình tổ chức dữ liệu khác nhau từ những mô hình truyền thống như mô hình mạng, mô hình phân cấp, mô hình quan hệ,... đến các mô hình hiện đại đang được dùng phổ biến hiện nay đó là: mô hình cơ sở dữ liệu phân tán, mô hình cơ sở dữ liệu hướng đối tượng,... Trong các mô hình dữ liệu, việc nghiên cứu lý thuyết và ứng dụng của các ràng buộc dữ liệu hay còn gọi là các phụ thuộc dữ liệu là một yêu cầu cấp thiết đặt ra.

Lớp các phụ thuộc hàm và hầu hết các phụ thuộc bậc cao phát triển sau đó như phụ thuộc đối ngẫu, phụ thuộc mạnh, phụ thuộc yếu, phụ thuộc sai khác ... đều dựa trên quan hệ đẳng thức khi so sánh các trị của các thuộc tính xuất hiện trong các bộ. Trong thực tế, ngoài so sánh theo đẳng thức còn tồn tại những loại hình so sánh khác. Ta xét một số thí dụ sau:

1. Trong CSDL quản lý khám chữa bệnh, bệnh nhân muốn biết lịch khám bệnh của mình thì phải cung cấp được số thứ tự lượt khám bệnh và số biên lai viện phí. Nếu số thứ tự lượt khám nhỏ hơn hoặc bằng 100 thì lịch khám bệnh sẽ tiến hành trong ngày. Ngược lại lịch khám bệnh của bệnh nhân sẽ được tiến hành vào

các ngày kế tiếp. Như vậy, ta thấy phụ thuộc $[sott(\leq 100) \wedge sobl(m)] \rightarrow [thoi\ gian(\leq 24)]$ có ngữ nghĩa rộng hơn PTH $sott \rightarrow thoi\ gian$.

2. Trong CSDL quản lý sinh viên, mã số sinh viên (MSSV) gồm 04 ký tự, trong đó 02 ký tự đầu sẽ cho biết sinh viên học chuyên ngành nào. Hàm $Left(MSSV)$ dùng để cắt hai ký tự đầu MSSV và cho biết sinh viên đó học ở chuyên ngành nào. Thí dụ $Left(AV015,2)=AV$ cho ta biết sinh viên này học chuyên ngành Anh ngữ. Như vậy, phụ thuộc $Left(MSSV) \rightarrow Mã\ ngành$ có ngữ nghĩa rộng hơn PTH $MSSV \rightarrow Mã\ ngành$.

MSSV	Mã ngành	Tên ngành
AV015	AV	Anh ngữ
AV033	AV	Anh ngữ
TH076	TH	Tin học

3. Trong CSDL quản lý biển số xe mô tô, biển số xe (BSX) gồm 08 ký tự, trong đó 02 ký tự đầu cho biết mã tỉnh thành. Hàm $Left(BSX)$ dùng để cắt 02 ký tự đầu biển số xe và cho biết biển số xe được đăng ký ở tỉnh nào. Thí dụ $Left(52V58850,2)=52$ cho biết biển số xe này được đăng ký ở thành phố Hồ Chí Minh. Ta thấy phụ thuộc $Left(BSX) \rightarrow Mã\ tỉnh$ có nghĩa rộng hơn PTH $BSX \rightarrow Mã\ tỉnh$.

BSX	Mã tỉnh	Tỉnh thành
52V58850	52	Hồ Chí Minh
52X31033	52	Hồ Chí Minh
95H34632	95	Hậu Giang

Trong thực tế các cơ sở dữ liệu có quy mô lớn và biến động được phân tán ở nhiều nơi, trên địa bàn rộng, phục vụ nhiều người dùng với nhiều ứng dụng khác nhau thì các yêu cầu như đồng bộ dữ liệu (theo khóa và theo trọng số thông tin), xử lý các yêu cầu tra cứu thông tin với các điều kiện khác nhau một cách nhanh chóng, đảm bảo dữ liệu không bị mất mát trên đường truyền, ... Vấn đề tổ chức, thiết kế, quản lý dữ liệu sao cho việc lưu trữ tốn ít bộ nhớ nhất, khai thác hiệu quả và thời gian truyền dữ liệu được giảm tối đa ... cũng luôn là yêu cầu cần thiết đặt ra. Một trong các giải pháp để thực hiện việc này là mở rộng khái niệm đối sánh các trị của các thuộc tính xuất hiện trong các bộ, tìm tập các phụ thuộc dữ liệu thu gọn tương đương với tập phụ thuộc dữ liệu ban đầu.

Đây cũng chính là mục đích và ý nghĩa của việc mở rộng khái niệm so sánh trong lớp các phụ thuộc dữ liệu như phụ thuộc Boole dương hay các phụ thuộc có bản chất là phụ thuộc Boole dương như phụ thuộc sai khác, phụ thuộc đối sánh được nghiên cứu sau này.

Mục tiêu của luận văn là tìm hiểu về lý thuyết phụ thuộc dữ liệu và ứng dụng lý thuyết này trong một số ứng dụng. Đây là vấn đề nghiên cứu đã và đang được nhiều nhà khoa học quan tâm.

Mục đích nghiên cứu

- Nghiên cứu và tìm hiểu các lớp phụ thuộc Boole dương trong đó tập trung chủ yếu việc vào tìm hiểu một số khái niệm, tính chất của lớp phụ thuộc Boole dương và một số khía cạnh ứng dụng của lớp phụ thuộc này.

Phương pháp nghiên cứu

- Tìm hiểu các tài liệu và kết quả nghiên cứu có liên quan đến đề tài. Vận dụng chủ yếu các phương pháp và cấu trúc của toán học rời rạc kết hợp với việc phát triển lớp các phụ thuộc logic nhằm đảm bảo ngữ nghĩa của dữ liệu trong cơ sở dữ liệu.

Tổng quan về lĩnh vực nghiên cứu

Một số công trình nghiên cứu khoa học về các dạng phụ thuộc dữ liệu bậc cao trong cơ sở dữ liệu được công bố trong nước gần đây như:

- *Positive boolean dependency in the database model of block form của các tác giả Trần Minh Tuyến (Bộ môn Tin học, đại học Công đoàn) , Nguyễn Xuân Huy (Viện Công nghệ thông tin, Viện Hàn lâm khoa học và Công nghệ Việt Nam) , Trịnh Đình Thắng (Trung tâm Tin học, đại học Sư phạm Hà Nội) công bố trong Hội nghị khoa học quốc gia lần thứ VII: Nghiên cứu cơ bản và ứng dụng công nghệ thông tin tại Thái Nguyên từ ngày 19-20/6/2014.*

- *Tương đương giữa phụ thuộc Boole dương và phụ thuộc yếu trong cơ sở dữ liệu quan hệ của các tác giả Nguyễn Xuân Huy (Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam), Trương Thị Thu Hà (Trường Đại học Kinh doanh và Công nghệ Hà Nội) công bố trong Hội thảo quốc gia lần thứ XVII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông tại Đắc Lắc từ ngày 30-31/10/2014.*

Bố cục của luận văn

Về cấu trúc, luận văn được trình bày trong 3 chương, phần mở đầu, phần kết luận, phần mục lục, tài liệu tham khảo. Các nội dung của luận văn trình bày theo cấu trúc như sau:

Chương 1. Tổng quan về lý thuyết cơ sở dữ liệu quan hệ

Trình bày một số khái niệm chung về mô hình quan hệ và lớp phụ thuộc đầu tiên của phụ thuộc logic là phụ thuộc hàm.

Chương 2. Một số khái niệm cơ sở trong phụ thuộc Bool dương

Trình bày một số khái niệm về phụ thuộc Bool dương và các vấn đề liên quan đến bài toán thành viên đối với lớp phụ thuộc Boole dương. Đồng thời, trong chương này cũng đề cập đến quy trình chuẩn hóa một công thức về dạng chuẩn hội (CNF), phương pháp chứng minh một công thức bằng phép hợp giải và thuật toán xây dựng tập PTBD thỏa mãn quan hệ R cho trước.

Chương 3. Một số ứng dụng của bài toán thành viên

Đề cập đến một số ứng dụng của bài toán thành viên như tìm phủ không dư của một tập PTBD, tính bao đóng của một tập PTBD cho trước và tìm khóa của lược đồ quan hệ với phụ thuộc Boole dương. Ngoài ra, trong chương này cũng trình bày một số thí dụ minh họa cho các thuật toán.

Những kết quả trong luận văn là bước đầu tìm hiểu về lý thuyết các phụ thuộc Boole dương trong cơ sở dữ liệu. Đồng thời, kết hợp với một số thuật toán có thể xây dựng các ứng dụng phục vụ trong một số lĩnh vực chẳng hạn như cơ sở dữ liệu. Do điều kiện về thời gian và trình độ còn hạn chế, những vấn đề trình bày trong luận văn không tránh khỏi những thiếu sót. Tác giả rất mong được sự thông cảm và góp ý của các nhà khoa học, đồng nghiệp và bạn bè để luận văn được hoàn thiện tốt hơn.

Chương 1 TỔNG QUAN VỀ LÝ THUYẾT CƠ SỞ DỮ LIỆU QUAN HỆ

Lý thuyết về các phụ thuộc dữ liệu đóng vai trò quan trọng trong việc mô tả thế giới thực, nó phản ánh ngữ nghĩa của cơ sở dữ liệu. Trong quản lý cơ sở dữ liệu (CSDL), phụ thuộc dữ liệu được hiểu là những mệnh đề dùng để mô tả các ràng buộc mà dữ liệu phải đáp ứng trong thực tế. Nhờ có những mô tả phụ thuộc này mà hệ quản trị cơ sở dữ liệu có thể quản lý dữ liệu một cách chặt chẽ. Phụ thuộc dữ liệu được Codd[13], tác giả của mô hình dữ liệu quan hệ đặt nền móng từ những năm 70 với khái niệm phụ thuộc hàm. Sau đó các nhà nghiên cứu về phụ thuộc dữ liệu tiếp tục đề xuất các dạng phụ thuộc bậc cao. Đồng thời, họ cũng xây dựng các hệ tiên đề cho các lớp phụ thuộc này. Tức là họ đã đặt cơ sở lý thuyết về phụ thuộc dữ liệu. Thực tế cho thấy là ngay từ những ngày đầu phát triển lý thuyết cơ sở dữ liệu thì Logic đã được chọn như một ngôn ngữ hữu hiệu để đặc tả phụ thuộc dữ liệu. Do đó, trong số các loại hình phụ thuộc dữ liệu được đề xuất và phát triển sau này, các lớp phụ thuộc Logic luôn được các nhóm nghiên cứu đặc biệt quan tâm.

