

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Thị Quyên

**NGHIÊN CỨU CƠ SỞ DỮ LIỆU SUY DIỄN
VÀ ỨNG DỤNG XÂY DỰNG HỆ THỐNG TÌM ĐƯỜNG ĐI**

Chuyên ngành: Khoa học máy tính
Mã số : 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI-2012

MỞ ĐẦU

Chúng ta có cơ sở dữ liệu từ thế giới thực đa dạng và phong phú như cơ sở dữ liệu đường biển, cơ sở dữ liệu các tuyến đường trong thành phố, giữa các huyện, tỉnh và thành phố trong một quốc gia,... Với hệ quản trị cơ sở dữ liệu quan hệ dùng đại số quan hệ hiện tại đang dùng thì việc đặt câu hỏi và trả lời câu hỏi còn hạn chế nhất là cơ sở dữ liệu có kích thước lớn như dữ liệu đường đi giữa các tuyến đường trong huyện, tỉnh, thành phố và cả nước, việc hỏi và trả lời câu hỏi tìm đường đi giữa hai điểm bất kỳ trên cơ sở dữ liệu thực tế có tồn tại hay không, và sau đó là câu hỏi tìm đường đi ngắn nhất giữa hai điểm bất kỳ là công việc hết sức khó khăn với đại số quan hệ mà chúng ta đang dùng, và vấn đề đệ quy thì trong đại số quan hệ không giải quyết được, hiện giờ trong SQL mới giải quyết được đệ quy tuyến tính nhờ sự mở rộng đại số quan hệ.

Với cơ sở dữ liệu suy diễn thì nó gồm các sự kiện (lấy dữ liệu từ cơ sở dữ liệu) và các luật sẽ cho chúng ta các câu trả lời từ câu hỏi, và cung cấp cho chúng ta những tri thức mới. Do vậy luận văn đã lựa chọn đề tài “*Nghiên cứu cơ sở dữ liệu suy diễn và ứng dụng xây dựng hệ thống tìm đường đi*” để giải quyết bài toán đặt ra nêu trên.

CHƯƠNG 1: LẬP TRÌNH LOGIC VÀ CƠ SỞ DỮ LIỆU

1. 1. Giới thiệu

1. 1. 1. Các mốc phát triển của lập luận logic và cơ sở dữ liệu suy diễn

Tiếp cận của Codd năm 1970, năm 1984 của F. D. Ullman, năm 1986 của Date, năm 1989 của Gardarin và Valurier, năm 1989 của Brodie và Manola. Tư tưởng cơ bản đằng sau lập trình logic là sử dụng logic toán học như ngôn ngữ lập trình, đã được đề cập trong tài liệu của Kowalski năm 1970, và được Colmerauer đưa vào thực hành năm 1975 trong các cài đặt ngôn ngữ lập trình logic Prolog (Programming logic).

1. 1. 2. Nhu cầu về cơ sở dữ liệu suy diễn

Các cơ sở dữ liệu suy diễn có thể quản lý hầu hết các ứng dụng hiện tại, các vấn đề phức tạp về dữ liệu.

1. 1. 3. Nhu cầu về hệ quản trị cơ sở dữ liệu thế hệ tiếp theo

Do nhu cầu của thực tế nghiên cứu và ứng dụng, một thế hệ mới của các hệ quản trị cơ sở dữ liệu mới được đề xuất. Đó là cơ sở dữ liệu hướng đối tượng, cơ sở dữ liệu suy diễn và cơ sở dữ liệu phân tán.

1. 1. 4. Hạn chế của hệ quản trị cơ sở dữ liệu quan hệ

Các đối tượng phức tạp, không xử lý tri thức, rút ra được các tri thức mới, môi trường ứng dụng phân tán

1. 2. Lập trình logic

1. 2. 1. Giới thiệu về lập trình logic

Lịch sử phát triển : thế hệ phát triển lập trình logic của Fischer Black năm 1964, James Slagle năm 1965 và Cordell Green năm 1969, các hệ thống hỏi đáp do McCarthy đề xuất. Năm 1973, Hayes phát triển ngôn ngữ Golux. Kowalski tập trung vào lời giải SL trong các thủ tục với đích suy giảm. Từ ngôn ngữ Prolog có các ngôn ngữ khác nhau như ALF, Fril, G \square del, Mercury, Oz, Ciao, Visual Prolog, XSB, và λ Prolog.

1. 2. 2. Một số định nghĩa

Hạng thức được định nghĩa đệ qui như sau: Mỗi hằng là một hạng thức; mỗi biến là một hạng thức; nếu f là ký hiệu hàm có n -ngôi và t_1, \dots, t_n là các hạng thức thì $f(t_1, \dots, t_n)$ là một hạng thức; hạng thức chỉ được sinh ra bởi các quy tắc trên.

Literal là một nguyên tố hoặc phủ định một nguyên tố. Một nguyên tố là literal dương và phủ định của nguyên tố là *literal âm*. Với p là một nguyên tố, *literal âm* được ký hiệu $\square p$.

