# LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy cô giáo trong khoa Công nghệ thông tin cũng như cùng toàn thể các thầy cô giáo đã trực tiếp giảng dạy nhiệt tình học phần “Thực tập chuyên đề” cho chúng em trong suốt thời gian môn học này.

Chúng em xin bày tỏ lòng biết ơn đến cô Lưu Thị Phương Mai đã trực tiếp giúp đỡ, truyền đạt kiến thức quý báu và tận tình hướng dẫn của cô. Nếu không có sự hướng dẫn, chỉ bảo tận tình của cô chắc hẳn nhóm chúng em sẽ rất khó hoàn thành bài báo cáo theo đúng yêu cầu đặt ra.

Em cũng xin gửi cảm ơn các bạn, anh, chị trong trường Đại học Kỹ thuật –Hậu cần CAND đã giúp đỡ, động viên, chia sẻ kinh nghiệm của mình để nhóm chúng em có thể hoàn thành bài báo cáo này theo đúng tiến độ.

Mặc dù đã cố gắng thực hiện đề tài, xong nhóm chúng em cũng khó có thể tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm cùng với những góp ý, chỉ bảo của quý thầy cô để bài báo cáo có thể được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc511339215)

[MỤC LỤC ii](#_Toc511339216)

[DANH MỤC HÌNH iv](#_Toc511339217)

[DANH MỤC BẢNG iv](#_Toc511339218)

[DANH MỤC CÁC CỤM TỪ VIẾT TẮT vi](#_Toc511339219)

[MỞ ĐẦU 1](#_Toc511339220)

[Chương 1. CƠ SỞ LÝ THUYẾT 2](#_Toc511339221)

[1.1. Nguồn gốc và sự phát triển của lập luận logic 2](#_Toc511339222)

[1.2. Nguồn gốc cơ sở dữ liệu suy diễn và nhu cầu phát triển cơ sở dữ liệu suy diễn trong thực tế 3](#_Toc511339223)

[1.3. Khái niệm tri thức 4](#_Toc511339224)

[1.4. Biểu diễn tri thức bằng luật sinh 6](#_Toc511339225)

[1.4.1. Khái niệm 6](#_Toc511339226)

[1.4.2. Cơ chế suy luận dựa trên luật sinh 7](#_Toc511339227)

[1.4.3. Ưu nhược điểm của việc biểu diễn tri thức bằng luật 8](#_Toc511339228)

[1.5. Các phương pháp xử lý tri thức 9](#_Toc511339229)

[1.5.1. Xử lý tri thức bằng luật 9](#_Toc511339230)

[1.5.2. Xử lý tri thức bằng luật có biến 10](#_Toc511339231)

[1.5.3. Xử lý tri thức bằng lập luận 10](#_Toc511339232)

[1.6. Ngôn ngữ lập trình logic Prolog 11](#_Toc511339233)

[Chương 2. CƠ SỞ DỮ LIỆU SUY DIỄN VÀ NGÔN NGỮ DATALOG 13](#_Toc511339234)

[2.1. Cơ sở dữ liệu suy diễn 13](#_Toc511339235)

[2.1.1. Khái niệm 13](#_Toc511339236)

[2.1.2. Cấu trúc của một cơ sở dữ liệu suy diễn 16](#_Toc511339237)

[2.2. Chương trình Datalog 20](#_Toc511339238)

[2.2.1. Mô hình dữ liệu 20](#_Toc511339239)

[2.2.2. Cú pháp 20](#_Toc511339240)

[2.2.3. Ngữ nghĩa 21](#_Toc511339241)

[2.2.4. Cấu trúc cơ bản 21](#_Toc511339242)

[2.2.5. Truy vấn trong Datalog 22](#_Toc511339243)

[Chương 3. CÀI ĐẶT DEMO VÀ ỨNG DỤNG 27](#_Toc511339244)

[3.1 Công cụ thực hiện 27](#_Toc511339245)

[3.2. Đặt vấn đề bài toán 27](#_Toc511339246)

[3.3. Quy trình xử lý bài toán 28](#_Toc511339247)

[3.4. Demo chương trình 30](#_Toc511339248)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 35](#_Toc511339249)

[4.1. Kết quả đạt được 35](#_Toc511339250)

[4.2. Các mặt chưa làm được 35](#_Toc511339251)

[4.3. Phương hướng phát triển 35](#_Toc511339252)

[DANH MỤC TÀI LIỆU THAM KHẢO 37](#_Toc511339253)

# DANH MỤC HÌNH

[Hình 1.1. Sơ đồ AND/OR xác định lỗi thiết bị âm thanh. 8](#_Toc511373654)

[Hình 3.1. Các sự kiện của chương trình 28](#_Toc511373667)

[Hình 3.2. Các luật của chương trình 29](#_Toc511373668)

[Hình 3.3. Giao diện khởi động của chương trình 30](#_Toc511373669)

[Hình 3.4. Giao diện chính của chương trình 31](#_Toc511373670)

[Hình 3.5. Giao diện chính của chương trình 32](#_Toc511373671)

[Hình 3.6. Giao diện chính của chương trình 33](#_Toc511373672)

[Hình 3.7. Chương trình đưa ra kết luận 33](#_Toc511373673)

[Hình 3.8. Giao diện quản lý luật của chương trình 34](#_Toc511373674)

# DANH MỤC BẢNG

[Bảng 2.1. Kiểu dữ liệu trong DES. 18](#_Toc511373588)

[Bảng 2.2. Bảng quan hệ Quản lý công nhân 24](#_Toc511373589)

# DANH MỤC CÁC CỤM TỪ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Nội dung** | **Ghi chú** |
| 1 | CSDL | Cơ sở dữ liệu |  |
| 2 | DES | Datalog Education System |  |
| 3 | EDB | Extended Database | Cơ sở dữ liệu ngoại diện |
| 4 | IDB | Intentional Database | Cơ sở dữ liệu nội hàm |
| 5 | RBTV | Ràng buộc toàn vẹn |  |

# MỞ ĐẦU

Chúng ta đang sống trong sự phát triển bùng nổ của ngành công nghệ thông tin cùng với công cuộc cách mạng 4.0 đã và đang trở thành xu hướng phát triển của thế giới. Thì việc áp dụng công nghệ thông tin vào đời sống, thực tiễn là vô cùng quan trọng bởi lẽ nó sẽ trở thành cơ sở cho những ngành khác có thể ứng dụng vào để có được hiệu quả cao.

Việc áp dụng công nghệ thông tin vào đời sống hiện nay hầu hết đều phải sử dụng tới cơ sở dữ liệu(CSDL), nó là lĩnh vực không thể thiếu cho việc lưu trữ, xử lý khối lượng thông tin lớn chẳng hạn như trong việc xây dựng web, xây dựng các chương trình quản lý, xây dựng các ứng dụng, robot,... như hiện nay.

CSDL có nhiều mô hình khác nhau như: CSDL quan hệ, CSDL phân tán, CSDL hướng đối tượng,... trong đó CSDL quan hệ là được sử dụng phổ biến và đạt được thành công nhất từ trước tới giờ.

Tuy nhiên, với sự phát triển nhanh chóng và mạnh mẽ của công nghệ và sự bùng nổ của mạng Internet trong công cuộc cách mạng 4.0. đã đặt ra những yêu cầu cấp thiết là việc CSDL cần phải có khả năng suy luận từ các thông tin đã có trong CSDL. Tức là ngoài khả năng lưu trữ, tiếp nhận thông tin, xử lý thông tin thì cần phải có thể suy luận ra những thông tin mới từ thông tin đầu vào, nhằm tối ưu hóa cơ sở dữ liệu.

CSDL suy diễn kế thừa và mở rộng một số chức năng của CSDL quan hệ bởi lẽ ngoài những tính chất, nguyên tắc như CSDL quan hệ thì CSDL suy diễn còn có thêm các yếu tố như các luật suy diễn, các ràng buộc toàn vẹn (RBTV) mà người lập trình có thể tự tạo ra.

# Chương 1. CƠ SỞ LÝ THUYẾT

## 1.1. Nguồn gốc và sự phát triển của lập luận logic

Dựa theo truyền thống, logic được nghiên cứu như là một nhánh của triết học, nó còn được biết đến như là một lĩnh vực nghiên cứu về tiêu chí đánh giá về các luận cứ, lập luận. Tuy nhiên đến bây giờ, định nghĩa chính xác của logic vẫn là một vấn đề được các triết gia đang tranh cãi. Nhưng nhiệm vụ của logic học vẫn vô cùng quan trọng. Đó là, làm đẩy mạnh tiến bộ của việc phân tích các suy luận có hiệu lực và suy luận biện chứng để có thể phân biệt được luận cứ nào là hợp lý và không hợp lý.