Trong chương này sẽ trình bày tổng quan quá trình phát triển của lớp phụ thuộc Logic đầu tiên đó là phụ thuộc hàm. Các nội dung chính được trình bày đó là: khái niệm cơ bản của lý thuyết cơ sở dữ liệu quan hệ, khái niệm về phụ thuộc hàm, các tính chất của phụ thuộc hàm, tiên đề , Logic mệnh đề và lược đồ quan hệ.

1.1 Một số khái niệm

Các khái niệm cơ bản về cơ sở dữ liệu quan hệ được trình bày lần đầu tiên trong công trình của Codd [13]. Các khái niệm liên quan đến các cơ sở dữ liệu và tri thức đã được trình bày đầy đủ trong tài liệu [2].

Các khái niệm về quan hệ, thuộc tính, bộ được định nghĩa như sau:

Định nghĩa 1.1

Mỗi quan hệ $R(U)$ có hình ảnh là một bảng, mỗi *cột* ứng với một thuộc tính, mỗi *dòng* là một bộ.

Ta ký hiệu $t(U)$ là một bộ trên tập thuộc tính U .

Một *quan hệ rỗng*, ký hiệu \emptyset , là quan hệ không chứa bộ nào.

Vì mỗi quan hệ là một tập các bộ nên trong quan hệ *không có hai bộ trùng lặp*.

Thí dụ 1.1 Cho quan hệ bán hàng

Bảng 1.1 Mô tả quan hệ bán hàng

Bộ	Mã hàng	Tên hàng	Đơn giá	Số lượng	Thành tiền
t1	01	bút	10	02	20
t2	02	vở	02	05	10
t3	01	bút	10	03	30
t4	03	kẹp giấy	10	02	20
t5	02	vở	02	04	8

Quan hệ trên có 05 thuộc tính, cụ thể là

- Mã hàng
- Tên hàng
- Đơn giá
- Số lượng
- Thành tiền

Đồng thời, quan hệ trên hiện có 05 bộ, cụ thể là t1, t2, t3, t4, t5.

1.2 Phụ thuộc hàm

Phụ thuộc hàm là lớp phụ thuộc đầu tiên của phụ thuộc Logic, đây cũng là lớp phụ thuộc kinh điển được Codd [13] đề xuất sớm nhất và đóng vai trò quan trọng trong việc thiết kế CSDL. Phần này sẽ trình bày khái niệm về phụ thuộc hàm, một

số tính chất của phụ thuộc hàm, khái niệm về lược đồ quan hệ cũng được đề cập trong chương này.

1.2.1 Khái niệm phụ thuộc hàm

Định nghĩa 1.2.1

Cho tập thuộc tính U . Một *phụ thuộc hàm* (PTH) trên U là biểu thức dạng

$$f: X \rightarrow Y; X, Y \subseteq U$$

Nếu $f: X \rightarrow Y$ là một phụ thuộc hàm trên U thì ta nói tập thuộc tính Y *phụ thuộc* vào tập thuộc tính X , hoặc tập thuộc tính X *xác định hàm* tập thuộc tính Y . X là *vế trái* và Y là *vế phải* của PTH f .

Cho quan hệ $R(U)$ và một PTH $f: X \rightarrow Y$ trên U . Ta nói quan hệ R *thỏa PTH* f và viết $R(f)$, nếu hai bộ tùy ý trong R giống nhau trên X thì chúng cũng giống nhau trên Y .

$$R(X \rightarrow Y) \Leftrightarrow (\forall u, v \in R): (u.X = v.X) \rightarrow (u.Y = v.Y)$$

Ta dùng ký hiệu $X \nrightarrow Y$ với ý nghĩa tập thuộc tính Y *không phụ thuộc hàm* vào tập thuộc tính X .

Thí dụ 1.2.1: Cho quan hệ bán hàng

Bảng 1.2 Minh họa một phần của quan hệ bán hàng

Mã hàng	Tên hàng	Đơn giá	Số lượng	Thành tiền
01	bút	10	02	20
02	vở	02	05	10
01	bút	10	03	30

Ta có thể liệt kê các phụ thuộc hàm

Mã hàng \rightarrow tên hàng

Mã hàng \rightarrow đơn giá

Mã hàng \rightarrow tên hàng, đơn giá

Số lượng, đơn giá \rightarrow Thành tiền

Số lượng, thành tiền \rightarrow đơn giá

Ngoài ra trong quan hệ bán hàng bên trên còn có các phụ thuộc hàm tầm thường đó là

Mã hàng \rightarrow Mã hàng

Số lượng, đơn giá \rightarrow đơn giá

Đơn giá \rightarrow đơn giá, ...

1.2.2 Một số tính chất của phụ thuộc hàm

Cho tập thuộc tính U và các tập phụ thuộc hàm F, G trên U , tập các quan hệ \Re trên U , các quan hệ R và S trên U . Khi đó:

1. Nếu $F \subseteq G$ thì $SAT(F) \supseteq SAT(G)$

2. $SAT(FG) = SAT(F) \cap SAT(G)$

3. $FD(R \cup S) \subseteq FD(R) \cap FD(S)$

4. $R \subseteq S \rightarrow FD(R) \supseteq FD(S)$

5. $F \subseteq FD(SAT(F))$

6. $\Re \subseteq SAT(FD(\Re))$

7. $SAT(FD(SAT(F))) = SAT(F)$

8. $FD(SAT(FD(\Re))) = FD(\Re)$

- Trong đó:

+ Toán tử $SAT(F)$ cho ta tập toàn thể các quan hệ trên U thỏa tập PTH F .

+ Toán tử $FD(R)$ chứa toàn thể các phụ thuộc hàm đúng trong quan hệ R .

1.2.3 Quan hệ thỏa phụ thuộc hàm

Định nghĩa 1.2.3

Cho tập PTH F trên tập thuộc tính U . Ta nói quan hệ $R(U)$ *thỏa tập PTH F* , và viết $R(F)$, nếu R thỏa mọi PTH trong F ,

$$R(F) \Leftrightarrow (\forall f \in F): R(f)$$

Nếu quan hệ R thỏa PTH f ta cũng nói PTH f *đúng trong quan hệ R* .

Thí dụ 1

Cho lược đồ quan hệ: $Q(A, B, C, D, E)$ và tập PTH $F = \{ B \rightarrow D, DE \rightarrow A, AB \rightarrow C \}$

Bảng 1.3 Minh họa lược đồ quan hệ

$R(ABCDE)$

A	B	C	D	E
1	2	3	4	5
1	4	3	4	5
1	2	4	4	1

Nhận xét

	f1	f2	f3
	$B \rightarrow D$	$DE \rightarrow A$	$AB \rightarrow C$
R	thỏa	thỏa	ko thỏa
Lý do			cặp bộ 1 và 3

Thí dụ 2 Ta có quan hệ sau: Dạy(Giáo viên, môn học, tài liệu)

Bảng 1.4 Minh họa quan hệ thỏa phụ thuộc hàm

Dạy	Giáo viên	Môn học	Tài liệu
	AA	Môn học 1	XX
	AA	Môn học 2	YY
	BB	Môn học 3	ZZ
	CC	Môn học 4	TT

Nhận xét

	f1	f2
	Môn học \rightarrow Tài liệu	Giáo viên \rightarrow Môn học
R	thỏa	không thỏa
Lý do		cặp bộ 1 và 2

1.2.4 Hệ tiên đề Armstrong**Định nghĩa 1.2.4**

Cho tập PTH F trên tập thuộc tính U . Bao đóng của F , ký hiệu F^+ là tập nhỏ nhất các PTH trên U chứa F và thỏa các tính chất $F1-F3$ của hệ tiên đề Armstrong A° sau đây:

$\forall X, Y, Z \subseteq U$:

F1. Tính phản xạ: Nếu $X \supseteq Y$ thì $X \rightarrow Y \in F^+$

Thí dụ

Ta có $ABC \supseteq BC$ thì $ABC \rightarrow BC$

F2. Tính gia tăng: Nếu $X \rightarrow Y \in F^+$ thì $XZ \rightarrow YZ \in F^+$

Thí dụ

Ta có $C \rightarrow D$ thì $ABC \rightarrow ABD$

F3. Tính bắc cầu: Nếu $X \rightarrow Y \in F^+$ và $Y \rightarrow Z \in F^+$ thì $X \rightarrow Z \in F^+$

Thí dụ

Ta có $AB \rightarrow CD$ và $CD \rightarrow EF$ thì $AB \rightarrow EF$

Nhận xét:

Các PTH có vế trái chứa vế phải như mô tả trong F1 được gọi là *tầm thường*. Các PTH tầm thường đúng trong mọi quan hệ. Ngoài ra, các quan hệ trên tập thuộc tính U có không quá một bộ thỏa mọi PTH trên U .

1.2.5 Phụ thuộc hàm và Logic mệnh đề

Xét quan hệ bán hàng

Bảng 1.5 Minh họa PTH được phát biểu theo Logic mệnh đề

Mã hàng	Tên hàng	Đơn giá	Số lượng	Thành tiền
01	bút	10	02	20
02	vở	02	05	10
01	bút	10	03	30
03	kẹp giấy	10	02	20
02	vở	02	04	8

Với quan hệ trên ta có các phụ thuộc hàm

- Mã hàng \rightarrow tên hàng
- Đơn giá, số lượng \rightarrow thành tiền

Nhận xét: Các phụ thuộc hàm trên có thể phát biểu lại dưới dạng Logic mệnh đề.