1. 2. 3. Câu chương trình và đích

Câu không rỗng có thể được viết là $M_1 \vee \dots \vee M_n \vee \neg N_m, n + m \geq 1$, Trong đó M_i và N_j là các literal dương và các biến ngầm định đã có lượng từ \forall trong biểu thức tuyên. Mỗi công thức xác định tốt Wff và có thể viết ở dạng câu.

Đích chuẩn: Đích chuẩn, hay đơn giản là đích, hay sự phủ nhận là công thức có dạng $\leftarrow L_1 \wedge \dots \wedge L_n, n \geq 1$, trong đó mỗi L_i là nguyên tử hoặc phủ định của nguyên tử, phần $L_1 \wedge \dots \wedge L_n$ là thân đích.

1. 2. 4. Chương trình

Chương trình tổng quát (chương trình): là tập hữu hạn các câu tổng quát. Chương trình tuyến là tập hữu hạn các câu tuyến. Chương trình chuẩn là tập hữu hạn các câu chuẩn.

1. 2. 5. Chương trình phân tầng, đệ quy, phân cấp

Phân cấp: Nếu đồ thị phụ thuộc đối với chương trình chuẩn không chứa chu trình nào thì chương trình được gọi là phân cấp.

Chương trình đệ quy: Nếu đồ thị phụ thuộc đối với chương trình chuẩn có chu trình thì chương trình là đệ quy. Một chương trình đệ quy là đệ quy trực tiếp nếu đồ thị phụ thuộc của nó không có chu trình với độ dài lớn hơn 1.

Phân tầng: chương trình chuẩn là phân tầng nếu mỗi chu trình của đồ thị phụ thuộc được tạo chỉ bởi cung dương, cho dù phần còn lại của đồ thị có thể có cung âm.

1. 3. Cơ sở dữ liệu

1. 3. 1. Các mô hình dữ liệu

Mô hình dữ liệu mạng; mô hình dữ liệu phân cấp; mô hình dữ liệu quan hệ; mô hình dữ liệu đối tượng; mô hình dữ liệu suy diễn. Các mô hình này được nêu rõ trong [4] và [11].

1. 3. 2. Mô hình dữ liệu suy diễn

1. 3. 2. 1. Mô hình dữ liệu

Một mô hình hình dữ liệu gồm: Ký pháp toán học để mô tả hình thức dữ liệu và các quan hệ; kỹ thuật để xử lý dữ liệu như trả lời các câu hỏi, kiểm tra điều kiện toàn vẹn.

1. 3. 2. 2. Hệ thống cơ sở dữ liệu suy diễn và hệ thống lập trình logic

Một số điểm khác nhau giữa cơ sở dữ liệu suy diễn và lập trình logic gồm:

- Trong cơ sở dữ liệu suy diễn điển hình, số các sự kiện là lớn so với số các luật. Điều này không như vậy đối với hệ thống lập trình logic.
- Một cơ sở dữ liệu suy diễn chỉ tập các ký hiệu vị từ thành hai tập không giao nhau là tập tất các vị từ cơ sở và tập các vị từ ảo.. Ngược lại, lập trình logic không phân chia như vậy.
- Các cơ sở dữ liệu suy diễn có giao diện riêng để mô tả và xử lý dữ liệu, ở đó người ta dùng ngôn ngữ bậc cao. Trong các hệ thống lập trình logic, khả năng về các câu xử lý còn ở mức sơ khởi.
- Thuật ngữ dùng trong các hệ thống lập trình logic không giống như trong các hệ thống cơ sở dữ liệu suy diễn..

1. 3. 2. 3. Lý thuyết mô hình và lý thuyết chứng minh

Một cơ sở dữ liệu có thể được nhìn dưới quan điểm của logic theo: Diễn giải của lý thuyết bậc một; lý thuyết bậc một

1. 3. 2. 4. Lý thuyết mô hình với cơ sở dữ liệu suy diễn

Giả sử (D, L) là cơ sở dữ liệu chuẩn. Các tiên đề của lý thuyết T như sau:

1. Các tiên đề đầy đủ: Tiên đề có được do hoàn thiện mỗi ký hiệu vị từ của L tương ứng với các câu trong D.
2. Tiên đề duy nhất tên và về tương đương: Các tiên đề về lý thuyết tương đương tùy theo các ký hiệu hằng số, hàm số và vị từ của L.
3. Tiên đề về bao đóng của miền: Nếu a_1, \dots, a_p là tất cả những phần tử của L và f_1, \dots, f_q là các ký hiệu hàm số của L, thì tiên đề về bao đóng của miền, theo Lloyd năm 1987, Mancarella năm 1988 như sau:

$\forall x((x = a_1) \vee (\exists x_1, \dots, \exists x_m (x = f_1(x_1, \dots, x_m))) \vee \dots \vee (\exists y_1, \dots, \exists y_n (x = f_1(y_1, \dots, y_n))))$ Vì D là cơ sở dữ liệu chuẩn, quan điểm lý thuyết chứng minh như trên giảm về Comp (D) công với tiên đề về bao đóng của miền như vừa nêu.