Ngay ở giữa thế kỷ 19, logic đã được nghiên cứu trong toán học và luật. Gần đây nhất, logic còn được áp dụng vào công nghệ khoa học máy tính và đặc biệt là trí tuệ nhân tạo. Logic là một ngành khoa học hình thức, phân loại nghiên cứu cấu trúc của các khẳng định và các lý lẽ, cả hai đều thông qua việc nghiên cứu các hệ thống hình thức của việc suy luận và nghiên cứu lý lẽ trong ngôn ngữ tự nhiên. Tầm bao quát của logic rất rộng, đi từ các đề tài cốt lõi như là nghiên cứu các lý lẽ nghịch lý và ngụy biện, đến phân tích chuyên gia về các lập luận. Chẳng hạn như lập luận có tỷ lệ xác suất đúng và các lý lẽ liên quan đến quan hệ nhân quả, quan hệ rằng buộc trước sau. Ngày nay logic còn được sử dụng trong lý thuyết lý luận.

Một trong những tác phẩm logic sớm nhất còn tồn tại đến ngày nay là của Aristotle. Logic của Aristole được chấp nhận vô cùng rộng rãi trong khoa học và toán học ở Phương Tây đến đầu thế kỷ 19. Hệ thống logic của Aristotle phù hợp cho việc giới thiệu suy diễn giả định và logic quy nạp. Ở Châu Âu, trong cuối thời kỳ trung đại, có nhiều nỗ lực nhằm chứng tỏ những tư tưởng của Aristotle tương thích với niềm tin Cơ Đốc. Trong suốt thời kỳ Trung Kỳ Trung Cổ, logic trở thành đề tài chính của các triết gia trong những cuộc tranh luận triết học về phân tích logic học.

Suy luận hay lập luận được định nghĩa rất khác nhau tùy theo ngữ cảnh của hiểu biết về lý tính như là một hình thức của tri thức. Định nghĩa logic là hành động sử dụng lý tính để rút ra một kết luận từ các tiền đề nhất định bằng cách sử dụng một phương pháp luận cho trước.

Suy luận là các phương pháp thuộc về nhận thức tự phát. Nhận thức suy luận là kiểu nhận thức gián tiếp. Nhận thức một định lý toán học, một định luật khoa học ... đều là nhờ suy luận. Vì rằng, suy luận nhất thiết đòi hỏi phải có yếu tố trung gian, đó là những phán đoán, những khái niệm. Ví dụ để nhận biết sự tương đương giữa A và C khi A=B, B=C, thì phải sử dụng các khái niệm phương trình, sự tương đương, số hạng.

Suy luận có tính trừu tượng và tổng quát. Suy luận dựa vào khái niệm, phán đoán mà chúng ta phải thừa nhận. Ví dụ khái niệm "người" chỉ định tất cả mọi người và nói lên đặc tính chung của loài người, chứ không nói đến tính riêng biệt như tính chất thông thái vượt trội của Socrates hay dũng mãnh như Hercules...vv.

Tuy nhiên, trong các ngữ cảnh triết học lý tưởng, lập luận là quy trình trí óc đem lại cho ý nghĩ, tri giác, sự tưởng tượng, và cảm giác của ta bất cứ cái gì có thể hiểu được mà những hành vi trí óc kia có thể hàm chứa, và từ đó liên hệ trải nghiệm của ta với ý nghĩa toàn thể. Các phương pháp lập luận logic: suy luận diễn dịch, suy luận quy nạp, suy luận loại suy và phương pháp thứ tư là phép tương tự. Lập luận bằng phép tương tự có thể đi từ trường hợp cụ thể này tới trường hợp cụ thể khác nhưng kết luận của lập luận bằng phép tương tự chỉ là có một đáp án là đúng. Chúng ta gặp lập luận bằng tương tự rất thường xuyên trong nhận thức thông thường, khoa học, triết học và khoa học nhân văn.

Suy diễn logic, lập luận bằng suy diễn hay suy diễn là lập luận mà trong đó kết luận được rút ra từ các sự kiện được biết trước theo kiểu: nếu các tiền đề đúng thì kết luận phải đúng. Ngoài ra, suy diễn còn được định nghĩa là kiểu suy luận từ trường hợp tổng quát hơn tới trường hợp cụ thể hơn, hay là suy luận mà trong đó kết luận có độ chính xác tính ngang bằng với các tiền đề.

## 1.2. Nguồn gốc cơ sở dữ liệu suy diễn và nhu cầu phát triển cơ sở dữ liệu suy diễn trong thực tế

Xét dựa trên lý thuyết, CSDL suy diễn có thể coi như là các chương trình logic với sự khái quát hóa khái niệm về CSDL quan hệ. Đó là cách tiếp cận của Brodie và Manola vào năm 1989, của Codd vào năm 1970, của Date vào năm 1986, hay của Gardarin và Valdurier vào năm 1989 và của Ullman vào năm 1984.

CSDL suy diễn là sản phẩm tự nhiên của lập trình logic, trong đó logic toán được sử dụng cho các khái niệm tính toán của mô hình trực tiếp. Kỹ thuật CSDL suy diễn được ứng dụng nhiều trong các hệ hỗ trợ quyết định và các hệ chuyên gia. Cùng với các kỹ thuật phát triển của CSDL quan hệ, CSDL suy diễn cũng có khả năng lưu trữ một lượng lớn thông tin cũng như việc suy diễn trên các thông tin đó.

Các hệ chuyên gia, các hệ hỗ trợ quyết định...là những lĩnh vực có ứng dụng kỹ thuật CSDL suy diễn. Ngoài ra nó còn được ứng dụng rộng rãi trong việc phân tích tài chính, phân tích cú pháp và ngôn ngữ,... Nó cho phép phân tích một lượng lớn các dữ liệu đầu vào và đưa ra những suy luận, định hướng có độ chính xác nhất định cho tương lai. Trong CSDL suy diễn, mặc dù đã có nhiều kết quả có giá trị quan trọng nhưng cũng có nhiều vấn đề khó khăn cần nghiên cứu tiếp, chẳng hạn như vấn đề tối ưu hóa câu truy vấn và về cả vấn đề ngữ nghĩa.

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. Lập luận logic, cơ sở dữ liệu suy diễn là tiền đề quan trọng của lập trình logic và các ứng dụng của hiện tại. Tư tưởng cơ bản đằng sau lập trình logic và 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 logic Prolog (Programming logic).Nhờ sự hình thức hóa, Kowalski đã xem xét tập con của các logic bậc một, gọi là logic mệnh đề Horn. Một mệnh đề hay một câu theo logic có thể có nhiều điều kiện đúng nhưng chỉ có một hay không có kết luận đúng.

Một khía cạnh khác nhau giữa CSDL suy diễn và lập trình logic là hệ thống lập trình logic nhấn mạnh các chức năng, trong khi CSDL suy diễn nhấn mạnh tính hiệu quả. Cơ chế suy diễn dùng trong CSDL suy diễn để tính toán trả lời không được tổng quát như trong hệ thống lập trình logic.

Với hệ quản trị CSDL quan hệ dùng đại số quan hệ hiện tại thì việc đặt câu hỏi và trả lời câu hỏi còn hạn chế nhất là CSDL có kích thước lớn. Hơn nữa, vấn đề đệ quy thì trong đại số quan hệ không giải quyết được. CSDL suy diễn gồm các sự kiện và các luật sẽ cho chúng ta các câu trả lời từ câu hỏi, và cung cấp cho chúng ta những tri thức quan trọng mới.

Việc nghiên cứu CSDL được thực hiện trên thế giới từ những năm 80 của thế kỹ XX, nhưng ở nước ta hiện nay vẫn còn rất ít những công trình nghiên cứu về nó. Tuy nhiên, việc tìm hiểu, nghiên cứu CSDL suy diễn, những ứng dụng của nó là một vấn đề vô cùng quan trọng, có ý nghĩa lý thuyết và thực tiễn.