- Nếu hai mặt hàng có cùng mã hàng thì tên hàng giống nhau.
- Nếu hai mặt hàng có cùng đơn giá và số lượng thì có giá tiền giống nhau.

Như vậy, phụ thuộc hàm là một biểu thức logic nên phụ thuộc hàm có thể xem là một logic mệnh đề.

Cho tập biến $U = \{x_1, x_2, \dots, x_n\}$

Biến của Logic mệnh đề chỉ nhận giá trị 0 hoặc 1

Các phép toán phổ biến trong Logic mệnh đề

- Phép tuyển, ký hiệu là (\vee)
- Phép hội, ký hiệu là (\wedge)

- Phép phủ định, ký hiệu (\sim)
- Phép kéo theo, ký hiệu (\rightarrow)
- Phép tương đương, ký hiệu (\Leftrightarrow)

Qui ước

Khi viết

x_1, x_2 ta hiểu là $x_1 \wedge x_2$

$A \rightarrow B$ ta hiểu là $A' + B$ hay $A' \vee B$

$$\left. \begin{array}{l} X = x_1, x_2, \dots, x_k \\ Y = y_1, y_2, \dots, y_m \end{array} \right\} \in U$$

$X \rightarrow Y$ tương đương với

$$x_1 \wedge x_2 \wedge \dots \wedge x_k \rightarrow y_1 \wedge y_2 \wedge \dots \wedge y_m$$

Cho tập thuộc tính $A(u, v)$

Ta có $u.A = v.A$ nghĩa là tại đó trị logic bằng 1

$u.A \neq v.A$ nghĩa là tại đó trị logic bằng 0

Thí dụ 1.2.5 Cho quan hệ bán hàng R , ta có bảng trị V_R

Bảng 1.6 Minh họa bảng trị của quan hệ bán hàng

	Mã hàng	Tên hàng	Đơn giá	Số lượng	Thành tiền
t1&t2	0	0	0	0	0
t1&t3	1	1	1	0	0
t1&t4	0	0	0	0	0
t1&t5	0	0	0	0	0
t2&t3	0	0	0	0	0
t2&t4	0	0	0	0	0
t2&t5	1	1	1	0	0
t3&t4	0	0	1	0	0
t3&t5	0	0	0	0	0
t4&t5	0	0	0	0	0

1.3 Lược đồ quan hệ

Định nghĩa

Lược đồ quan hệ (LĐQH) là một cặp $a = (U, F)$, trong đó U là tập hữu hạn các thuộc tính, F là tập các phụ thuộc hàm trên các miền trị (dữ liệu) của các thuộc tính trong U .

Thí dụ 1

Cho lược đồ quan hệ $a = (T, F)$, trong đó $T(ABCDEFGH)$ và tập phụ thuộc hàm

$$F_T = \{f1:B \rightarrow A; f2:DA \rightarrow CE; f3:D \rightarrow H; f4:GH \rightarrow C; f5:AC \rightarrow D\}$$

Thí dụ 2

Cho lược đồ quan hệ $a = (Q, F)$, trong đó $Q(\text{MaSV}, \text{Tên}, \text{NS}, \text{DC}, \text{TênLop}, \text{KhoaHoc}, \text{MaMH}, \text{TênMH}, \text{Diem})$ và tập PTH $F = \{f1: \text{MaSV} \rightarrow \text{Tên}, \text{NS}, \text{DC}, \text{TênLop}; f2: \text{TênLop} \rightarrow \text{KhoaHoc}; f3: \text{MaMH} \rightarrow \text{TênMH}; f4: \text{TênMH} \rightarrow \text{MaMH}; f5: \text{MaSV}, \text{MaMH} \rightarrow \text{Diem}\}$.

Thí dụ 3

Cho lược đồ quan hệ $a = (\text{SINH VIÊN}, F)$, trong đó quan hệ $\text{SINHVIEN}(\text{MaSV}, \text{Họ tên}, \text{Quê quán}, \text{Năm sinh})$ và tập phụ thuộc hàm $F = \{f1: \text{MaSV} \rightarrow \text{Họ tên}; f2: \text{MaSV}, \text{Họ tên} \rightarrow \text{Quê quán}; f3: \text{MaSV}, \text{Họ tên} \rightarrow \text{Năm sinh}\}$.

1.4 Kết chương

Chương này đã trình bày một số khái niệm cơ bản liên quan đến các lớp phụ thuộc hàm, các tính chất của phụ thuộc hàm, tiên đề, lược đồ quan hệ và các thí dụ minh họa.

Chương 2

MỘT SỐ KHÁI NIỆM CƠ SỞ TRONG PHỤ THUỘC BOOLE DƯƠNG

Trong chương 1 đã nêu tổng quan về lớp phụ thuộc logic đầu tiên là lớp phụ thuộc hàm, cùng với các tính chất của nó. Trong chương 2 sẽ tiếp tục trình bày một số khái niệm cơ bản về lớp phụ thuộc được mở rộng từ phụ thuộc hàm theo trình tự thời gian, đó là lớp phụ thuộc Boole dương.

Nội dung chương này sẽ trình bày một số khái niệm về phụ thuộc Bool dương và các vấn đề liên quan đến bài toán thành viên đối với lớp phụ thuộc Boole dương. Ngoài ra, trong chương này cũng đề cập đến quy trình chuẩn hóa một công thức về dạng chuẩn hội (CNF), phương pháp chứng minh một công thức bằng phép hợp giải và thuật toán xây dựng tập PTBD thỏa mãn quan hệ R cho trước.

2.1 Biểu thức Boole

Bảng 2.1 Mô tả mối liên hệ giữa PTH và Biểu thức Boole

Phụ thuộc hàm	Biểu thức Boole
$X \rightarrow Y$	$\wedge X \rightarrow \wedge Y$
Thỏa trong R	$T_R \subseteq T_{X \rightarrow Y}$

Thí dụ 2.1 Cho biểu thức Boole $f: a \rightarrow bc$

Bảng 2.2 Minh họa bảng trị và bảng chân lý của biểu thức Boole

a	b	c	bc	$f: a \rightarrow bc$	Bảng trị V_f				Bảng chân lý T_f			
0	0	0	0	1	a	b	c	bc	a	b	c	bc
0	0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	0
0	1	1	1	1	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	□	1	0	1	1	1
1	0	1	0	0	1	0	0	0	1	1	1	1
1	1	0	0	0	1	0	1	0	1	1	1	1
1	1	1	1	1	1	1	0	0	1	1	1	1
					1	1	1	1				

Nhận xét:

- Bảng trị V_f của biểu thức Boole chứa trị của mọi phép gán trị cho f .
- Bảng chân lý của biểu thức Boole chỉ chứa giá trị 1 của f .

Thí dụ 2.2 Cho biểu thức Boole $f: a \vee b \rightarrow bc$

Bảng 2.2 Minh họa bảng trị và bảng chân lý của biểu thức Boole

a	b	c	bc	$f: a \rightarrow bc$	Bảng trị V_f				Bảng chân lý T_f			
0	0	0	0	1	a	b	c	bc	a	b	c	bc
0	0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	0
0	1	1	1	1	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	1	0	0	1	1	1
1	0	1	0	0	1	0	0	0	1	0	1	1
1	1	0	0	0	1	0	1	0	1	1	0	0
1	1	1	1	1	1	1	0	0	1	1	1	1
					1	1	1	1				

2.2 Phụ thuộc Boole dương

Tiếp theo một số phụ thuộc logic như phụ thuộc hàm, phụ thuộc đối ngẫu, phụ thuộc mạnh, phụ thuộc yếu, phụ thuộc sai khác, ... việc đề xuất mở rộng khái niệm phụ thuộc hàm sang phụ thuộc Boole dương của các nhóm nghiên cứu độc lập với nhau (J Berman và W.J.Blok và Sagiv, Delobel et al) đã góp phần quan trọng trong việc phát triển các lớp phụ thuộc trong cơ sở dữ liệu. Phụ thuộc Boole dương là một dạng của các lớp phụ thuộc Logic bao gồm các ràng buộc dữ liệu được mô tả thông qua các công thức Boole dương là những công thức nhận giá trị (1) khi tất cả các biến đều có trị là 1.

Trong mô tả phụ thuộc hàm và phụ thuộc Boole dương phép sánh trị của thuộc tính vẫn là phép sánh đẳng thức ($=$). Đối với phụ thuộc Boole dương *định lý tương đương* [2] vẫn bảo toàn hiệu lực.

2.2.1 Công thức Boole dương

Định nghĩa 2.1

Công thức $f \in L(U)$ được gọi là *công thức Boole dương (CTBD)* nếu $f(e) = 1$, trong đó e là phép gán trị đơn vị, $e = (1, 1, \dots, 1)$.

Ký hiệu $P(U)$ là tập toàn bộ *các công thức dương* trên U . Ta có thể xem $P(U)$ bao gồm các công thức được xây dựng từ các phép toán $\wedge, \vee, \rightarrow$ và hằng 1.

Các công thức sau đây là các CTBD

- $A \wedge B \wedge C$
- $(A \wedge B) \vee (C \wedge B)$
- $(A \wedge B) \vee ((C \wedge B) \rightarrow B) \vee C \vee 1$

- Các công thức suy dẫn, suy dẫn mạnh, yếu và đối ngẫu đều là các CTBD.

- Công thức suy dẫn

$$P \wedge Q \rightarrow P \wedge Q$$

- Công thức suy dẫn mạnh

$$P \vee Q \rightarrow P \wedge Q$$

- Công thức suy dẫn yếu

$$P \wedge Q \rightarrow P \vee Q$$

- Công thức suy dẫn đối ngẫu

$$P \vee Q \rightarrow P \vee Q$$

3. Công thức $A \wedge (\neg B)$ không phải là CTBD, vì khi $A = B = 1$ ta có $1 \wedge 0 = 0$.

2.2.3 Một số tính chất của công thức Boole dương

Mệnh đề 2.3

Tích các công thức Boole dương (CTBD) là một CTBD. Tổng các CTBD là một CTBD.