1. 3. 2. 5. Các cơ sở dữ liệu suy diễn như các chương trình logic

Một cơ sở dữ liệu chuẩn gồm có:

- Một tập các câu chuẩn, trong đó các phần tử đều là bậc hạn chế
- Cách giải SLD đưa ra trong tài liệu [1]
- Một lý thuyết chứng minh đối với cơ sở dữ liệu xác định (D, L) có thể được xây dựng bằng cách chính xác như đối với cơ sở dữ liệu chuẩn. Tất nhiên theo quan điểm thực hiện thì một cơ sở dữ liệu xác định sẽ gồm:

- Một tập các câu xác định, trong đó các phần tử đều là bậc hạn chế
- Cách giải SLD

1. 3. 2. 6. Các giao tác trên các cơ sở dữ liệu suy diễn

Giao tác: Một giao tác trong cơ sở dữ liệu suy diễn là xâu hữu hạn của các phép toán, hay các hành động bổ sung, loại bỏ hay cập nhật các câu.

Khẳng định: Một giao tác được gọi là khẳng định tốt nếu toàn bộ xâu các phép toán tạo nên kết quả tốt của giao tác.

1. 3. 3 Các hệ quản trị cơ sở dữ liệu

1. 3. 3. 1. Hệ quản trị cơ sở dữ liệu.

Một hệ quản trị cơ sở dữ liệu là một tập hợp chương trình giúp cho người sử dụng tạo ra, duy trì và khai thác một cơ sở dữ liệu. Người ta gọi cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu bằng một thuật ngữ chung là hệ cơ sở dữ liệu.

1. 3. 3. 2. Sơ lược những hệ quản trị cơ sở dữ liệu

Giữa những năm 60, thế hệ đầu tiên là CODASYL và IMS. Thế hệ thứ hai, với mô hình quan hệ được đưa ra những năm 70 bởi E. F Codd. Thế hệ tiếp theo được phát triển từ những năm 80, theo mô hình hướng đối tượng; cơ sở dữ liệu phân tán.

1. 4. Kết luận

Chương này đã trình bày về các mốc phát triển của lập luận logic và cơ sở dữ liệu suy diễn; nhu cầu về cơ sở dữ liệu suy diễn; nhu cầu về hệ quản trị cơ sở dữ liệu; cơ sở dữ liệu và hạn chế của hệ quản trị cơ sở dữ liệu quan hệ; các mô hình dữ liệu và mô hình dữ liệu suy diễn; đồng thời giới thiệu về lập trình logic.

CHƯƠNG 2. CƠ SỞ DỮ LIỆU SUY DIỄN

2. 1. Cơ sở dữ liệu dựa trên logic

2. 1. 1. Cú pháp

Người ta sử dụng bộ các kí hiệu. Vị từ so sánh : <tên thuộc tính> θ <giá trị>. (Biến) so sánh với (giá trị). $\theta ::= \{<, >, <=, >=, =, <>\}$

Cách biểu diễn các luật : $Q \leftarrow P_1, P_2, \dots, P_n$ P_i : là các tiên đề, giả thiết, đích con, vị từ; Q : là kết luận hay là sự kiện. Nếu $n = 0$: $Q \leftarrow \approx$ Các sự kiện của CSDL cài đặt. Nếu $P \leftarrow P_1, P_2, \dots, P_n$ thì P là luật đệ quy (hay vị từ ở trong thân và đầu luật)

2. 1. 2. Ngữ nghĩa

Là tập tất cả các sự kiện được suy diễn ra từ chương trình DATALOG.

2. 1. 3. Cấu trúc cơ bản

Hệ thống DATALOG gồm hai loại quan hệ: Các quan hệ cơ sở được lưu trữ trong CSDL, còn gọi là CSDL mở rộng EDB (Extended Database); Các quan hệ suy diễn không cần lưu trong CSDL, dùng như quan hệ tạm thời, chứa kết quả trung gian khi trả lời câu hỏi. Các quan hệ này được gọi là CSDL theo mục đích IDB (Intentional Database).

2. 1. 4. Cấu trúc của câu hỏi

Để trình bày cấu trúc của câu hỏi người ta sử dụng đồ thị luật

2. 2. Cơ sở dữ liệu suy diễn

2. 2. 1. Mô hình cơ sở dữ liệu suy diễn

Mô hình dữ liệu gồm: Kí pháp toán học để mô tả hình thức dữ liệu và các quan hệ, và kỹ thuật để xử lý dữ liệu như trả lời các câu hỏi, kiểm tra điều kiện toàn vẹn.

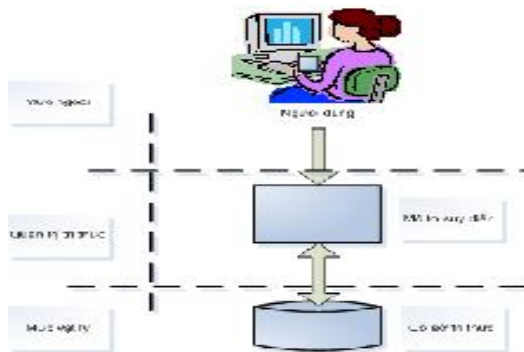
2. 2. 2 Các điều kiện toàn vẹn dữ liệu

Hai thành phần của tính toàn vẹn: tính hợp lệ và đầy đủ. Một cơ sở dữ liệu có tính toàn vẹn nếu dữ liệu của nó là chính xác (có giá trị) và nếu nó chứa đựng tất cả dữ liệu liên quan (nó là đầy đủ).