## 1.3. Khái niệm tri thức

Tri thức hay kiến thức bao gồm những dữ kiện, thông tin, sự mô tả, hay kỹ năng có được nhờ trải nghiệm hay thông qua giáo dục.Tri thức có thể chỉ sự hiểu biết về một đối tượng, về mặt lý thuyết hay thực hành. Nó có thể ẩn tàng, chẳng hạn những kỹ năng hay năng lực thực hành, hay tường minh, như những hiểu biết lý thuyết về một đối tượng; nó có thể ít nhiều mang tính hình thức hay có tính hệ thống. Mặc dù có nhiều lý thuyết về tri thức, nhưng hiện không có một định nghĩa nào về tri thức được tất cả mọi người chấp nhận.

Sự thành tựu tri thức liên quan đến những quá trình nhận thức phức tạp: tri giác - quá trình thu thập, giải nghĩa, lựa chọn và tổ chức các thông tin từ giác quan, truyền đạt,tiếp thu,giao tiếp, liên hệ, và suy luận.

Tri thức có 2 dạng tồn tại chính là tri thức ẩn và tri thức hiện:

- Tri thức hiện là những tri thức được giải thích và mã hóa dưới dạng văn bản, tài liệu, âm thanh, phim, ảnh,… thông qua ngôn ngữ có lời hoặc không lời, nguyên tắc hệ thống, chương trình máy tính, chuẩn mực hay các phương tiện khác. Đây là những tri thức đã được thể hiện ra ngoài và dễ dàng chuyển giao, thường được tiếp nhận qua hệ thống giáo dục và đào tạo chính quy.

- Tri thức ẩn là những tri thức thu được từ sự trải nghiệm thực tế, dạng tri thức này thường ẩn trong mỗi cá nhân và rất khó "mã hóa" và chuyển giao, thường bao gồm: niềm tin, giá trị, kinh nghiệm, bí quyết, kỹ năng... VD: Trong bóng đá, các cầu thủ chuyên nghiệp có khả năng cảm nhận bóng rất tốt. Đây là một dạng tri thức ẩn, nó nằm trong mỗi cầu thủ. Nó không thể "mã hóa" thành văn bản, không thể chuyển giao, mà người ta chỉ có thể có bằng cách tự mình luyện tập.

Dựa vào sự phân loại tri thức, có thể chia các hình thức chia sẻ tri thức thành bốn dạng chính:

- Ẩn - Ẩn: Khi người chia sẻ và người tiếp nhận giao tiếp trực tiếp với nhau (ví dụ: học nghề, giao tiếp, giảng bài...) thì việc tiếp nhận này là từ tri thức ẩn thành tri thức ẩn. Tri thức từ người này không qua trung gian mà chuyển ngay thành tri thức của người kia.

- Ẩn - Hiện: Một người mã hóa tri thức của mình ra thành văn bản hay các hình thức hiện hữu khác thì đó lại là quá trình tri thức từ ẩn (trong đầu người đó) trở thành hiện (văn bản, tài liệu, v.v.).

- Hiện - Hiện: Tập hợp các tri thức hiện đã có để tạo ra tri thức hiện khác. Quá trình này được thể hiện qua việc sao lưu, chuyển giao hay tổng hợp dữ liệu.

- Hiện - Ẩn: Tri thức từ dạng hiện trở thành dạng ẩn. Điển hình quá trình này là việc đọc sách. Học sinh đọc sách (tri thức hiện) và rút ra được các bài học, tri thức cho mình (ẩn).

* Phân loại tri thức

*Tri thức thủ tục*: Loại tri thức này đưa ra các giải pháp để thực hiện một công việc nào đó, hay đưa ra cách thức để giải quyết một vấn đề. Loại tri thức này được thể hiện như các luật, chiến lược, lịch trình và thủ tục.

*Tri thức khai báo (mô tả)*: cho biết một vấn đề được thấy như thế nào. Loại tri thức này bao gồm các phát biểu đơn giản, dưới dạng khẳng định logic đúng hoặc sai.

*Tri thức heuristic (tri thức may rủi)*: mô tả các "mẹo" để dẫn dắt tiến trình lập luận. Tri thức này còn được gọi là tri thức nông cạn do không bảo đảm hoàn toàn chính xác về kết quả giải quyết vấn đề.

*Siêu tri thức*: mô tả tri thức về tri thức. Loại tri thức này giúp lựa chọn tri thức thích hợp nhất trong số các tri thức khi giải quyết một vấn đề.

*Tri thức cấu trúc*: Mô tả cấu trúc, mô hình của tri thức về một vấn đề nào đó.

Ví dụ: các tập luật, các quan hệ khái niệm, khái niệm về quan hệ của các đối tượng.

## 1.4. Biểu diễn tri thức bằng luật sinh

### 1.4.1. Khái niệm

Tri thức được biểu diễn dưới dạng những câu trong ngôn ngữ máy, chúng được coi nhưng là sự mã hóa về tri thức hiện thực của một vấn đề nào đó mà người ta đưa vào ngôn ngữ máy. Ngôn ngữ biểu diễn tri thức gồm hai thành phần cơ bản là cú pháp và ngữ nghĩa.

- Cú pháp của một ngôn ngữ bao gồm các ký hiệu tuân thủ theo các quy tắc liên kết (các luật cú pháp) để tạo thành các câu (công thức) trong ngôn ngữ.

- Ngữ nghĩa của ngôn ngữ cho phép ta xác định ý nghĩa của các câu trong một miền nào đó của thế giới hiện thực.

Ngoài hai thành phần cú pháp và ngữ nghĩa, ngôn ngữ biểu diễn tri thức cần được cung cấp cơ chế suy luận. Một suy luận cho phép ta suy ra một công thức từ một tập nào đó các công thức. Như vậy người ta xác định:

Ngôn ngữ biểu diễn tri thức = Cú pháp + Ngữ nghĩa + Cơ chế suy luận.

Ý tưởng cơ bản của biểu diễn tri thức bằng luật sinh là tri thức có thể được cấu trúc bằng một cặp điều kiện – hành động : "NẾU điều kiện xảy ra THÌ hành động sẽ được thi hành". Ví dụ : NẾU trời mưa THÌ đường sẽ ướt, NẾU đèn không sáng THÌ hãy kiểm tra nguồn điện, …

Ngày nay, việc sử dụng luật sinh ngày càng được sử dụng một các thông dụng và rộng rãi trong các hệ chuyên gia, trí tuệ nhân tạo. Luật sinh có thể đưa ra các mô tả để giải quyết các vấn đề thực tế từ những thông tin nhận được thay cho các kiểu phân tích vấn đề truyền thống. Trong trường hợp này,các luật tuy còn có thể chưa hoàn chỉnh nhưng hỗ trợ việc tìm kiếm là rất hữu ích.

Một cách tổng quát luật sinh có dạng như sau : P1ʌP2 ʌ...Pn → Q

Tùy vào các vấn đề đang quan tâm mà luật sinh có những ngữ nghĩa hay cấu tạo khác nhau:

Trong logic vị từ : P1, P2, ..., Pn, Q là những biểu thức logic.

Trong ngôn ngữ lập trình, mỗi một luật sinh là một câu lệnh.

IF (P1 AND P2 AND .. AND Pn) THEN Q.

Để biễu diễn một tập luật sinh, người ta thường phải chỉ rõ hai thành phần chính sau:

1. Tập các sự kiện F(Facts)

F={f1,f2,...fn}

1. Tập các quy tắc R(Rules) áp dụng trên các sự kiện dạng như sau:

f1ʌ f2,.. ʌ fn→q

Trong đó, các fi, q đều thuộc F

Ví dụ : Cho 1 cơ sở tri thức được xác định như sau :

Các sự kiện : A, B, C, D, E, F, G, H, K

Tập các quy tắc hay luật sinh (rule):

R1 : A→ K

R2 : B →C

R3 : H→G

R4 : C ʌ G→ E

R5 : E ʌ B→ D

R6 : G ʌ K ʌ A→F

### 1.4.2. Cơ chế suy luận dựa trên luật sinh

*Suy diễn tiến*: Là quá trình suy luận xuất phát từ một số sự kiện ban đầu, xác định các sự kiện có thể được "sinh" ra từ sự kiện này. Suy diễn tiến thường được sử dụng cho các hệ dự báo như chương trình dự báo thời tiết, dự báo kinh tế, chính trị,...