Chứng minh

Gọi f và g là hai CTBD và e là phép gán trị đơn vị thì $f(e) = g(e) = 1$, do đó $f(e)g(e) = 1.1 = 1$ và $f(e) + g(e) = 1+1 = 1$.

Định nghĩa 2.4

Mỗi tập CTBD $F = \{f_1, \dots, f_k\}$ được hiểu là một CTBD duy nhất có dạng tích như sau: $F = f_1 \wedge \dots \wedge f_k$.

Định nghĩa 2.5

Tập CTBD F suy dẫn ra được tập CTBD G nếu $F \Rightarrow G$ với mọi G trong G .

Tập CTBD F và G tương đương, $F \Leftrightarrow G$ nếu $F \Rightarrow G$ và $G \Rightarrow F$. Nếu $G \Leftrightarrow F$ thì G là phủ của F .

Mệnh đề 2.6

Tập CTBD được chuẩn hóa CNF nếu mọi công thức thành phần của nó đều có dạng CNF.

Thí dụ

$$F = \{(a+b)(c+a), (a'+b)c\}.$$

Mệnh đề 2.7

Mọi tập CTBD đều có thể được chuẩn hóa CNF.

Thí dụ 1

Cho $g = (a+b)(c'+a)$ là một CTB CNF. Hãy đưa g' về dạng CNF?

$$\text{Ta có } g' = [(a+b)(c'+a)]' = (a+b)' + (c'+a)' = a'b' + ca' =$$

$$(a'+c)(a'+a')(b'+c)(b'+a') = (a'+c)(b'+c)(b'+a')a'.$$

Thí dụ 2

Cho $g = (a'+b)(a+c'+d)e$ là một CTB CNF. Hãy đưa g' về dạng CNF?

$$g' = (a+a'+e')(a+c+e')(a+d'+e')(b'+a'+e')(b'+c+e')(b'+d'+e').$$

Định nghĩa 2.8

- Tập CTBD F được gọi là không dư nếu với mọi f thuộc về tập F ta có $F \setminus \{f\}$ không tương đương với F .
- Tập CTBD G được gọi là một phủ không dư của tập CTBD nếu G là một phủ của F và G là tập không dư.

2.2.4 Định lý tương đương

Đối với phụ thuộc hàm ba loại suy dẫn sau đây là tương đương

(i) Suy dẫn logic: $F \models f$,

(ii) Suy dẫn theo quan hệ: $F \vdash f$,

(iii) Suy dẫn theo quan hệ có không quá hai bộ: $F \vdash_2 f$.

Vấn đề này đã được chứng minh trong tài liệu [2].

Định lý tương đương đối với phụ thuộc Boole dương cho ba loại suy dẫn

- Suy dẫn logic $F \models f$ nếu $f \rightarrow g$ khi $T_f \subseteq T_g$.
- Suy dẫn theo quan hệ $F \vdash f$ nếu thỏa F thì thỏa f .
- Suy dẫn theo quan hệ không qua 2 bộ $F \vdash_2 f$ nếu thỏa F thì cũng thỏa f .

Chứng minh:

Cho tập thuộc tính $U = \{a, b\}$; $f: a \rightarrow b$ và $g: a' \rightarrow a$

Giả sử $F \models f$ nên $f \rightarrow g$, ta đi chứng minh $T_f \subseteq T_g$

Ta lập các bảng trị của f và g

Bảng 2.3 Minh họa bảng trị, bảng chân lý của công thức Boole

V_f			V_g			T_f		T_g	
a	B	f	a	a'	f	a	b	a	a'
0	0	1	0	1	0	0	0	1	0
0	1	1	1	0	1	0	1		
1	0	0				1	1		
1	1	1							

Nhận xét $T_f \not\subseteq T_g$ nên $f \nleftrightarrow g$

- Đối với quan hệ không quá 02 bộ

Với quan hệ rỗng, ta có $T_R = \emptyset$ nên $f \rightarrow g$

Với quan hệ có 01 bộ

R

a	b

Nhận xét: T_R chứa phép gán trị đơn vị $e(1, 1)$

Với quan hệ có 02 bộ

R	a	b	e	
			0	0
			0	1
u			1	<input type="checkbox"/>
v				

Nhận xét: $T_R \subseteq T_f$

Tóm lại, đối với phụ thuộc Boole dương thì 03 loại suy dẫn sau đây là tương đương

- Suy dẫn logic $F \models f$.
- Suy dẫn theo quan hệ $F \vdash f$.
- Suy dẫn theo quan hệ không qua 2 bộ $F \vdash_2 f$.

Như vậy, để giải quyết bài toán $f \models g$ hay không? Ta có thể tiến hành theo hai cách.

- Cách 1: Lập bảng trị để biết $T_f \subseteq T_g$ hay không?

- Cách 2: Ta dùng phép hợp giải (do Robinson đề xướng vào giữa thập niên 60, nó hoạt động dựa trên phương pháp chứng minh phản chứng).

Tuy nhiên, đối với các quan hệ với số lượng khoảng 1000 bộ trở lên thì việc giải quyết bài toán suy dẫn theo cách 1 gặp rất nhiều khó khăn. Do đó, người ta thường áp dụng phép hợp giải.

2.2.5 Quy trình áp dụng phép hợp giải

Xét thí dụ: $f \models g$?

Yêu cầu $f \rightarrow g$?

Để giải quyết vấn đề ta, tiến hành theo các bước sau,

- Bước 1: Đặt $h = f \rightarrow g$ và xét fg'

- Bước 2: Đưa h về dạng chuẩn hội CNF (Conjunctive Normal Form). Dạng chuẩn hội tích các tổng V.

- Bước 3: Thay các biểu thức trong V có dạng $(A+P)(A'+Q)$ bằng $(P+Q)$, sau khi thực hiện các bước thay thế rút gọn,

+ Nếu nhận được kết quả 0 thì kết luận $f \rightarrow g$.

+ Nếu nhận được kết quả khác 0 thì kết luận $f \not\models g$.

Các bước tiến hành chuẩn hóa dạng chuẩn hội (CNF)

- Bước 1: Đưa các dấu phủ định \sim lên trước.

- Bước 2: Thay xor (\oplus), \rightarrow , \leftrightarrow bằng tổ hợp các phép toán \sim , $\&$, $+$.

- Bước 3: Thay luật phân phối bằng 2 tích: $ab + c = c + ab = (a+c)(b+c)$.

Một số luật thường sử dụng

Bảng 2.4 Một số luật logic thường trong phép biến đổi tương đương

$a'' \equiv a$
$a \rightarrow b \equiv a' + b$
$(a \rightarrow b)' \equiv ab'$
$a(b+c) \equiv ab+ac$
$a+(bc) \equiv (a+b)(a+c)$
$(ab)' \equiv a' + b'$
$(a+b)' \equiv a'b'$
$(a \leftrightarrow b) \equiv (a \rightarrow b)(b \rightarrow a)$
$a \oplus b \equiv (a+b)(a'+b')$

* Quy tắc đối ngẫu

- Giá trị của một biểu thức sẽ không đổi khi ta thay phép cộng (+) hay phép tuyển (\vee) bằng phép nhân (*) hay phép hội (\wedge) và giá trị 1 bằng giá trị 0.

Xét thí dụ

$$\begin{aligned}
 & (a \rightarrow b) \wedge (a+1) \\
 & \equiv (a' + b) \wedge (a+1) \\
 & \equiv (a' \wedge b) \vee (a \wedge 0)
 \end{aligned}$$

2.2.6 Bài toán thành viên

Bài toán thành viên trong lược đồ quan hệ với phụ thuộc Boole dương là bài toán suy dẫn logic được phát biểu như sau,

Cho f và g là hai công thức Boole dương từ tập phụ thuộc Boole dương cho trước, yêu cầu của bài toán là kiểm tra xem $f \models g$?

Nhận xét

Khi quản trị CSDL, người ta thường cho trước một tập các phụ thuộc Boole dương F , sau đó với mỗi công thức Boole dương f , ta phải xác định $F \models f$?

Nếu biến đổi trước F về dạng tiện lợi C nào đấy thì mỗi lần giải bài toán trên chỉ việc gọi thủ tục hợp giải với tham trị là C và f , trong đó C đã được thu gọn.

Từ định lý tương đương cho lớp các PTBD, ta có nhận xét:

$$F \models f \equiv F \rightarrow f \equiv \neg F \vee f.$$

Đặt $E \equiv \neg F \vee f$, ta có $\neg E \equiv \neg(\neg F \vee f) \equiv F \wedge \neg f$.

Theo giả thiết F là tập các PTBD do đó trước hết cần đưa F về dạng chuẩn hội và thực hiện các bước hợp giải đến mức tối đa.

Do đó, ta vận dụng kết quả của phép hợp giải và thuật toán Reduction (thu gọn PTBD) để giải quyết bài toán thành viên.

Thuật toán Member

Algorithm Member_PTBD

Function: Giải bài toán thành viên $F \models f$

Input: - Tập PTBDTQ F

- PTBDTQ f

Output: - True nếu $F \models f$;

- False nếu $F \not\models f$;

Method

1. $C := \text{Reduce}(F) \wedge \text{cnf}(\neg f)$

2. *return* Resolution(C);

End Member.

Gọi thủ tục thu gọn tập phụ thuộc Boole dương tổng quát

Algorithm Reduction

Function: Thu gọn tập PTBDTQ

Input: Tập PTBDTQ F

Output: Công thức thu gọn C của F

Method

1. Đưa F về dạng chuẩn hội:

$C := \emptyset;$

for each member f in F do

$C := C \wedge \text{cnf}(f);$

endfor;

2. return (Unification(C));

End Reduction.