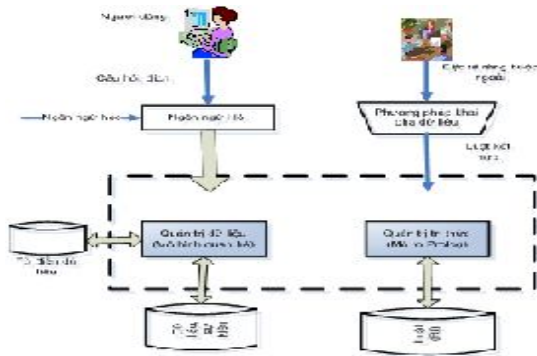
2. 2. 3 Quản trị cơ sở dữ liệu suy diễn

Một mô hình cơ sở dữ liệu suy diễn bao gồm:

1. Mức ngoài: cho phép xác định các yêu cầu của hệ thống, cho phép hỏi dữ liệu theo đích. Quá trình nhập dữ liệu và tri thức được thực hiện tại mức này. Thực tế, giao diện tương tác tạo điều kiện chuyển yêu cầu của người dùng sang câu hỏi phù hợp với hệ quản trị cơ sở dữ liệu.
2. Mức quản trị: gồm quản trị dữ liệu và quản trị tri thức. Tiếp cận sử dụng mô tơ suy luận của ngôn ngữ Prolog là phù hợp, đáp ứng yêu cầu về xử lý các khía cạnh literal âm, đặc tả mức cao.
3. Mức vật lý: có trách nhiệm lưu trữ và tổ chức các sự kiện, luật. Các tri thức dựa trên luật là phù hợp với suy luận của Prolog. Một vài dạng khác nhau của thể hiện tri thức sẽ được chuyển sang dạng luật.



Hình 2. 3. Mô hình suy diễn với 3 mức



Hình 2. 4. Kiến trúc của mô hình suy diễn với cơ sở dữ liệu động

2. 3. Ngôn ngữ Prolog

2. 3. 1. Chương trình và đích

Prolog là ngôn ngữ lập trình logic. Chương trình Prolog chuẩn là chương trình logic chuẩn. Theo định nghĩa của chương trình logic chuẩn, Chương trình prolog là tập hữu hạn các câu chuẩn. Đích Prolog chuẩn là đích chuẩn.

2. 3. 2. Cú pháp

Một số ký pháp sử dụng trong Prolog không xa lạ. Người ta sử dụng các ký hiệu: =, ←, ...

Kiểu dữ liệu đơn trong Prolog là term. Các khái niệm liên quan được trình bày trong mục 2. 2. 1. như term, nguyên tử, biến, term phức hợp.

2. 3. 3. Nền tảng lý thuyết

Cơ chế suy luận của Prolog giống như giải SLDNF với một số hạn chế sau:

1. Luật tính toán của Prolog luôn chọn các literal trái nhất trong đích
2. Prolog chuẩn sử dụng trật tự câu trong chương trình như trật tự cố định mà câu được xét, tức là tìm kiếm trên cây theo kiểu tìm chiều sâu.
3. Prolog bỏ qua điều kiện kiểm tra việc xảy ra trong phép so sánh khớp hai biểu thức

2. 4. Datalog

Datalog hay chương trình viết trên Datalog được xem là một cơ sở dữ liệu suy diễn. Tuy ngôn ngữ Datalog không phải Prolog, người ta vẫn quen lấy ví dụ trong ngôn ngữ Prolog làm ví dụ trong Datalog và xem Datalog hoạt động như Prolog. Ngôn ngữ Datalog được xét trên nền ngôn ngữ Prolog.

2. 5. Khả năng mở rộng của Datalog

- a. Những đặc trưng của quá trình xử lý câu hỏi.
- b. Các nghiên cứu hệ thống về các khía cạnh ràng buộc toàn vẹn.
- c. Các cơ sở dữ liệu suy diễn song song.
- d. Việc hình thức hóa các chức năng gộp lớn và các dữ liệu toàn vẹn.
- e. Như trong tài liệu [1] có một số mở rộng nghiên cứu về Datalog được trình bày là mở rộng vị từ, suy luận lùi, thay thế đồng nhất suy luận lùi dựa trên quan hệ, tính các quan hệ trên cây đích, đánh giá cây luật/đích, cây luật/đích, mẫu buộc duy nhất và sắp thứ tự đích con. Trong tài liệu [13] đưa ra các phương pháp ràng buộc toàn vẹn động và tĩnh.

2. 6. Kết luận

Trong chương này chúng ta đã trình bày về cơ sở dữ liệu suy diễn, cụ thể là trình bày về cơ sở dữ liệu dựa trên logic; định nghĩa cơ sở dữ liệu suy diễn; ngôn ngữ Prolog; Datalog và khả năng mở rộng của Datalog.