Sự kiện ban đầu : B,H

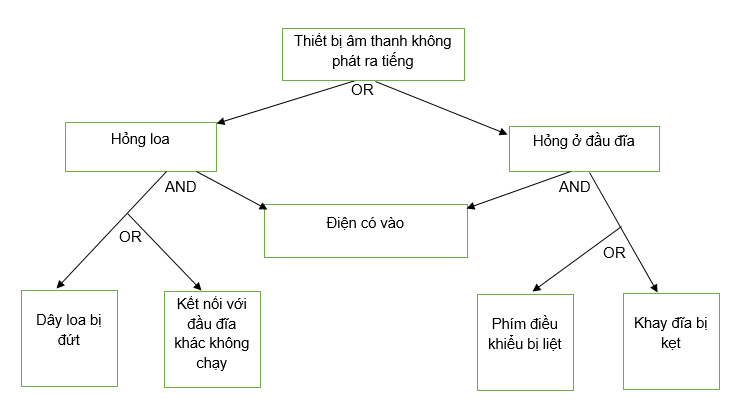
R3 : H→G{G, H, B}

R2 : B →C{C, G, H, B}

R4 : CG→E{E, B, C, H, G}

*Suy diễn lùi*: Là quá trình suy luận ngược xuất phát từ một số sự kiện ban đầu, ta tìm kiếm các sự kiện đã "sinh" ra sự kiện này. Ví dụ cho suy diễn lùi, như trong việc chuẩn đoán bệnh của bác sĩ, từ các tập sự kiện là các triệu chứng của bệnh nhân, các luật liên quan đến các sự kiện ấy từ đó bác sĩ có thể đưa ra việc chuẩn đoán bệnh của mình.

Ví dụ: Việc chuẩn đoán thiết bị âm thanh không phát ra tiếng là do đâu?



Hình 1.1. Sơ đồ AND/OR xác định lỗi thiết bị âm thanh.

### 1.4.3. Ưu nhược điểm của việc biểu diễn tri thức bằng luật

#### 1.4.3.1. Ưu điểm

Khi cần đưa ra một hành động nào đó từ những sự kiện có thể quan sát được, thì việc sử dụng các luật là rất có ích cho việc nhận các thông tin trực tiếp rồi từ các luật đó để đưa ra quyết định hoạt động. Nó có những ưu điểm chính sau đây:

* Vì các luật gần với ngôn ngữ tự nhiên nên dễ dàng cho việc giao tiếp với người dùng
* Dễ dàng cho việc cấu hình, suy luận,giải thích
* Có thể dễ dàng hiệu chỉnh và bảo trì hệ thống
* Có thể cải tiến dễ dàng để tích hợp các luật mờ.
* Các luật thường ít phụ thuộc vào nhau.

#### 1.4.3.2. Nhược điểm

- Luật sinh có thể được dùng quá nhiều trong khi diễn tả các tri thức phức tạp. Điều này gây khó khăn cho việc quản trị cũng như ảnh hưởng tới tốc độ xử lý.

- Luật sinh được người xây dựng trí tuệ nhân tạo sử dụng nhiều vì đơn giản, dễ cài đặt tuy nhiên nhiều trường hợp có phương pháp tốt hơn,hiệu quả hơn lại không được sử dụng.

- Cơ sở tri thức luật sinh lớn làm cho việc áp dụng các luật sinh, đánh giá các hệ để suy luận trở nên khó khăn, điều đó có thể làm cho hiệu quả của phần mềm không cao.

## 1.5. Các phương pháp xử lý tri thức

Thông qua phần trên chúng ta có thể kết luận tri thức, sự hiểu biết là nền tảng của trí tuệ, sự thông minh. Nhưng bên cạnh đó, để trí tuệ, sự thông minh thật sự tối ưu thì còn phải phụ thuộc vào việc chúng ta vận dụng, xử lý tri thức.

Và để xử lý được các tri thức, thông tin đầu vào thì trí tuệ nhân tạo cần phải có kỹ thuật để mã hóa thông tin đó. Các kĩ thuật mã hóa mà chúng ta nói ở đây không phải là các cách các dữ liệu trong các phương trình, thuật toán xử lý dữ liệu thường gặp mà là mã hóa các thông tin nền tảng mà hệ thống quan tâm dưới dạng một dữ liệu nào đó. Ví dụ khi chúng ta cung cấp thông tin một người là 65 tuổi thì sẽ xác định người đó thuộc đối tượng người cào tuổi, từ đó chương trình sẽ xác định được người đó có nguy cơ mắc những bệnh gì. Theo như ví dụ trên thì khoảng tuổi chúng ta cung cấp vào chính là mã hóa cho việc chung ta cung cấp thông tin người đó thuộc đối tượng người cáo tuổi hay chưa.

Một trong những biện pháp xử lý tri thức phổ biến hiện nay là sử dụng luật. Luật vừa có thể được sử dụng để biểu diễn tri thức vừa có thể được dùng để xử lý tri thức.

### 1.5.1. Xử lý tri thức bằng luật

Như đã đề cập ở phần trên, luật thuộc dạng tri thức thủ tục. Nó gắn các sự kiện mà chúng ta cung cấp với các hoạt động cụ thể. Các hoạt động này sẽ giúp chúng ta tìm ra các thông tin mới hay là đưa ra các hành vi sẽ được thực hiện. Vì thế, luật mô tả cách giải quyết vấn đề.

Từ việc xác định nguyên tắc hoạt động của luật, chúng ta có thể hiểu quá trình xử lý tri thức là quá trình gắn các thông tin được cung cấp với các thông tin khác để đưa ra kết luận. Ví dụ như: chúng ta cung cấp thông tin “5 là một số nguyên tố lớn hơn 2” và trong hệ thống có luật “Các số nguyên tố lớn hơn 2 thì không chia hết cho 2”. Vậy sau khi liên kết thông tin đã cho và thông tin “Các số nguyên tố lớn hơn 2 thì không chia hết cho 2” thì hệ thống sẽ đưa ra kết luận là số 5 không chia hết cho 2.

Quá trình xử lý tri thức trải qua các bước sau:

* Thêm thông tin vào bộ nhớ tri thức cơ sở của hệ thống.
* Sử dụng các luật cùng với thông tin trong bộ nhớ để giải bài toán và đưa kết luận có được vào bộ nhớ. Kết quả thu được cũng có thể làm luật mới được thực hiện (Kết quả ở đây không nhất thiết phải là thông tin mà cũng có thể là một hành động đã được thực thi).

Để thực hiện các thao tác phức tạp, hệ thống có thể truy nhập vào các thủ tục bên ngoài. Ví dụ như: IF cần biết tên của một học viên trong lớp B1D5 OPEN Bảng HOCVIEN AND Tên lớp = B1D5.

Như đã đề cập ở phần trên, luật thuộc dạng tri thức thủ tục. Nó gắn các sự kiện mà chúng ta cung cấp với các hoạt động cụ thể. Các hoạt động này sẽ giúp chúng ta tìm ra các thông tin mới hay là đưa ra các hành vi sẽ được thực hiện. Vì thế, luật mô tả cách giải quyết vấn đề.

### 1.5.2. Xử lý tri thức bằng luật có biến

Để tăng hiệu quả xử lý cho các luật, người ta sử dụng các biến trong luật để thực hiện các thao tác trên cùng một tập đối tượng tương tự nhau.

Ví dụ 1: Luật sau cho biết A có bị rớt môn Thực tập chuyên đề: Nếu điểm trung bình môn Thực tập chuyên đề của A là 4,9 thì A rớt môn Thực tập chuyên đề.

Để xét điều kiện rớt các môn khác cho các học sinh khác thì hệ thống phải viết ra rất nhiều luật. Tuy nhiên người ta có thể dùng các biến để làm cho luật khớp với dãy giá trị rộng hơn.

Ví dụ 2: Nếu điểm trung bình môn X của Y dưới 5 thì Y rớt môn X.

### 1.5.3. Xử lý tri thức bằng lập luận

Một phương pháp thông dụng khác dùng để xử lý tri thức là dùng lập luận.

Lập luận là sử dụng lý tính để đưa ra kết luận thông qua các phương pháp luận cho trước. Hai phương pháp lập luận thông dụng nhất là lập luận suy diễn và lập luận quy nạp.

Một số phương pháp lập luận:

**Lập luận suy diễn:** là phương pháp lập luận rút ra thông tin mới dựa trên những thông tin đã biết và những thông tin này có quan hệ logic với nhau. Lập luận suy diễn sử dụng các tiền đề (sự kiện) và các kéo theo (luật) để giải bài toán.

Ví dụ 1: Có tiền đề A đúng, kéo theo A 🡪 B thì kết luận được B đúng.