Phép hợp giải

Algorithm Unification

Function: Hợp nhất các nhân tử trong CTBD dạng chuẩn hội C đến tối đa.

Input: Công thức dương C dạng chuẩn hội

$C = C_1 \wedge C_2 \wedge \dots \wedge C_k;$

Output: công thức sau khi hợp nhất.

Method

while (còn xử lý được) do

Tìm hai nhân tử trong C có dạng

$C_i = (p \vee x)$ và $C_j = (q \vee \neg x);$

if (tìm được)

Thay C_i và C_j trong C bằng $(p \vee q)$

else break;

end if;

end while;

return $C;$

End Unification.

Thí dụ 2.6

Cho tập thuộc tính $U=(a \ b \ c)$, F và g là hai công thức Boole dương, trong đó

$$F=\{a \rightarrow b, a \rightarrow c\}; \ g: a \rightarrow bc$$

$$\text{Yêu cầu: } F \models g \equiv F \rightarrow g?$$

$$\text{Đặt } h = F \rightarrow g \equiv h = Fg'$$

$$h = (a \rightarrow b)(a \rightarrow c)(a \rightarrow bc)'$$

$$\equiv (a' + b)(a' + c)(bc)'a$$

$$\equiv (a' + b)(a' + c)(b' + c')a \quad \text{*/dạng CNF}$$

Hợp giải

$$\equiv (a' + b)(a' + c)(b' + c')a$$

$$\equiv (a' + b)(a' + b')a$$

$$\equiv a'a = 0$$

$$\text{Kết luận: } F \rightarrow g$$

2.2.7 Ưu điểm của Thuật toán Member so với các thuật toán khác

Ngoài thuật toán *Member* áp dụng phép hợp giải của Robinson được dùng để giải quyết bài toán thành viên trong tài liệu này còn có một số thuật toán khác cũng giải quyết vấn đề tương tự như thuật toán *Member*. Chẳng hạn như thuật toán dùng cây suy diễn của Vương Hạo [7].

Thuật toán Vương Hạo bao gồm các bước sau

- Bước 1: Phát biểu lại giả thiết và kết luận của vấn đề theo dạng chuẩn

$$GT_1, GT_2, \dots GT_n \rightarrow KL_1, KL_2, \dots KL_m$$

Trong đó các giả thiết GT_i và các kết luận KL_i là các biểu thức dạng chuẩn (chỉ chứa 03 phép toán cơ bản \vee, \wedge, \sim).

- Bước 2:

+ Nếu GT_i có chứa phép \vee thì tách thành hai dòng con.

+ Nếu KL có chứa phép \wedge thì tách thành hai dòng con.

Thí dụ

$p \wedge (\sim p \vee q) \rightarrow q$ thì tách thành

$p \wedge \sim p \rightarrow q$ và $p \wedge q \rightarrow q$

- Bước 3: Một dòng được chứng minh nếu thỏa một trong hai điều kiện

+ Tồn tại chung một mệnh đề ở cả hai phía.

Thí dụ

$p \wedge q \rightarrow q \vee r$

+ Tồn tại hai mệnh đề phủ định lẫn nhau (p và $\sim p$).

Thí dụ

$p \wedge \sim p \rightarrow q$

- Bước 4:

+ Nếu một dòng không còn phép tuyển (\vee) hoặc phép hội (\wedge) ở cả hai vế và ở hai vế không có chung một biến mệnh đề thì dòng đó không được chứng minh.

+ Một vấn đề được chứng minh nếu tất cả dòng dẫn xuất từ dạng chuẩn ban đầu đều được chứng minh.

Thí dụ

Cho tập các biểu thức logic $F = \{p \Rightarrow q, q \Rightarrow r, \neg p \Rightarrow (r \rightarrow r)\}$

Giải**Áp dụng thuật toán Vương Hạo**

- Bước 1 (đưa về dạng chuẩn)

$\neg p \vee q, \neg q \vee r, p \vee r \rightarrow r$

- Bước 2 (chuyển về các mệnh đề có dạng phủ định)

$\neg p \vee q, \neg q \vee r, p \vee r \rightarrow r$

- Bước 3 (Thay phép toán \vee ở GT và phép toán \wedge ở KL bằng dấu “,”)

$$\neg p \vee q, \neg q \vee r, p \vee r \rightarrow r$$

- Bước 4: Xét về trái, ta chia ra làm 2 trường hợp:

$$+ \text{TH1: } q, (\neg q \vee r), (p \vee r) \rightarrow r$$

$$+ \text{TH2: } \neg p, (\neg q \vee r), (p \vee r) \rightarrow r$$

- Xét:

$$+ \text{TH1: } r \rightarrow r, p, q$$

Đúng

$$+ \text{TH2: } p \rightarrow r, p, q$$

Đúng

Vậy giả thuyết \rightarrow kết luận đúng, do tất cả các dẫn xuất đều được chứng minh.

Nhận xét: Thuật toán *Member* có ưu điểm so với các thuật toán khác là nhờ tính đơn giản và không đòi hỏi chúng ta phải nhớ nhiều quy tắc một cách chi tiết khi áp dụng thuật toán, chúng ta chỉ cần nắm rõ một số luật biến đổi tương đương trong logic.

2.2.8 Xây dựng tập PTBD từ quan hệ R cho trước

Cho quan hệ R trên tập thuộc tính U . Xây dựng tập PTBD thỏa mãn R .

Để giải quyết bài toán trên, ta có thể sử dụng một thuật toán đơn giản là liệt kê toàn bộ các PTBD có trong $P(U)$, rồi kiểm tra PTBD nào thỏa trong quan hệ R thì đưa vào kết quả. Vì U là tập hữu hạn nên thuật toán sẽ kết thúc sau hữu hạn bước.

Thuật toán BD(R) tìm tập PTBDTQ F thỏa mãn quan hệ R cho trước như dưới đây.

*Algorithm BD**Input:* - Quan hệ R trên U *Output:* - Tập PTBDTQ $BD(R)$.*Method*1. Xây dựng bảng T_R ;2. Làm đóng T_R thu được bảng $T := \text{Closed\&_GPBD}(T_R)$;3. $G := DF(T)$;4. $F := \text{Nonredundant_GPBD}(G)$;*return* F ;*End BD.***2.2.9 Xác định công thức Boole dương từ một biểu thức Boole cho trước**

Với một biểu thức Boole bất kỳ ta có thể xác định xem biểu thức Boole đã cho có phải là một biểu thức Boole dương hay không?

Có hai cách để giải quyết vấn đề,

- Cách 1: Thay các biến trong biểu thức f với phép gán trị đơn vị $e(1,1)$, nếu $f(e)=1$ thì kết luận f là công thức Boole dương.

- Cách 2: Xây dựng thuật toán để xác định một biểu thức Boole bất kỳ có phải là một công thức Boole dương hay không?

Thuật toán xác định công thức Boole dương từ một biểu thức Boole bất kỳ.

*Algorithm_pos**Input:* - F biểu thức Boole bất kỳ*Output:* - True nếu $f(e)=1$ - False nếu $f(e)=0$ *Method*

1. Gán các biến của Boole với giá trị

2. Kiểm tra xem $f(e)=?1/0$

3. Trả về kết quả

End pos.

Thí dụ 1

Cho biểu thức Boole $f: (a \wedge b \vee c) \rightarrow a$

Yêu cầu $f(e)=1$?

$$(a \wedge b \vee c) \rightarrow a$$

$$\equiv (a \wedge b \vee c)' \vee a$$

$$\equiv (a' \vee b' \wedge c') \vee a$$

Gán trị đơn vị $e(1,1)$ cho các biến trong biểu thức Boole f

Kết quả: $f(e)=1$

Kết luận: f là một công thức Boole dương.

Thí dụ 2

Cho biểu thức Boole $f: A \wedge (\neg B)$

Yêu cầu $f(e)=1$?

Gán trị đơn vị $e(1,1)$ cho các biến trong biểu thức Boole f

Kết quả: $f(e)=0$

Kết luận: f không là một công thức Boole dương.

Nhận xét: Qua hai thí dụ trên thì việc xác định một biểu thức Boole có phải là một công thức Boole không được thực hiện khá dễ dàng. Tuy nhiên, đối với các biểu thức Boole chứa nhiều biến, nhiều toán tử phức tạp thì ta nên giải quyết vấn đề bằng cách áp dụng thuật toán Algorithm_pos.

2.2.10 Sự cần thiết của phụ thuộc Boole dương

Xét $U = \{A\}$, PT $f: A'$; $g: A$

Ta không có: $\neg A \rightarrow A$ (logic), Lí do $T_{A'}$ ko nằm trong T_A

A	A'	f: A'	g: A
0	1	*	
1	0		*

Mặt khác: Với mọi quan hệ $R(A)$ ta có $e = (1) \in TR$ (chứa vector đơn vị e).

Ta có: $T_A = \{e\}$; $T_{\neg A} = 0$ vector 0

$\neg A \Rightarrow R_A$, và $\neg A \Rightarrow 2A$ với nghĩa nếu R thỏa $\neg A$ thì R thỏa A .

Hay nếu $T_R \subseteq T_{\neg A}$ thì $T_R \subseteq T_A$ điều này luôn đúng vì $T_R \subseteq T_{\neg A}$.

2.10 Kết chương

Chương này đã trình bày một số khái niệm cơ bản liên quan đến các lớp phụ thuộc Boole dương, đồng thời cũng trình bày thuật toán giải bài toán thành viên cho lớp PTBD.

Chương 3

MỘT SỐ ỨNG DỤNG CỦA BÀI TOÁN THÀNH VIÊN

Vận dụng kết quả của bài toán thành viên chúng ta có thể ứng dụng để giải quyết một số vấn đề như tìm phủ không dư, tính bao đóng và tìm khóa của lược đồ quan hệ.