CHƯƠNG 3: THỬ NGHIỆM VỚI HỆ THỐNG DES

3. 1. Giới thiệu hệ thống DES 1.7.0

Hệ thống giáo dục Datalog (DES) là phần mềm mã nguồn mở, miễn phí, thích hợp với nhiều nền tảng, được thực hiện dựa trên Prolog của một hệ thống cơ sở dữ liệu suy diễn căn bản, nó thích ứng với ngôn ngữ Datalog và ngôn ngữ hỏi SQL.

3. 1. 1. Cài đặt DES 1.7.0

Có thể tải về từ trang web của DES tại địa chỉ: <http://des.sourceforge.net/>.

3. 1. 2. Các nguồn phân phối

Dưới nguồn phân phối, có vài thể hệ quản trị cơ sở dữ liệu duy diễn phụ thuộc vào trình biên dịch Prolog mà bạn chạy trên DES: Ciao Prolog, GNU Prolog, SICtus Prolog, và SWI Prolog.

3. 1. 3. Các thành phần để thực hiện chạy chương trình trên DES trên nền tảng Windows

Các thành phần để thực hiện chạy chương trình trên DES bao gồm: des.exe, deswin, des. Pl, des_debug. Pl, des_sql. Pl, des_glue. Pl, systems/{ciao, gnu, sicstus, swi}, main. sav.

. Dll, doc/manualDES<version>. Pdf, examples/. Dl, sp311

3. 2. Các thành phần tri thức và luật trong hệ thống trong DES

3. 2. 1. Ngôn ngữ hỏi

DES bao gồm trình biên dịch Datalog đơn giản đối với trạng thái hiện tại của nó, mà dựa vào máy cơ sở dữ liệu suy diễn mà có thể được hỏi với cả ngôn ngữ Datalog và ngôn ngữ SQL và giao diện Prolog cũng được cung cấp để làm sáng tỏ sự khác nhau giữa hệ thống Prolog và Datalog.

3. 2. 2. Các thành phần của Datalog

3.2.2.1. Cú pháp

Cú pháp của DES được lấy từ Prolog. Câu phức: như trong Prolog, chúng có dạng $t(t_1, t_2, \dots, t_n)$, trong đó t là một ký hiệu hàm và $t_i (1 \leq i \leq n)$ là các câu. Các toán tử có thể là: tiền tố và hậu tố.

3.2.2.2. Các luật

Các luật Datalog có dạng head :- body, hoặc là head. Cả hai đề kết thúc bằng một dấu chấm. Đầu Datalog là một nguyên tử dương mà dùng ký hiệu vị từ không được xây dựng sẵn. Thân Datalog chứa một dãy các literal được phân cách bằng một dấu phẩy có thể chứa các ký hiệu được xây dựng sẵn.

3.2.2.3. Các chương trình

Các chương trình DES bao gồm một tập các luật. Các chương trình có thể chứa ghi chú, các ghi chú bắt đầu với ký hiệu %, và kết thúc tại cuối dòng.

3.2.2.4. Các câu hỏi

Một câu hỏi (dương) là tên của một quan hệ với nhiều đối số như số chẵn của quan hệ (một literal dương). Các câu hỏi được phân loại trong hệ thống dấu nhắc của DES. Câu trả lời cho một câu hỏi là tập các sự kiện khớp với câu hỏi mà được suy diễn từ ngữ cảnh của chương trình, từ cả cơ sở dữ liệu bên trong (EDB) và bên ngoài (IDB).

3.2.2.5. Các khung nhìn tạm thời

Một khung nhìn tạm thời là một luật mà được thêm vào cơ sở dữ liệu; đầu của thân luật được xem xét như câu hỏi và được thực hiện.

3.2.2.6. Các khung nhìn tạm thời tự động

Các khung nhìn tạm thời tự động, Các khung nhìn tự động đáp ứng ngay, là các khung nhìn tạm thời mà không cần một đầu và cho phép viết các câu hỏi của phép giao.

3.2.2.7. Sự phủ định (negation)

DES đảm bảo rằng thông tin âm có thể được tập hợp lại từ chương trình với các đích phủ định được cung cấp mà một dạng hạn chế của sự phủ định được dùng.

3.2.2.8. Giá trị Null

Giá trị null được bao gồm trong mỗi ký hiệu chương trình để ký hiệu chưa biết, trong một cách tương đương, nó là một phần thừa kế của các hệ cơ sở dữ liệu quan hệ.

3.2.2.9. Các phép kết nối bên ngoài

Ba phép tính kết nối bên ngoài được cung cấp, theo ngôn ngữ truy vấn cơ sở dữ liệu quan hệ (SQL, đại số quan hệ mở rộng): kết nối trái, kết nối phải và kết nối bên ngoài.

3.2.2.10. Các tập hợp

Các tập hợp chỉ tới các hàm và các vị từ mà tính toán các giá trị tương ứng với một tập các giá trị thay vì các giá trị đơn. Các tập hợp được cung cấp bằng các cách của năm tính toán thông thường: sum, count, avg, min, và max.