**Lập luận quy nạp:** Là phương pháp đưa ra kết luận tổng quát từ một tập các sự kiện cụ thể.

Ví dụ 2: B thuộc dòng họ A, C là con B nếu chứng minh được C ũng thuộc dòng họ A thì các trường hợp D, E, F là con cháu của C thì sẽ thuộc dòng họ A.

**Lập luận tương tự, loại suy:** Là phương pháp lập luận người ta sẽ tạo ra mô hình về một số khái niệm.Sau đó người ta sẽ tìm ra các điểm tương tự và khác biệt của các đối tượng cụ thể so với mô hình để hướng dẫn việc lập luận của họ. Nếu là các điểm tương tự thì những đối tượng này sẽ có các tính chất mà khái niệm đó có. Còn nếu là điểm khác biệt thì sẽ là cơ sở dung để hiểu biết các khai niệm mới.

Thông qua việc tìm hiểu các phương pháp lập luận trên, ta thấy hầu hết việc lập luận (suy diễn, quy nạp) đều phải sử dụng một công cụ là luật. Lập luận dựa trên luật có hai kiểu cơ bản là: *lập luận tiến* và *lập luận lùi*.

Quá trình lập luận tiến: Đầu tiên ta sẽ so sánh các sự kiện trong cơ sở sự kiện với mỗi điều kiện trong luật, nếu tất các điều kiện được thỏa mãn thì suy ra được sự kiện trong phần kết luận. Nếu sự kiện này là sự kiện mới thì sẽ được thêm vào bộ nhớ. Ta lặp lại công việc trên cho đến khi không còn laut65 nào sinh ra được sự kiện mới.

Quá trình lập luận lùi: Lập luận lùi là việc ta chứng minh hay bác bỏ một giả thiết nào đó. Vì thế quá trình lập luận lùi là ta sẽ so sánh giả thiết với các sự kiện trong bộ nhớ. Nếu có sự kiện khớp với giả thiết thì giả thiết đó được chứng minh, còn không thì ta sẽ so sánh giả thiết của với phần kết luận của các luật. Nếu không có phần kết luận nào trùng với giả thiết thì coi như giả thuyết bị bác bỏ, còn nếu trùng thì ta lại coi phần điều kiện của các luật đó như giả thiết mới và lặp lại công việc trên cho đến khi luật dược chứng minh hay bác bỏ.

## 1.6. Ngôn ngữ lập trình logic Prolog

Trước khi tìm hiểu về Prolog, chúng ta hãy đi tìm hiểu lập trình logic là gì.

Lập trình logic là mẫu hình lập trình dựa trên logic toán. Lập trình logic là lập trình khai báo, sử dụng ngôn ngữ mô tả để đặc tả các vấn đề. Một chương trình được viết bằng ngôn ngữ lập trình logic là tập hợp các mệnh đề, mỗi mệnh đề được xây dựng từ nhiều vị từ, mỗi vị từ là phát biểu về một đối tượng có thể đúng hoặc sai. Một mệnh đề là một trong hai hình thức sau:

* Sự kiện: khẳng định một thực thể có 1 hoặc vài tính chất.

Ví dụ: Cường, Đông, Quyền là học viên.

* Luật: định nghĩa các quan hệ dựa trên các quan hệ.

Ví dụ: Nếu N không âm và N chia hết cho 2 thì N là số chẵn.

Prolog là ngôn ngữ lập trình phổ biến nhất hiện nay.

Prolog là ngôn ngữ mô tả, với những sự kiện và luật đã được mô tả, Prolog sẽ suy luận cho ta kết quả. Nguyên lý hoạt động của Prolog dựa trên các mệnh đề Horn.

Ví dụ:

1. Cường là học viên trường T36.
2. Nếu ai là học viên trường T36 phải chấp hành điều lệnh CAND.
3. A là con B và B là con C thì A là cháu C.
4. Jonh là con Jim.

Nếu ta đặt câu hỏi “Cường phải chấp hành điều lệnh CAND là đúng hay sai ?” đến CSDL này thì hệ thống sẽ thực hiện suy luận - tìm kiếm câu trả lời dựa trên vốn hiểu biết đã có của chương trình – CSDL để chứng minh câu hỏi trên là một khẳng định đúng hay sai. Ví dụ với câu hỏi trên thì hệ thống tìm thấy mệnh đề 2 thỏa mãn vế thì nên lập luận lùi, đưa “Cường là học viên T36” làm giả thiết mới và hệ thống tìm thấy sự kiện Cường là học viên T36 nên khẳng định Cường phải chấp hành điều lệnh CAND.

Prolog là ngôn ngữ sử dụng các ký hiệu để diễn tả các tính chất của đối tượng và các quan hệ giữa các đối tượng.

Qui ước:

* Sự kiện:
* P(A): A có tính chất P. Ví dụ: hocvien(cuong,dong,quyen).
* P(A,B): A la P đối với B. Ví dụ: giaovien(mai,cuong).
* P(A1,A2,...,An) :P là tính chất, A1,...., An là các đối.
* Luật:
* “:-“ : nếu trong prolog
* Vế trái chỉ kết luận
* Vế phải chỉ điều kiện, nếu có nhiều điều kiện thì cách nhau bằng dấu phẩy.

Ví dụ: Ong(A,C):- Ba(A,B), Ba(B,C).

# Chương 2. CƠ SỞ DỮ LIỆU SUY DIỄN VÀ NGÔN NGỮ DATALOG

## 2.1. Cơ sở dữ liệu suy diễn

### 2.1.1. Khái niệm

CSDL suy diễn là CSDL có khả năng suy diễn ra một số sự kiện mới từ những sự kiện đã có trước đó. Bên cạnh đó, CSDL suy diễn là ngôn ngữ lập trình datalog mà ta sẽ trình bày ở phần bên dưới.

CSDL suy diễn có mối quan hệ với lập trình logic nhưng giữa chúng vẫn có sự khác biệt. Nếu lập trình logic chú trọng về các chức năng thì CSDL suy diễn lại thiên về tính hiệu quả. Cơ chế suy diễn để tìm kiếm câu trả lời trong CSDL suy diễn cũng không tổng quát như trong lập trình logic.

Datalog là ngôn ngữ phi thủ tục dựa trên logic vị từ bậc nhất được sử dụng để mô tả những thông tin cần thiết được lấy bằng cách dựa trên logic chứ không theo như cách trong các thủ tục thông thường.

#### 2.1.1.1. Một số khái niệm về CSDL suy diễn

#### a. Ngôn ngữ cấp một

Một ngôn ngữ cấp một là ngôn ngữ được xây dựng trên một bộ ký tự và những công thức xây dựng trên bộ ký tự đó.

Một ngôn ngữ cấp một bao gồm:

* Bộ ký tự, gồm các thành phần sau:
* Các ký hiệu hằng, ký hiệu biến.
* Các ký hiệu hàm.
* Các ký hiệu vị từ.
* Các hằng vị từ: true,false.
* Các ký hiệu kết nối: ¬, ˄, ˅, →, ↔.
* Các ký hiệu lượng tử: C:\Users\nvcqu\Desktop\với mọi.PNG, .
* Cặp dấu ngoặc đơn (), dấu phẩy (,).
* Hạng thức:

Được định nghĩa đệ quy như sau:

* Mỗi hằng và biến là một hạng thức.
* Nếu f là hàm n ngôi và a1,…,an là các hạng thức thì f(a1,…,an) là một hạng thức.
* Các hạng thức chỉ được sinh ra bằng các quy tắc trên.
* Nguyên tố:
* Nếu p là một vị từ n ngôi và a1,…,an là các hạng thức thì p(a1,…,an) là một nguyên tố.
* Literal là một nguyên tố hoặc phủ định của một nguyên tố, trong đó:
* Literal dương là một nguyên tố.
* Literal âm là một nguyên tố. Ký hiệu: ¬p
* Công thức:

Được định nghĩa đệ qui như sau:

* Nguyên tố là một công thức.
* True và false là các công thức.
* Nếu E và F là các công thức thì: (E ˄ F), ¬E, (E ˅ F), (E → F), (E ↔ F) là các công thức.
* Nếu E là công thức, X là biến thì (C:\Users\nvcqu\Desktop\với mọi.PNGX) E, (X) E là các công thức.
* Công thức chỉ được sinh ra bởi một số hữu hạn các quy tắc trên.

#### b. Chương trình logic

Chương trình logic là một tập hợp hữu hạn khác rỗng các mệnh đề chương trình, tức là chỉ chứa hai loại mệnh đề:

A←

A←B1˄B2˄…˄Bm

Vậy mệnh đề chương trình là gì?