3.1 Ứng dụng bài toán thành viên để tìm phủ không dư

Định nghĩa 3.1

Cho hai tập PTBD F và G trên tập thuộc tính U . G gọi là phủ không dư của F nếu:

- 1) G là một phủ của F , và
- 2) G có dạng không dư: $\forall g \in G: G \setminus \{g\} \neq G$

Thuật toán tìm phủ không dư của tập PTBD

Algorithm Nonredundant

Format: Nonredundant(F)

Input: - Tập PTBD F

Output: - Một phủ không dư G của F

$G \equiv F$

$\forall g \in G: G \setminus \{g\} \neq G$

Method

$G := F;$

for each FD g in F do

if IsMember($g, G \setminus \{g\}$) then

$G := G \setminus \{g\};$

endif;

endfor;

return G ;

end Nonredundant.

Quy trình tìm phủ không dư

1. CNF: Chuyển các công thức f trong F sang dạng CNF. Nếu f là công thức hằng đúng thì loại.

Trùng lặp thì lược bớt.

2. Với mỗi CNF f trong F xây dựng một CNF cho f' .

3. Duyệt

Với mỗi f trong F :

Nếu $Ff' = 0$ thì $F = F \setminus \{f\}$; return F ;

Thí dụ 3.1.1

Tìm phủ không dư của $F = \{a' \vee b, a+b, b \rightarrow c\}$

Bảng 3.1 Minh họa các bước phủ không dư của thí dụ 3.1.1

	F	CNF F	CNF f'	Xét	Loại?
1	$a' \vee b$	$a'+b$	ab'	1	
2	$a+b$	$a+b$	$a'b'$	2	Loại
3	$b \rightarrow c$	$b'+c$	bc'	3	
Kết quả: $G = \{a'+b, b'+c\}$					

Thí dụ 3.1.2

Tìm phủ không dư của $F = \{a \rightarrow b, c' \vee b, ac \rightarrow b\}$

Bảng 3.1 Minh họa các bước phủ không dư của thí dụ 3.1.2

	F	CNF F	CNF f'	Xét	Loại?
1	$a \rightarrow b$	$a'+b$	ab'	1	
2	$c' \vee b$	$c'+b$	cb'	2	
3	$ac \rightarrow b$	$a'+c'+b$	acb'	3	Loại
Kết quả: $G = \{a'+b, c'+b\}$					

Thí dụ 3.1.3

Tìm phủ không dư của $F = \{a \rightarrow b, b \rightarrow c, a \rightarrow d, ab \rightarrow d, b'+c, acd \rightarrow d\}$.

Chuẩn hóa CNF: $F = \{a'+b, b'+c, a'+d, a'+b'+d, b'+c, a'+c'+d'+d\}$

Loại $a'+b'+d$ và $a'+c'+d'+d$ vì đây là các hằng đúng (tautology)

Khi đó $F = \{a'+b, b'+c, a'+d\}$

Đưa các phủ định của F về CNF: $F^* = \{ab', bc', ad'\}$

Bảng 3.3 Minh họa các bước phủ không dư của thí dụ 3.1.3

	F	CNF f'	Xét	Loại?
1	$a'+b$	ab'	1	
2	$b'+c$	bc'	2	
3	$a'+d$	ad'	3	
Kết quả: $G = \{a'+b, b'+c, a'+d\}$				

3.2 Ứng dụng bài toán thành viên để tính bao đóng

Định nghĩa 3.2.1

Cho LĐQH $a = (U, F)$, trong đó F là tập các PTBD và một tập con $X \subseteq U$.

Bao đóng của tập thuộc tính X trong LĐQH a là tập thuộc tính X^+ được xác định như sau:

$$X^+ = \{A \in U \mid X \rightarrow A\}$$

Thuật toán tìm bao đóng

Algorithm Closure

Format: Closure(X, F)

Input: - Tập PTBD F trên U

- Tập con thuộc tính X của U

Output: - $Y = X^+ = \{A \in U \mid X \rightarrow A\}$

Method

$Y := X;$

for each element A in $U \setminus X$ do

if member_GPBD($F, X \rightarrow A$) then

Add A to Y ;

endif;

endfor;

return Y ;

End Closure.

Thí dụ 1

Cho $F = \{a \rightarrow bc, b \wedge d, a \vee c\}$. Tính b^+ ?

Giải

Với tập F đã cho thì $U = abcd$. Ta tìm bao đóng của $X = b$ như sau:

$Y := \{b\}$;

Lần lượt xét các phần tử $A \in U \setminus X$. Kiểm tra xem nếu $F \models (X \rightarrow A)$ thì bổ sung A vào G . Sau đây là các bước thực hiện với $X = b$:

+ Trường hợp $A = a$: Ta có $G = \{b \rightarrow a\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + bc)(bd)(a + c)ba' \quad // \text{ Do } G = b \rightarrow a \text{ nên } G' = ba'$$

$$H = (a' + b)(a' + c)(bd)(a + c)ba'$$

$$H = (a' + b)(a' + c)(a + c)a'bd \quad // \text{ có dạng CNF}$$

- Hợp giải:

$$H = (a' + b)(c + c)a'bd = (a' + b)a'bcd \neq 0$$

+ Trường hợp $A = c$: Ta có $G = \{b \rightarrow c\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + bc)(bd)(a + c)bc' \quad // \text{ Do } G = b \rightarrow c \text{ nên } G' = bc'$$

$$H = (a' + b)(a' + c)(bd)(a + c)bc'$$

$$H = (a' + b)(a' + c)(a + c)bc'd \quad // \text{ có dạng CNF}$$

- Hợp giải:

$$H = (a' + b)(a' + c)(a + c)bc'd = (a' + b)(c + c)bc'd = (a' + b)cc'bd = 0$$

$$G := G \cup c = \{b, c\}$$

+ Trường hợp $A = d$: Ta có $G = \{b \rightarrow d\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + bc)(a + c)bd' \quad // \text{ Do } G = b \rightarrow d \text{ nên } G' = bd'$$

$$H = (a' + b)(a' + c)(a + c)bdd' \quad \text{có dạng CNF}$$

- Hợp giải:

$$H = (a' + b)(a' + c)(a + c)bdd' = (a' + b)(a' + c)(a + c)(dd')b = 0$$

$$G := G \cup d = \{b, c, d\}.$$

Kết luận

Vậy bao đóng của $X=b$ là $X^+ = bcd$.

Thí dụ 2

Cho $F = \{a \rightarrow b \vee c, b \wedge d, a \vee c\}$. Tính b^+ ?

Giải

Với tập F đã cho thì $U = abcd$. Ta tìm bao đóng của $X=b$ như sau:

$$Y := \{b\};$$

Lần lượt xét các phần tử $A \in U \setminus X$. Kiểm tra xem nếu $F \models (X \rightarrow A)$ thì bổ sung A vào G . Sau đây là các bước thực hiện với $X=b$:

+ Trường hợp $A=a$: Ta có $G = \{b \rightarrow a\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + b \vee c)(b \wedge d)(a + c)ba' \quad // \text{ Do } G = b \rightarrow a \text{ nên } G' = ba'$$

$$H = (a' + b + c)(bd)(a + c)ba' =$$

$$H = (a' + b + c)(a + c)a'bd \quad // \text{ có dạng CNF}$$

- Hợp giải:

$$H = (a' + b + c)(a + c)a'bd = (a' + b + c)(c + bd) = (a' + b + c)(b + c)(c + d) \neq 0$$

$$H = (a' + b + c)(b + c)(c + d) = (a' + b + c)(c + d) \neq 0$$

+ Trường hợp $A=c$: Ta có $G = \{b \rightarrow c\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + b + c)(bd)(a + c)bc' \quad // \text{ Do } G = b \rightarrow c \text{ nên } G' = bc'$$

$$H = (a' + b + c)(bd)(a + c)bc' = (a' + b + c)(a + c)c'bd$$

$$H = (a' + b + c)(a + c)c'bd \quad // \text{ có dạng CNF}$$

- Hợp giải:

$$H = (a' + b + c)(a + c)c'bd = (b + c)c'bd = bd \neq 0$$

+ Trường hợp $A=d$: Ta có $G = \{b \rightarrow d\}$. Kiểm tra $F \models G$:

- Đưa về CNF: Đặt $H = FG'$. Ta có:

$$H = (a' + b + c)bd(a + c)bd' \quad // \text{ Do } G = b \rightarrow d \text{ nên } G' = bd'$$

$H=(a'+b+c)bd(a+c)bd'$ có dạng CNF

- Hợp giải:

$$H=(a'+b+c)bd(a+c)bd' = (a'+b+c)(a'+c)bdd' = 0$$

$$G:=G \cup d = \{b, d\}.$$

Kết luận

Vậy bao đóng của $X=b$ là $X^+=bd$.

3.3 Ứng dụng bài toán thành viên để tìm khóa của lược đồ quan hệ

Định nghĩa 3.3

Cho lược đồ quan hệ $a = (U, F)$, trong đó U là tập các thuộc tính, F là tập các PTBD.

Khóa của lược đồ quan hệ là tập đủ nhỏ các thuộc tính có thể xác định không quá một bộ trong quan hệ, ký hiệu là K .

$$K \subseteq U$$

$$(i) \text{ K đủ mạnh: } K \rightarrow U$$

$$(ii) \text{ K đủ nhỏ: } \forall A \in K: K - \{A\} \not\rightarrow U$$

Hai tính chất trên tương đương,

$$(i') F \models (K \rightarrow U)$$

$$(ii') \forall A \in K: F \not\models (K - \{A\} \rightarrow U)$$

Thuật toán tìm khóa

Algorithm Key

Input: - F là PTBD trên U

- U là tập thuộc tính

Output: $K \subseteq U$

$$(i) \text{ member}(F, K \rightarrow U)$$

$$(ii) \forall A \in K: !\text{member}(F(K-A) \rightarrow U)$$

Method

$$K=U;$$

for each A in K do

```

    if member(F, (K - {A}) → U)
    then
        K := K - {A};
    endif;
endfor;
return K;
End Key.