3. 2. 3. Các câu lệnh trong DES

Các câu lệnh được thực thi bằng cách đặt trước câu lệnh với một dấu gạch chéo (/) tại dấu nhắc hệ thống DES. Chi tiết các câu lệnh của từng lại xem tại tài liệu [13]

3. 3. Bài toán tìm đường đi

3. 3. 1. Dạng dữ liệu của hệ thống thông tin địa lý

3. 3. 1. 1. Dữ liệu không gian

3.3.1.2. Dữ liệu vector

Một số kiểu đối tượng trong dữ liệu vector : Đối tượng điểm. Các đối tượng kiểu điểm có đặc điểm: là toạ độ đơn (x, y); Đối tượng đường : Đường được xác định như một tập hợp dãy của các điểm.

3.3.1.3. Dữ liệu raster

Mô hình raster biểu diễn không gian như là một ma trận số nguyên, mỗi giá trị số nguyên đại diện cho một thuộc tính, vị trí của số nguyên chính là vị trí của đối tượng

3. 3. 1. 4. Dữ liệu phi không gian

Số liệu phi không gian hay còn gọi là thuộc tính là những mô tả về đặc tính, đặc điểm và các hiện tượng xảy ra tại các vị trí địa lý xác định.

4. 3. 2. Tìm đường đi dùng cơ sở dữ liệu suy diễn

Ta có tập dữ liệu các đường đi và độ dài đường, vấn đề thứ nhất ở đây là tìm tất cả các con đường đi có thể giữa

hai điểm, thứ hai là tìm đường đi ngắn nhất giữa hai điểm.

3. 3. 2. 1. Biểu diễn các sự kiện và các luật dùng cơ sở dữ liệu suy diễn

Để biểu diễn các nội dung của một cơ sở dữ liệu suy diễn gồm các sự kiện và các luật, cũng giống như ngôn ngữ Prolog, chúng ta dùng Datalog. Một trong các việc mở rộng đối Datalog là khả năng dùng các tệp tin cơ sở dữ liệu bên ngoài đã và đang tồn tại như các phần sở hữu của cơ sở dữ liệu bên ngoài. Các tệp

tin này được dùng để giữ hầu hết các dữ liệu được mô tả tình huống thực tế của các mạng đường đi và thông tin về các trạng thái dự đoán trước.

Để tránh miêu tả cú pháp và ngữ nghĩa phức tạp trên dữ liệu thực tế cho phép em dùng khái niệm quan trọng của một chương trình Datalog bằng một ví dụ đơn giản để trợ giúp, mà rất gần với vấn đề giao thông mà chúng ta đang xây dựng – tức là định nghĩa của một đường đi của một mạng lưới giao thông.

$road(a, b). road(a, c). road(b, a). road(b, d). \dots$ (1)

$reach(X, Y) :- edge(X, Y).$ (2)

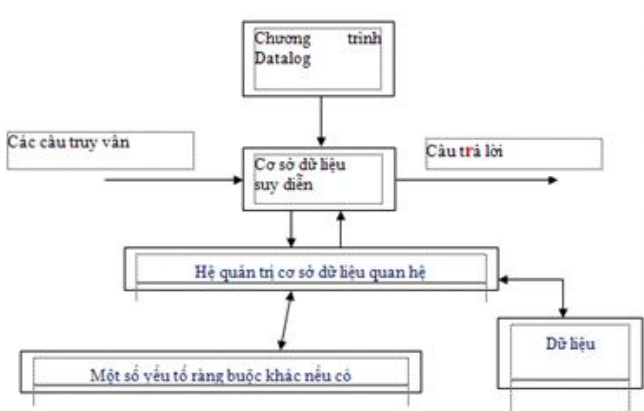
$reach(X, Y) :- reach(X, Z), edge(Z, Y).$ (3)

$reach(X, X).$ (4)

Dòng (1) định nghĩa các sự kiện, ví dụ: có một đường đi từ a tới b. Dòng (2) và (3) miêu tả các luật.

3. 3. 2. 2. Điều khiển tìm đường đi trên bản đồ giao thông

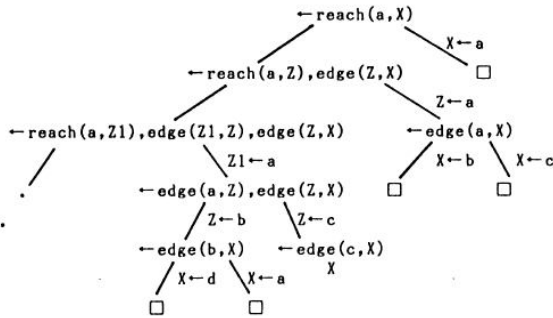
Luận văn trình bày vấn đề tìm đường đi tối ưu trong đồ thị hệ thống đường đi. Trước khi giới thiệu cấu trúc của dữ liệu đầu vào, chúng ta xem cấu trúc gợi ý của một hệ thống giao thông trong một bức tranh đơn giản sau:



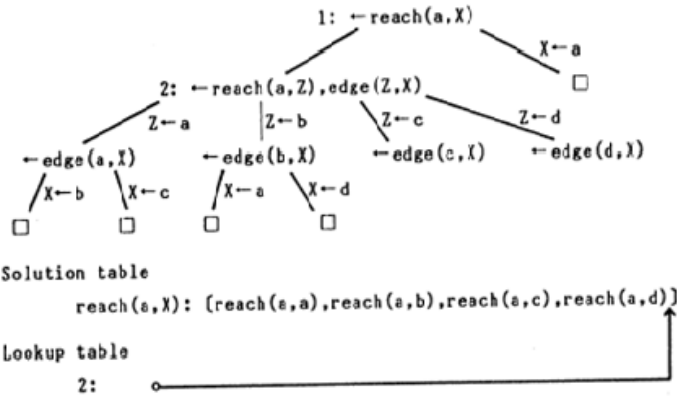
Hình 3. 15: Miêu tả hệ thống tìm đường đi

Trong hình 3. 16 thể hiện cây tìm kiếm, mà được gọi là cây OLD. Một câu truy vấn $\leftarrow reach(a, X)$, mỗi nút (được gán nhãn) là một trạng thái đích hoặc mệnh đề âm, và mỗi nút con của một nút là kết quả của việc áp dụng một mệnh đề trong chương trình đối với đích trái nhất (nguyên tử) của một trạng thái đích cha. Ký hiệu \square là ký hiệu trạng thái đích trống hoặc mệnh đề null. Mỗi cạnh được gán nhãn với tình huống con cho các biến của trạng thái đích cha cần thiết để tạo mệnh đề này có thể ứng dụng. một cây OLD là trường hợp đặc biệt của cây SLD thể hiện ở trong tài liệu [3] và [5].

Sau 1 số bước tìm kiếm, ta hoàn tất quá trình tìm kiếm



Hình 3. 16: Cây tìm kiếm OLD

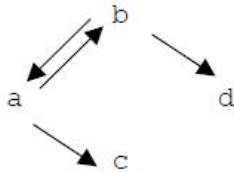


Hình 3. 20: Hoàn tất quá trình tìm kiếm các đường

3. 4. Giải bài toán trên DES 1.7.0

3. 4. 1. Giải bài toán tìm đường đi

Mô tả hình thức trên hệ thống DES như sau: bài toán trong mục 3. 3. 2. 2. dùng sự hồi quy trong DES bằng cách định nghĩa đồ thị như trong hình 3. 20 và một tập các bộ <xuất phát, đích đến> như một đường đi từ “xuất phát” tới “đích đến”.



Hình 3. 21: Các đường đi trong đồ thị

```

Dùng dữ liệu đưa vào SQL ở tệp tin “Paths. Sql”
create table edge (origin, destination);
insert into edge values ('a', 'b');insert into edge values ('a', 'c');
insert into edge values ('b', 'a');insert into edge values ('b', 'd');
create view paths (origin, destination) as with
recursive path (origin, destination) as (select * from edge)
union (select path. origin, edge. Destination from path, edge
where path. destination =edge. origin)
select * from path;
    
```

Đưa các luật vào trong tệp tin paths. dl chứa đoạn mã Datalog

```
% Paths in a Graph
edge (a, b). edge (a, c).
edge (b, a). edge (b, d).
path (X, Y) :- path (X, Z), edge (Z, Y).
           path (X, Y) :- edge (X, Y).
```

Trong phần chạy DES ta gõ đoạn mã lệnh sau vào:

```
DES-SQL> select * from paths;
answer (paths. origin, paths. destination) ->
{
    answer (a, a), answer (a, b), answer (a, c),
    answer (a, d), answer (b, a), answer (b, b),
    answer (b, c), answer (b, d)
}
```

Info: 8 tuples computed.

3. 4. 2. Giải bài toán tìm đường đi ngắn nhất

Thay đổi tập luật trong tập tin paths.dl như sau, ta sẽ tìm được đường đi ngắn nhất

```
path (X, Y, 1) :-edge (X, Y).
path (X, Y, L) :-path (X, Z, L0), edge (Z, Y), count (edge (A, B), Max), L0<Max, L is L0+1.
sp (X, Y, L) :-min (path (X, Y, Z), Z, L).
```

Câu truy vấn trả về tất cả các đường đi có khả năng và khoảng cách phụ thuộc nhỏ nhất:

```
DES-Datalog> sp (X, Y, L)
{
    sp (a, a, 2), sp (a, b, 1), sp (a, c, 1),
    sp (a, d, 2), sp (b, a, 1), sp (b, b, 2),
    sp (b, c, 2), sp (b, d, 1)
}
```

Info: 8 tuples computed

Dưới đây là việc xây dựng SQL cho vấn đề tương tự trong tập tin *Spaths.Sql*

```
DES-SQL> create or replace view spaths (origin, destination, length) as with
recursive path (origin, destination, length) as (select edge. *, 1 from edge)
union
```

```
(select path. origin, edge. destination, path. length+1 from path, edge where path. destination=edge. origin and
path. length< (select count (*) from edge)) select origin, destination, min (length) from path group by origin,
destination;
```

```
DES-SQL> select * from spaths
answer (spaths. origin, spaths. destination, spaths. length) ->
{
    answer (a, a, 2), answer (a, b, 1), answer (a, c, 1),
```

answer (a, d, 2), answer (b, a, 1), answer (b, b, 2),
 answer (b, c, 2), answer (b, d, 1)

}

Info: 8 tuples computed.