Trước khi tìm hiểu mệnh đề chương trình là gì, chúng ta hãy cùng nhìn qua khái niệm của mệnh đề.

Mệnh đề là một công thức có dạng:

C:\Users\nvcqu\Desktop\với mọi.PNGX1…C:\Users\nvcqu\Desktop\với mọi.PNGXs(L1˅…˅Ln)

Trong đó L1,…,Ln là các literal, X1,…,Xs là các biến xuất hiện trong L1˅…˅Ln.

Ta chia các Literal thành các Literal dương A1,…,Ak và các Literal âm ¬B1,…, ¬Bm ta có:

A1˅…˅Ak˅¬B1˅…˅¬Bm

* A1˅…˅Ak˅¬(B1˄…˄Bm)
* A1˅…˅Ak←B1˄…˄Bm

Đây chính là mệnh đề kéo theo có phần hệ quả là A1˅…˅Ak và phần B1˄…˄Bm là tiền đề.

Vậy, mệnh đề chương trình là mệnh đề chỉ có một hệ quả, có dạng:

A←B1˄…˄Bm

Ta xét:

* m = 0:

Ta có: A← , tức với mọi điều kiện thì A luôn đúng, đây là một mệnh đề đơn vị hay một unit.

* m ≠ 0:

Ta có: A←B1˄…˄Bm, tức là phần kết luận A sẽ đúng nếu các điều kiện đúng, đây là một quy tắc.

#### 2.1.1.2. Khái niệm tổng quan

CSDL suy diễn là CSDL có khả năng suy diễn ra một số sự kiện mới từ những sự kiện được lưu trong CSDL. CSDL suy diễn gồm 2 thành phần:

* CSDL ngoại diên: là một CSDL quan hệ tiêu chuẩn, có lược đồ gồm một tập các lược đồ quan hệ.
* CSDL nội hàm: Là CSDL được xác định bằng các lược đồ quan hệ và một chương trình Datalog định nghĩa các quan hệ đó.

VD: Ta có hai lược đồ quan hệ Mother(A,B) và Father(A,B) tạo thành CSDL ngoại diên. Ta có thể xác định quan hệ Parents(A,B) như sau:

Parents(A,B):-

Mother(A,B) ;

Father(A,B).

Đây là CSDL nội hàm.

#### 2.1.1.3. Khái niệm trên phương diện toán học

Một CSDL suy diễn là một tập hữu hạn các mệnh đề có dạng:

A1˅…˅Am←B1˄…˄Bn (m ≥ 0, n ≥ 0) (1)

Trong đó Ai (i = 0,...,m) là các nguyên tố và Bj (j = 0,...,n) là các literal. Tất cả các Ai và Aj đều không chứa các ký hiệu hàm.

Một số trường hợp đặc biệt của (1):

* Khi m = 1, n ≥ 1 thì mệnh đề có dạng là một quy tắc: A←B1˄…˄Bn
* Khi m > 1, n = 0 thì mệnh đề có dạng là một mệnh đề đơn: A1˅…˅Am←
* Khi mọi vị từ Ai là hằng thì nó được gọi là sự kiện tuyển.
* Khi m > 1, n ≥ 1 thì mệnh đề có dạng là một quy tắc Datalog dạng tuyển:

A1˅…˅Am←B1˄…˄Bn

* Khi CSDL suy diễn gồm các quy tắc xác định có dạng:

A←B1˄…˄Bn với n ≥ 0 thì nó là một CSDL suy diễn xác định.

* Khi CSDL suy diễn không xác định cho phép chứa các quy tắc có dạng:

A1˅…˅Am←B1˄…˄Bn với m > 1, n ≥ 0

* Nếu CSDL suy diễn có các Bi là các Literal không âm thì đó là một chương trình Datalog dạng tuyển,ngược lại là Datalog dạng tuyển mở rộng.

### 2.1.2. Cấu trúc của một cơ sở dữ liệu suy diễn

#### 2.1.2.1. Tập các sự kiện

Trong cơ sở dữ liệu suy diễn, sự kiện là vị từ cho phép biểu diễn những thông tin được cho là đúng.

Ví dụ 1: “Số 1 là số lẻ” được biểu diễn dưới dạng vị từ như sau: so\_le(1).

Từ các CSDL được thiết kế ban đầu thì tập các sự kiện dựa vào đó để mô tả đặc điểm, thuộc tính, mối quan hệ,… của các đối tượng trong CSDL.

Ví dụ 2:

Dùng các vị từ sự kiện để mô tả sách:

Gia(ten)// *mô tả giá sách*

Tac\_gia(ten,ten\_tg)// *mô tả tên tác giả*

Nha\_xuat\_ban(ten,nxb)// *mô tả nhà xuất bản*

The\_loai(ten,the\_loai)// *mô tả đặc điểm thể loại*

#### 2.1.2.2. Tập các luật suy diễn

Luật suy diễn cũng là các vị từ diễn tả quy luật suy diễn mà ta công nhận chúng. Luật được trình bày dưới dạng một mệnh đề. Nó cho phép suy diễn ra những sự kiện mới từ các sự kiện được lưu trữ trong CSDL.

Cấu trúc chính của một luật suy diễn gồm có 2 phần: phần đầu và phần thân.

Ví dụ:

Các sự kiện ban đầu :

Gia(ten,gia)// *mô tả giá sách*

Tac\_gia(ten,ten\_tg)// *mô tả tên tác giả*

The\_loai(ten,the\_loai)// *mô tả đặc điểm thể loại*

Luật suy diễn như sau:

mua(nam,X):-gia(X,re),tac\_gia(X,Nguyen\_Van\_A),the\_loai(X,truyen\_tranh).//

Có nghĩa là nam sẽ mua sách X với điều kiện X phải có những đặc điểm như:giá rẻ, tác giả Nguyễn Văn A, thể loại truyện tranh.

Tùy theo chương trình đơn giản hay phức tạp mà người ta sẽ sử dụng luật là nhiều hay ít, đơn giản hay phức tạp.

#### 2.1.2.3. Ràng buộc toàn vẹn

RBTV được sử dụng để lấy ra các giá trị hợp lệ theo ý người lập trình trong các quan hệ . Ngoài có những ràng buộc cơ bản như trong cơ sở dữ liệu SQL như: kiểu, khóa chính, khóa ngoại, thì cơ sở dữ liệu suy diễn còn cung cấp cho người dùng một số ràng buộc về chức năng đã được xác định.

Người lập trình có thể tự tạo ra các ràng buộc theo ý mình, điều này rất thuận tiện cho người lập trình, gọi là RBTV do người lập trình tự định nghĩa. Tất cả những ràng buộc này được hệ thống giám sát và thực hiện. Tuy nhiên hệ quản trị CSDL suy diễn cho phép vô hiệu hóa việc kiểm tra các ràng buộc.

Tất cả các ràng buộc xác định ở trên được áp dụng cho các sự kiện, nhưng ràng buộc về kiểu cũng được áp dụng cho các luật. Ngoài ra, các ràng buộc do người dùng tự định nghĩa được áp dụng cho cả sự kiện và luật.

#### a. Ràng buộc về kiểu (type).

Một ràng buộc kiểu xác định các giá trị trong miền của một đối số vị từ ( bảng cột trong thuật ngữ quan hệ) cho phép.

Ví dụ:

DES> :- type(p,[int,string])//Khai báo RBTV về kiểu trong Datalog.

Hoặc:

DES> :- type(p(int,string))

Các kiểu cho phép trong Datalog bao gồm ( mỗi dòng trong cột đầu tiên chứa các kiểu đồng nghĩa).

|  |  |
| --- | --- |
| int,integer | Kiểu số nguyên |
| float, real | Kiểu số thực |
| char | Một ký tự |
| char(N), varchar(N) | Chuỗi có tối đa N ký tự |
| varchar, string | Chuỗi dài vô hạn |

Bảng 2.1. Kiểu dữ liệu trong DES.

#### b. Ràng buộc khóa chính (Primary key).

Ràng buộc khóa chính sẽ ràng buộc trong một cột, một trường không thể có các giá trị trùng nhau.

Xét ví dụ sau:

Ví dụ 1:

DES> :-type(p,[a:int,b:string])

DES> :-pk(p,[a]) //Khai báo RBTV về khóa chính trong Datalog.

DES> :-pk(p,[c])

Error: Unknown column c.//lỗi do quan hệ p không có cột c.