```

Thí dụ 3.3.1

Cho lược đồ quan hệ $a = (U, F)$, trong đó $U = abc$ và $F = \{a \rightarrow bc, c + a\}$

Tìm $K = ?$

Giải

Theo giả thuyết $U=abc$ nên ta có thể khởi trị $K = U = abc$

Lần lượt loại bớt các thuộc tính $A \in K$ sao cho $K - \{A\} \rightarrow U$. Kiểm tra xem nếu $F \rightarrow (K - A)$ thì loại A ra khỏi K .

+ Trường hợp $A=a$: Ta có $K=bc$, đặt $g = bc$, tính $K^+ = ?$

Kiểm tra $F \models g$ hay $F \rightarrow g$?

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H = (a \rightarrow bc)(c+a)(bc)'$$

$$H = (a' + bc)(a+c)(b' + c') \quad \text{*/có dạng CNF}$$

- Hợp giải:

$$H(a' + bc)(a+c)(b' + c') = (a' + b)(a' + c)(a+c)(b' + c')$$

$$H = (a' + b)(c+c)(b' + c') = (a' + b)(b' + c')c = (a' + b)b' \neq 0$$

Nên $K=a$

+ Trường hợp $A=b$: Ta có $K=ac$, đặt $g = ac$, tính $K^+ = ?$

Kiểm tra $F \models g$ hay $F \rightarrow g$?

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H = (a \rightarrow bc)(c+a)(ac)'$$

$$H = (a' + bc)(a' + c')(a+c) \quad \text{*/dạng CNF}$$

- Hợp giải:

$$H=(a'+b)(a'+c)(a'+c')(a+c)=(a'+b)(a'+c)(aa'+cc')=0$$

Nên $K=K-\{b\}$

+ Trường hợp $A=c$: Ta có $K=ab$, đặt $g = ab$, tính $K^+ = ?$

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H=(a \rightarrow bc)(c+a)(ab)'$$

$$H=(a'+bc)(c+a)(a'+b') \quad */\text{dạng CNF}$$

- Hợp giải:

$$H=(a'+b)(a'+c)(a+c)(a'+b')=(a'a'+bb')(a'+c)(a+c)=a'(c+c)=a'c \neq 0$$

Nên $K:=K \cup c = \{ac\}$.

Vậy khoá của lược đồ quan hệ là $K=ac$.

Thí dụ 3.3.2

Cho lược đồ quan hệ $a = (U, F)$, trong đó $U = abcd$ và $F = \{ab \rightarrow c, d, c \rightarrow d\}$

Giải

Theo giả thuyết $U=abc$ nên ta có thể khởi trị $K = U = abcd$

Lần lượt loại bớt các thuộc tính $A \in K$ sao cho $K-\{A\} \rightarrow U$. Kiểm tra xem nếu $F \rightarrow (K-A)$ thì loại A ra khỏi K .

+ Trường hợp $A=a$: Ta có $K= bcd$, đặt $g = bcd$, tính $K^+ = ?$

Kiểm tra $F \models g$ hay $F \rightarrow g$?

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H=d(ab \rightarrow c)(c \rightarrow d)(bcd)'$$

$$H=((ab)'+c)(c'+d)(b'+c'+d')d$$

$$H=(a'+b'+c)(c'+d)(b'+c') \quad */\text{có dạng CNF}$$

- Hợp giải:

$$H=(a'+b'+c)(b'+c'+d)=(a'+b'+d) \neq 0$$

Nên $K=a$

+ Trường hợp $A=b$: Ta có $K=acd$, đặt $g = acd$, tính $K^+ = ?$

Kiểm tra $F \models g$ hay $F \rightarrow g$?

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H=d(ab \rightarrow c)(c \rightarrow d)(acd)'$$

$$H=d((ab)' + c)(c' + d)(acd)'$$

$$H=d(a' + b' + c)(c' + d)(a' + c' + d') \quad */\text{dạng CNF}$$

- Hợp giải:

$$H=d(a' + b' + c)(c' + d)(a' + c' + d') = (a' + b' + d')(c' + d)d$$

$$H=(a' + b' + d')(c' + d) = (a' + b') \neq 0$$

$$\text{Nên } K=K \cup b$$

+ Trường hợp $A=c$: Ta có $K=abd$, đặt $g = abd$, tính $K^+ = ?$

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H=d(ab \rightarrow c)(c \rightarrow d)(abd)'$$

$$H=d((ab)' + c)(c' + d)(a' + b' + d')$$

$$H=d(a' + b' + c)(c' + d)(a' + b' + d') \quad */\text{dạng CNF}$$

- Hợp giải:

$$H=d(a' + b' + c)(c' + d)(a' + b' + d')$$

$$H=(a' + b' + d)dd' = 0$$

$$K:=K - \{c\}$$

+ Trường hợp $A=d$: Ta có $K=abc$, đặt $g = abc$, tính $K^+ = ?$

- Đưa về CNF: Đặt $H=FG'$. Ta có:

$$H=d(ab \rightarrow c)(c \rightarrow d)(abc)'$$

$$H=d((ab)' + c)(c' + d)(a' + b' + c')$$

$$H=d(a' + b' + c)(c' + d)(a' + b' + c') \quad */\text{dạng CNF}$$

- Hợp giải:

$$H=d(a' + b' + c)(c' + d)(a' + b' + c')$$

$$H=(a' + b' + d)cc' = 0$$

$$K:=K - \{d\}$$

Vậy khoá của lược đồ quan hệ là $K=ab$.

3.4 Xây dựng chương trình

3.4.1 Cài đặt chương trình

Chương trình được xây dựng bằng ngôn ngữ Prolog, là ngôn ngữ lập trình logic được sử dụng phổ biến trong lĩnh vực trí tuệ nhân tạo. Nguyên lý lập trình logic dựa trên các mệnh đề Horn (Horn logic). Nội dung của chương trình nhằm giải bài toán thành viên với dữ liệu là các biểu thức Boole.

3.4.2 Kết quả thực nghiệm

Khi thực hiện các lệnh gọi các thủ tục chương trình sẽ giải một cách nhanh chóng bài toán thành viên với các phụ thuộc boole dương f và g và cho ra kết quả.

Ngoài ra, chương trình còn được ứng dụng để tìm phủ không dư của tập phụ thuộc Boole dương và thực hiện việc tính bao đóng, tìm khóa trên lược đồ quan hệ với các tập phụ thuộc Boole dương cho trước.

3.5 Kết chương

Chương này đã trình bày một số ứng dụng của bài toán thành viên cho các lớp phụ thuộc Boole dương như tìm phủ không dư của tập phụ thuộc Boole dương cho trước, ứng dụng bài toán thành viên trong việc tính bao đóng và tìm khoá trong lược đồ quan hệ với phụ thuộc Boole dương. Đồng thời trình bày cách thực hiện, việc đưa dữ liệu và kết quả của chương trình ứng dụng viết trên ngôn ngữ Prolog.

KẾT LUẬN VÀ KIẾN NGHỊ HƯỚNG PHÁT TRIỂN

Kết luận

Lý thuyết phụ thuộc dữ liệu luôn là vấn đề được nhiều người quan tâm vì nó đóng vai trò hết sức quan trọng trong việc thiết kế và cài đặt cơ sở dữ liệu. Hiện nay, các nhà nghiên cứu vẫn tiếp tục đề xuất nhiều loại phụ thuộc dữ liệu khác nhau để đáp ứng yêu cầu của các ứng dụng trong thực tế.

Do đó, việc mở rộng các lớp phụ thuộc có sẵn đang là hướng đi được nhiều nhà nghiên cứu lựa chọn. Thực tế cho thấy lớp phụ thuộc dữ liệu đầu tiên là phụ thuộc hàm đã được E. F. Codd [13] giới thiệu năm 1970. Ngày nay, cùng với sự phát triển của lớp phụ thuộc hàm, nhiều lớp phụ thuộc khác đã và đang được nghiên cứu, phát triển như phụ thuộc đối ngẫu, phụ thuộc mạnh, phụ thuộc yếu, phụ thuộc đối sánh, phụ thuộc tuần tự, phụ thuộc sai khác... và một số lớp phụ thuộc Boole dương.

Với mong muốn đóng góp vào sự phát triển của lý thuyết phụ thuộc dữ liệu, luận văn đã cố gắng tìm hiểu và trình bày một số khái niệm cơ bản về lớp phụ thuộc Boole dương để xây dựng ứng dụng giải quyết một số vấn đề của lớp phụ thuộc này như tính bao đóng, tìm khoa, tìm phủ không dư và bài toán thành viên.

Kiến nghị hướng phát triển tiếp theo

- Xây dựng các ứng dụng, giải quyết bài toán thực tế dựa vào lớp các phụ thuộc Boole dương.
- Tiếp tục tìm hiểu và tổng quát hóa một số lớp phụ thuộc dữ liệu có bản chất là phụ thuộc Boole dương được nghiên cứu gần đây như phụ thuộc hàm mềm, phụ thuộc hàm có điều kiện, phụ thuộc sai khác ...