3. 5. Kết luận

Trong chương này, luận văn đã trình bày được vấn đề đặt ra của bài toán, cách thức mà các hệ thống thông tin địa lý lấy được dữ liệu từ thực tế, là bằng cách chụp ảnh vệ tinh, thông qua internet để phân phối ảnh vệ tinh, qua đó chúng ta lấy được hình ảnh và phân tích ảnh dựa và kỹ thuật Vector và Raster để đưa dữ liệu vào trong một cơ sở dữ liệu trên SQL và từ đó chúng ta có thể lấy dữ liệu ra để tìm tất cả các đường đi giữa hai điểm. sau đó mô tả bài toán dùng Datalog để biểu diễn các sự kiện và các luật, biểu diễn bằng đồ thị cho bài toán tìm đường đi dùng các luật. Cuối cùng chúng ta mô phỏng giải bài toán tìm đường đi một cách hình thức trong DES 1.7.0.

KẾT LUẬN

Những kết quả của luận văn

Theo như phần giới thiệu, luận văn chia thành các chương :

- Chương 1: Lập trình logic và cơ sở dữ liệu: giới thiệu về các mốc các mốc phát triển của lập luận logic và cơ sở dữ liệu suy diễn; nhu cầu về cơ sở dữ liệu suy diễn; nhu cầu về hệ quản trị cơ sở dữ liệu thế hệ tiếp theo; cơ sở dữ liệu và hạn chế của hệ quản trị cơ sở dữ liệu quan hệ; đồng thời giới thiệu về lập trình logic.
- Chương 2: Cơ sở dữ liệu suy diễn: tìm hiểu cơ sở dữ liệu dựa trên logic; định nghĩa cơ sở dữ liệu suy diễn; ngôn ngữ Prolog; Datalog và khả năng mở rộng của Datalog.
- Chương 3: Thử nghiệm trên hệ thống DES 1.7.0: Chúng ta đặt ra tìm hiểu và làm những vấn đề sau: Giới thiệu hệ thống DES 1.7.0; các thành phần tri thức và luật trong hệ thống trong DES; giải bài toán trên DES và đặt bài toán tìm đường đi.

Luận văn đã tìm hiểu được các vấn đề về lập trình logic và cơ sở dữ liệu, đồng thời cũng tìm hiểu được lý thuyết về cơ sở dữ liệu suy diễn để bước đầu làm lý thuyết áp dụng cho các bài toán trong thực tiễn

Một số phần của luận văn đã nêu lý thuyết logic, lập trình logic đã được đề xuất, và tìm hiểu về cơ sở dữ liệu suy diễn. Ứng dụng lý thuyết trình bày ứng dụng vào bài toán tìm đường đi trên thực tế và dùng DES 1.7.0 để thử nghiệm kết quả của luận văn.

Luận văn đã thực hiện mục đích :Nắm rõ kiến thức cơ bản về logic; Tìm hiểu lập trình logic và vai trò của lập trình logic; Làm rõ về cơ sở dữ liệu và các loại cơ sở dữ liệu; Kiến thức về cơ sở dữ liệu suy diễn, và ngôn ngữ Datalog; Tìm hiểu và cài đặt hệ thống DES (Datalog Educational System) 1.7.0; Ứng dụng lý thuyết tìm hiểu ở trên để giải quyết bài toán tìm đường đi; Cài đặt thử nghiệm bài toán tìm đường đi trên hệ thống DES 1.7.0

Phương hướng nghiên cứu tiếp theo

- Tiếp tục nghiên cứu ứng dụng bài toán tìm đường đi để tích hợp vào các hệ thống thông tin địa lý hiện tại và đồng thời mở rộng nghiên cứu những bài toán tương tự như tìm đường đi trong mạng máy tính, ...
- Tìm hiểu sâu về ràng buộc toàn vẹn trong cơ sở dữ liệu suy diễn.

DANH MỤC CÁC ĐỀ TÀI ĐÃ CÔNG BỐ CÓ LIÊN QUAN ĐẾN LUẬN VĂN

- [1] M. S. Lam, J. Whaley, V. B. Livshits, M. C. Martin, D. Avots, M. Carbin, and C. Unkel. Context-sensitive program analysis as database queries. In Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. ACM, June 2005.
- [2] Y. A. Liu and S. D. Stoller. From datalog rules to efficient programs with time and space guarantees. In PPDP '03: Proceedings of the 5th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming, pages 172–183. ACM Press, 2003.
- [3] R. Ramakrishnan and J. D. Ullman. A survey of research on deductive database systems. *J. Logic Programming*, 23 (2):125–149, 1993.
- [4] J. D. Ullman. Bottom-up beats top-down for datalog. In PODS '89: Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pages 140–149. ACM Press, 1989.
- [5] J. D. Ullman. Principles of Database and Knowledge-Base Systems. Computer Science Press, Rockville, MD., volume II edition, 1989.