Ví dụ 2:

DES> :-type(q,[a:int,b:int])

DES> /assert q(1,1)

DES> /assert q(2,2)

DES> /assert q(1,2)

DES> :-pk(q,[a])

Error: Primary key violation q.[a]// vi phạm ràng buộc khóa chính.

Offending values in database: [pk(1)]

Info: Constraint has not been asserted.

#### c. Ràng buộc khóa ngoại (foreign key).

Là ràng buộc mà giá trị của thuộc tính trong một bộ của quan hệ, một bảng phải phụ thuộc vào sự tồn tại của một bộ trong quan hệ khác.Thì tập thuộc tính phụ thuộc sẽ là khóa ngoại.

Ví dụ 1:

DES> :-type(p(a:int)),type(q(b:int)),pk(q,[b])

DES> :-fk(p,[a],q,[b])

Khi một quan hệ không tồn tại sẽ xảy ra lỗi.

DES> :-fk(p,[a],q,[b])

Error: Relation p has not been typed yet.

DES> :-type(p,[a:int]), type(q,[b:int])

Khi tạo khóa ngoại liên quan đến một quan hệ mà quan hệ đó không có khóa chính cũng sẽ xảy ra lỗi.

Ví dụ 2:

DES> :-fk(p,[a],q,[b])

Error: Referenced column list q.[b] is not a primary key.

DES> :-pk(q,[b])

DES> :-fk(p,[a],q,[b])

#### d. Ràng buộc do người dùng tự định nghĩa (User-defined Integrity Constraints).

Người dùng cũng có thể tự định nghĩa các RBTV, ràng buộc này là một luật không có phần đầu (head), phần thân của luật là một khẳng định ràng buộc dữ liệu theo yêu cầu. Trong quá trình thực thi nếu ràng buộc dữ liệu bị vi phạm sẽ thông báo cho người sử dụng.

Ví dụ: Tạo ràng buộc về giá trị khi tạo một mảng trong Datalog.

DES> :- type(t,[c:int])

DES> :- t(X),(X<0;X>10)

Khi ta thực thi như sau sẽ bị lỗi:

DES> /assert t(0)

DES> /assert t(11) //giá trị vượt quá 10

Error: Integrity constraint violation.

ic(X) :-

t(X),

X < 0,

X > 10.

Offending values in database: [ic(11)].

## 2.2. Chương trình Datalog

Một chương trình Datalog là một CSDL suy diễn xác định bao gồm tập hữu hạn các mệnh đề Horn:

p ← q1 ^...^ qn (n ≥ 0)

Trong đó các vị từ p, qi là các nguyên tố.

Như vậy chương trình Datalog là chương trình logic dương không chứa ký hiệu hàm.

### 2.2.1. Mô hình dữ liệu

Mô hình dữ liệu trong chương trình Datalog bao 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.

Một ngôn ngữ cấp một được xây dựng dựa trên một *bộ ký tự* và những công thức xây dựng trên bộ ký tự đó. Ngôn ngữ cấp một được dùng như kí pháp toán học để mô tả dữ liệu trong mô hình CSDL suy diễn và dữ liệu được xử lý trong các mô hình nhờ việc đánh giá công thức logic. Từ đó tiếp cận được nền tảng lý thuyết của các hệ thống CSDL suy diễn.

Tuy nhiên, để dễ biểu diễn khái niệm và các hình thức về CSDL suy diễn, ta thường dùng phép toán logic vị từ bậc nhất. Logic vị từ bậc nhất là ngôn ngữ hình thức dùng để thể hiện quan hệ giữa các đối tượng và để suy diễn ra quan hệ mới.

### 2.2.2. Cú pháp

* Kí hiệu: Các kí hiệu trong bảng chữ cái.
* Vị từ so sánh: <Tên thuộc tính> ϴ <Giá trị>

Hay <Biến> so sánh với <Giá trị>

ϴ ={<,>,=,<=,>=,<>}

Cách biểu diễn các luật (Clause – Rule)

Q ←P1,P2,…Pn

Dấu phẩy “ , ” tương đương với AND (^)

Dấu chấm phẩy “ ; ” tương đương với OR (v)

Dấu “←”: kéo theo

Q: là kết luận hay là sự kiện.

Pi: là các giả thiết, tiên đề, vị từ, đích con.

Nếu n=0 : Q ← các sự kiện của CSDL cài đặt.

Nếu P←P1,P2,…,Pn thì P là luật đệ quy (hay vị từ ở trong thân và đầu luật).

### 2.2.3. Ngữ nghĩa

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

Ví dụ:

(r1) Cha\_mẹ(x,y) ←Bố(x,y)

(r2) Cha\_mẹ(x,y) ←Mẹ(x,y)

(r3) Ông\_bà(x,y) ← Cha\_mẹ(x,z), Cha\_mẹ(z,y)

(r4) Bố(x,y) ←(r7): Tổ\_Tiên(x,y) ← Cha\_mẹ(x,z), Tổ\_Tiên(z,y)

(r5) Mẹ(x,y) ←(r6): Tổ\_Tiên(x,y), Cha\_mẹ(x,y)

### 2.2.4. Cấu trúc cơ bản

CSDL Datalog gồm hai loại quan hệ:

* Các quan hệ cơ sở được lưu trữ trong CSDL. Cơ sở này được gọi CSDL mở rộng EDB (Extended Database)
* Các quan hệ suy diễn không cần lưu trong CSDL. Chúng được dùng như quan hệ tam thời, chứa các kết quả trung gian khi trả lời câu hỏi. Cơ sở này được gọi là CSDLtheo mục đích IDB (Intentional Database)

Các chương trình Datalog có một tập hữu hạn các luật tác động đến các quan hệ suy diễn và quan hệ cơ bản.

Các thuộc tính của mỗi quan hệ trong Datalog không mang tên hiện rõ. Thay vì tên, mỗi thuộc tính căn cứ vào giá trị của nó.

Ta có một số công thức như sau:

(i) Các luật được xây dựng trên các Literal có dạng: P(A1, A2,...,An), trong đó P là tên của quan hệ cơ sở hay quan hệ suy diễn. Mỗi Ai (i = 1,2,...,n) là hằng số hay tên biến.

(ii) Một luật Datalog có dạng: P(X1, X2,...,Xn)←Q1(X11, X12,...,X1m1), Q2(X21, X22,...,X2m2),..., Qr(Xr1, Xr2,...,Xrmr), e

Trong đó:

* P : là tên của quan hệ suy diễn;
* Mỗi Qi là tên của quan hệ cơ sở hay quan hệ suy diễn;
* e là biểu thức vị từ số học đối với các biến xuất hiện trong P và tất cả các Qi (mỗi biến xuất hiện trong P cũng xuất hiện trong Qi nào đó).

Literal P(X1, X2,...,Xn) gọi là đầu của luật, phần còn lại gọi là thân của luật.

Để hiểu chính xác cách thức diễn giải một luật trong Datalog, người ta xác định khái niệm thay thế luật và hiện trạng của luật.

### 2.2.5. Truy vấn trong Datalog

#### 2.2.5.1. Khái niệm

Một câu truy vấn trong CSDL suy diễn bao gồm:

* Một tập hữu hạn, có thể rỗng của các luật được biểu diễn thành chương trình Datalog.
* Một literal đơn có dạng P(x1,x2,...,xn)? Trong đó, xi (i=1,2,...,n) là hằng số hoặc tên biến.

Việc khai thác câu hỏi trước tiên là tính chương trình Datalog (nếu có), tiếp theo P(x1,x2,...,xn) sẽ được đánh giá.

#### 2.2.5.2. Cấu trúc

Để trình bày cấu trúc của câu hỏi người ta sử dụng đồ thị luật (Rule Graph)

Một đồ thị luật đối với câu hỏi q là đồ thị có hướng mà:

* Các nút của đồ thị ứng với tập các kí hiệu Literal có mặt trong các luật của q.
* Cung của đồ thị ứng với quan hệ trước giữa Literal trong thân của luật và Literal có mặt trong đầu của luật đó.

Do vậy đồ thị sẽ có cung ai ←aj nếu luật này có mặt trong câu hỏi ai ←...aj...

Chú ý:

* Việc xây dựng này không tính đến tập các biến và các hằng số có mặt trong các luật đa dạng của câu hỏi này.
* Thông tin duy nhất người ta dùng là tập các ký hiệu.