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

- [1] Bùi Đức Minh, Lương Nguyễn Hoàng Hoa (2011), “Hệ sinh cân bằng và bài toán biểu diễn cơ sở hệ sinh ánh xạ đóng”, *Chuyên san các công trình nghiên cứu, phát triển và ứng dụng CNTT-TT*, Tập V-1, Số 5 (25), tr.15-21.
- [2] Nguyễn Xuân Huy (2006), *Các phụ thuộc logic trong cơ sở dữ liệu*, Viện KH&CN Việt nam, NXB Thống kê, Hà Nội.
- [3] Nguyễn Xuân Huy, Lê Đức Minh, Vũ Ngọc Loan (2000), “Các ánh xạ đóng và ứng dụng trong cơ sở dữ liệu”, *Tạp chí Tin học và Điều khiển học*, 16(4), tr.1-6.
- [4] Nguyễn Xuân Huy, Lê Thị Mỹ Hạnh (2005), “Giàn giao của ánh xạ đóng”, *Chuyên san các công trình nghiên cứu - triển khai viễn thông và công nghệ thông tin*, số 14, tr. 35-42.
- [5] Nguyễn Xuân Huy, Lê Thị Mỹ Hạnh (2005), “Thu gọn hệ sinh ánh xạ đóng”, *Chuyên san các công trình nghiên cứu - triển khai viễn thông và công nghệ thông tin*, số 15, tr. 53-58.
- [6] Nguyễn Xuân Huy, Lê Thị Mỹ Hạnh, Lương Nguyễn Hoàng Hoa, Bùi Đức Minh, Nguyễn Đức Vũ (2007), “Thiết kế cơ sở dữ liệu theo tiếp cận dịch chuyển lược đồ quan hệ”, *Kỷ yếu Hội thảo Khoa học Quốc gia “Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông”*, Đại Lải, 14-15/09/2007, NXB KHTN, tr.499-506.
- [7] Nguyễn Đình Thuận (2007), “Bài giảng Trí tuệ nhân tạo” Đại học Nha Trang tháng 8/2007.

Tài liệu tiếng Anh

- [8] Armstrong W.W., Delobel C. (1980), “Decomposition and Functional Dependencies in Relations”, *ACM TODS*, 5(4), pp. 404-430.

- [9] Bohannon, P., Fan, W., Geerts, F., Jia, X., and Kementsietsidis, A. (2007), “Conditional functional dependencies for data cleaning”, In *ICDE*, pp.746-755.
- [10] Cong, G., Fan, W., Geerts, F., Jia, X., and Ma, S. (2007), “Improving data quality: Consistency and accuracy”, In *VLDB*, pp.315-326.
- [11] Fan, W., Geerts, F., Lakshmanan, L. V. S., and Xiong, M. (2009), “Discovering conditional functional dependencies”, In *ICDE*, pp. 1231-1234.
- [12] Luong Nguyen Hoang Hoa (2011), “Some results concerning Generalized Positive Boolean Dependencies in relational database”, *International Journal of Computer Electrical Engineering (IJCEE)*, vol. 3, no. 6, pp. 779-783.
- [13] Song, S., and Chen, L. (2011), “Differential Dependencies: Reasoning and Discovery”, *ACM Trans. Datab. Syst.*, vol.9, no 4, Article 39.
- [14] Codd E. F. (1970), “A Relational Model of Data for Large Shared Data Banks”, *CACM* 13(6), pp. 377-387.

PHỤ LỤC

(Một số kết quả của chương trình ứng dụng)

- Gọi thực hiện lệnh “tv1.” để giải bài toán thành viên.

- Kích đúp chuột lên file chương trình là **thanhvien.pl**

- Màn hình hiển thị như sau:

Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 6.4.1)

Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam

*SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.*

Please visit <http://www.swi-prolog.org> for details.

For help, use?- help(Topic). or?- apropos(Word).

!?-

- Tại dấu nhắc ta đánh vào lệnh “tv.”

1 ?- tv.

$(a=>b) \& (b=>c) => a=>c?$

Given: $(a=>b) \& (b=>c) => (a=>c)$

$Y = \sim ((a=>b) \& (b=>c) => (a=>c))$

CNF: $(\sim a+b) \& (\sim b+c) \& (a \& \sim c)$

ToList: $[\sim a+b, \sim b+c, a, \sim c]$

List: $[[\sim a, b], [\sim b, c], [a], [\sim c]]$

[]

true !

Hình 3.1 Gọi thủ tục “tv.” để giải bài toán thành viên.

- Gọi thủ tục “go.” Để giải bài toán thành viên tiếp theo

2 ?- go.

$(a \Rightarrow b+c) \& (a \& d \Rightarrow e) \& (c+e \Rightarrow b) \Rightarrow b \& e \Rightarrow a+d?$

Given: $(a \Rightarrow b+c) \& (a \& d \Rightarrow e) \& (c+e \Rightarrow b) \Rightarrow (b \& e \Rightarrow a+d)$

$Y = \sim ((a \Rightarrow b+c) \& (a \& d \Rightarrow e) \& (c+e \Rightarrow b) \Rightarrow (b \& e \Rightarrow a+d))$

CNF: $(\sim a + (b+c)) \& (\sim a + \sim d + e) \& ((\sim c+b) \& (\sim e+b)) \& (b \& e \& (\sim a \& \sim d))$

ToList: $[\sim a + (b+c), \sim a + \sim d + e, \sim c+b, \sim e+b, b, e, \sim a, \sim d]$

List: $[[\sim a, b, c], [\sim a, \sim d, e], [\sim c, b], [\sim e, b], [b], [e], [\sim a], [\sim d]]$

$[[b], [e], [\sim a], [\sim d], [\sim a, b], [\sim a, \sim d, b]]$

By val $[0, 1, 0, 0, 1]$ *false !*

true

- Gọi thực hiện lệnh “tk1.” tìm khóa cho lược đồ quan hệ

4 ?- tk1.

$U = [a, b, c, d, e, g, h],$

$F = [a \Rightarrow b \& c, e \Rightarrow d, g \Rightarrow h, h \Rightarrow g]$

KEY TESTING

Given: $U = [a, b, c, d, e, g, h]$

$F = [a \Rightarrow b \& c, e, e \& d]$

key $= [a, g, h]$

true

Hình 3.2 Gọi thủ tục “tk1.” để tìm khóa của LĐQH

- Gọi thủ tục kiểm tra biểu thức logic có phải là biểu thức Boole dương

1 ?- pos($a \& b + c \Rightarrow a \& (\sim c)$).

Given: $a \& b + c \Rightarrow a \& \sim c$

Val = 0

false.

2 ?- pos($a \& b + c \Rightarrow a \& (c)$).

Given: $a \& b + c \Rightarrow a \& c$

Val = 1

true.

Hình 3.3 Gọi thủ tục kiểm tra biểu thức logic

- Gọi thủ tục tìm phủ không dư của tập PTBD

12 ?- nonred($[a \Rightarrow b, b \Rightarrow c, a \Rightarrow c], G$).

$G = [a \Rightarrow b, b \Rightarrow c, a \Rightarrow c]$.

13 ?- nonred($[a \Rightarrow b, b \Rightarrow c, a \Rightarrow c, (\sim a) + b], G$).

$G = [b \Rightarrow c, a \Rightarrow c, \sim a + b]$.

14 ?- nonred($[a \Rightarrow b, b \Rightarrow c, a \Rightarrow d, a \& b \Rightarrow d, (\sim b) + c, a \& c \& d \Rightarrow d], G$).

$G = [a \Rightarrow b, a \Rightarrow d, a \Rightarrow d]$

Hình 3.4 Gọi thủ tục tìm phủ không dư của tập PTBD

- Gọi thủ tục tính bao đóng của tập thuộc tính

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- closure([a,b,c,d,e], [a&b=>c,c&d=>e], [a,b], R).

R = [a, b, c].

4 ?- closure([a,b,c,d,e,f,g,h], [a&b=>c,c&d=>e,a+b,e+f], [a,b,e], R).

R = [a, b, c, e].

For help, use ?- help(Topic). or ?- apropos(Word).

5 ?- closure([a,b,c,d], [a&b=>c,d,c=>d], [a,b], R).

R = [a, b, c, d].

6?- closure([a,b,c,d,e], [a+b=>c,c&db=>e], [a,b], R).

R = [a, b, c].

Hình 3.5 Gọi thủ tục tính bao đóng của tập thuộc tính

CƠ SỞ DỮ LIỆU THỰC NGHIỆM

Việc ứng dụng các lớp phụ thuộc Boole trong cơ sở dữ liệu của các ứng dụng thực tế là rất cần thiết vì phụ thuộc Boole là một dạng phụ thuộc bậc cao. Nó đáp ứng được các yêu cầu ràng buộc trong cơ sở dữ liệu.

Trong cơ sở dữ liệu của một ứng dụng cung cấp lịch khám bệnh trực tuyến cho bệnh nhân. Dữ liệu được mô tả bao gồm các quan hệ sau:

Thông tin đăng nhập

Họ tên	Số tài khoản	Yêu cầu
Nguyễn A	7000215365	Khám tổng quát
...		

Thông tin bệnh nhân

STT	Họ tên	Giới tính	Năm sinh	Số tài khoản	Địa chỉ	Giờ khám
01	Nguyễn A	Nam	1989	7000215365	25 Lê Thánh Tôn	07 AM
...						

Biên lai viện phí

Mã BL	Số tiền VNĐ	STT Bệnh nhân	Số tài khoản
M01	100.000	01	7000215365
...			

Thông tin người khám bệnh

Mã VC	Họ tên	Giới tính	Năm sinh	Chuyên môn	Giờ khám
B01	Nguyễn Văn B	Nữ	1980	Da liễu	07 AM
...					

Yêu cầu:

Bệnh nhân sau khi đăng nhập vào ứng dụng nếu bệnh nhân cung cấp được số thứ tự lượt khám bệnh và số biên lai viện phí thì họ sẽ biết được lịch khám bệnh.

Tuy nhiên, số lượng bệnh nhân có thể đến đăng ký khám bệnh rất đông. Do đó, nếu bệnh nhân nhận số thứ tự lượt khám nhỏ hơn hoặc bằng 100 thì lịch khám bệnh sẽ tiến hành trong ngày. Ngược lại lịch khám bệnh của bệnh nhân sẽ được tiến hành vào các ngày kế tiếp.

Như vậy, trong trường hợp này với ràng buộc dữ liệu là phụ thuộc hàn thì không mô tả hết yêu cầu của cơ sở dữ liệu:

$$\text{STT, Số BL} \rightarrow \text{Giờ khám}$$

Khi đó ta cần mô tả ràng buộc dữ liệu trong trường hợp này với phụ thuộc Boole:

$$[\text{stt}(n \leq 100) \wedge \text{sobl}(m)] \rightarrow [\text{thoi gian}(t1 + t2)]$$