#### 2.2.5.3. So sáng Datalog với đại số quan hệ

Về cơ bản, ngôn ngữ Datalog với các câu hỏi không đệ quy được xem như tương đương với đại số quan hệ về khả năng thể hiện.

Với các câu hỏi đệ quy được xem như là một công cụ cho phép người truy vấn dữ liệu nhanh hơn, dễ dàng hơn các ngôn ngữ đại số quan hệ. Điều này ngôn ngữ Datalog cho phép hỏi các câu hỏi không được phép trong đại số quan hệ.

1. **Phép hợp**: Là tập các luật có cùng đầu luật:

Hỏi(X1, X2,...,Xn) ← r1(X1, X2, ..., Xn)

Hỏi(Y1, Y2,...,Yn) ← r2(Y1, Y2, ..., Yn)

Ví dụ:

Tìm tên của các khách hàng tại chi nhánh “Hà Nội”:

Hỏi(Y) ←Vay(“Hà Nội”, X,Y,Z)

Hỏi(B) ←Gửi\_Tiền(“Hà Nội”,A,B,C)

1. **Phép chọn**: ứng với một luật mà thân luật có một vị từ so sánh, còn gọi là biểu thức chọn.

Phép chọn các n\_bộ trong quan hệ r được viết dưới dạng câu hỏi: r(x1, x2,..,xn)?. Trong đó, xi (i=1,2,...n) là tên biến hay một hằng số.

Ví dụ:

Chọn (tìm kiếm) tên của những khách hàng vay quá 1000?

Hỏi(Y) ←Vay(“Hà Nội”, X,Y,Z), z>1000

1. **Phép chiếu:** là phép toán tương ứng với một số luật mà có một số biến ở thân luật mà không xuất hiện ở đầu luật.

Ví dụ:

Cha(x) = KQ(x) ← Cha\_mẹ(x, y), y=”Dũng”.

1. **Phép kết nối:** là phép ứng với luật mà có biến chung ở các vị từ của thân luật. Phép kết nối hai quan hệ r1, r2 được viết dưới dạng Datalog như sau: Hỏi(X1, X2,...,Xn, Y1, Y2,...,Ym) r1(X1, X2,..,Xn), r2(Y1, Y2,..,Ym). Trong đó, Xi, Yj (i=1,2,...,n và j=1,2,...,m) là các tên biến phân biệt nhau.
2. **Khả năng đệ quy:**

Ví dụ: Giả sử có lược đồ quan hệ: Quản lý( Tên nhân công, tên người quản lý).

Lược đồ này thể hiện mối quan hệ giữa người quản lý và nhân công. Giả sử “Quản lý“ là một quan hệ theo mô hình trên.

|  |  |
| --- | --- |
| Tên công nhân | Tên người quản lý |
| Cường | Đông |
| Hoa | Cường |
| Quyền | Cường |
| Lan | Cường |

Bảng 2.2. Bảng quan hệ Quản lý công nhân

Yêu cầu:

1. Tìm tên của những người làm việc trực tiếp dưới quyền của ông Cường, tức phụ thuộc mức 1, viết như sau:

Hỏi(X) ← Quản lý(X, “Cường”)

2) Tìm tên của những người trực tiếp dưới quyền của người do ông Cường quản lý, tức phụ thuộc mức 2 vào ông Cường, ta viết như sau: Hỏi(X) ← Quản lý(X, Y), Quản lý(Y, “Cường”).

Như vậy người ta không thể thể hiện yêu cầu tìm người phụ thuộc bậc n vào ông Cường trong đại số quan hệ được. Dĩ nhiên câu hỏi tìm tên của nhân công làm việc dưới quyền của ông Cường, trực tiếp hay gián tiếp, không thể tạo được bằng đại số quan hệ hay bằng Datalog với các câu hỏi không đệ quy. Nguyên nhân là do người ta không biết ông Cường quản lý đến mức nào. Tuy nhiên người ta có thể tạo câu hỏi này trong Datalog dưới dạng câu hỏi đệ quy như sau:

e(X) ← Quản lý(X, “Cường”)

e(X) ← Quản lý(X, Y), e(Y)

Hỏi(X) ← e(X)

**Chú ý:**

*Cách a.*

Đối với những câu hỏi đệ quy người ta cũng có thể chuyển về câu hỏi không đệ quy bằng cách sử dụng ngôn ngữ tựa pascal với một số lần hữu hạn các bước lặp. Việc lặp được thể hiện qua câu lệnh Repeat. Điều kiện trong câu Until sẽ kiểm tra về tập hợp, như tính bằng nhau, bao nhau hay rỗng. Trong câu Until các quan hệ suy diễn được coi như các tập. Do vậy câu hỏi đệ quy trên có thể được viết lại như sau:

e’(X) ← Quản lý(X, “Cường”)

Repeat

e(X) ← e’(X) e’(X) ← Quản lý(X, Y), e(Y)

Until e=e’

Mô tả:

* Luật đầu tiên tìm nhân công mà ông Cường trực tiếp quản lý. Khi hoàn thành các luật trong vòng Repeat được đánh giá.
* Tại mỗi lần lặp, mức tiếp theo của nhân công được tìm và bỗ sung vào tập e.
* Thủ tục này kết thúc khi tập e = e’( khi không còn nhân công mới có thể được bỗ sung vào e). Mặt khác, do tập những người quản lý là hữu hạn.

Cách thực hiện: Theo dõi chu trình với các dữ liệu trong bảng khi chạy.

e’ ={Hoa, Lan, Quyền }

e = { Hoa, Lan, Quyền }

Lặp lại

e’ ={ Hoa, Lan, Quyền, Chén, Tích} e = { Hoa, Lan, Quyền, Chén, Tích}

*Cách b.*

Ngoài cách làm như trên người ta cũng có thể làm theo cách khác mà vẫn đạt được kết quả như nhau:

m(X,Y) ← Quản lý(X, Y)

m(X,Y) ← Quản lý(X, Z), m(Z,Y)

Hỏi(X) ← m(X, “Cường”)

So sánh cách a và cách b:

- Cách a. Tìm ra các nhân công của ông Cường. Cách này cho phép tìm nhanh hơn.

- Cách b. Tìm tất cả quan hệ nhân công – người quản lý rồi chọn ra các cặp có tên người quản lý là ông Cường.

#### 2.2.5.4. Chương trình Datalog có chứa kí hiệu hàm

Ta giả sử các hạng thức trong quan hệ EDB là hạng thức nền, nghĩa là không chứa biến.

Khi các quy tắc là đệ quy, việc sử dụng các ký hiệu hàm có thể tạo ra một tập vô hạn các giá trị trong các quan hệ, như ví dụ sau đây:

Phép cộng có thể định nghĩa đệ quy như sau: Giả sử int(X) là ký hiệu vị từ và int(X) là true nếu X nguyên (không âm) và vị từ sum(X,Y,Z) là true nếu Z=X+Y. Giả sử succ là ký hiệu hàm và succ(X) là số nguyên X+1. Hằng 0 là số nguyên. Lúc đó succ(0) ta gọi là "1", succ(succ(0)) là "2", các số nguyên khác ngoại trừ 0 là không tồn tại trong dạng khác. Khi đó ta có các quy tắc sau đây thỏa mãn định nghĩa trên :

(1) int(0)

(2) int(succ(X)) ← int(X)

(3) sum(X,0,X) ← int(X)

(4) sum(X,succ(Y),succ(Z)) ← sum(X,Y,Z)

Hai quy tắc (1) và (2) định nghĩa 0, succ(0), succ(succ(0)),...là các số nguyên,...

Quy b 0 tắc (3) nói rằng tổng của mọi số nguyên với 0 là chính số nguyên đó.

Quy tắc (4) nói rằng X+(Y+1) = Z nếu X+Y=Z.

Nhận xét :

* Các vị từ int và sum tạo ra các quan hệ vô hạn mặc dù các quy tắc đã cho là an toàn.
* Mặc dù cú pháp đối với các ký hiệu hàm và các ký hiệu logic vị từ là như nhau (chuỗi bắt đầu bằng chữ thường) nhưng hoàn toàn không có sự nhầm lẫn giữa hai kiểu ký hiệu này. Các ký hiệu hàm lấy giá trị trên miền như các đối số và trả về các giá trị của miền, các ký hiệu vị từ cũng lấy giá trị trên miền như các đối số nhưng trả về giá trị chân lý (true hoặc false). Như ở ví dụ trên thì succ lấy một đối số là số nguyên và trả về một số nguyên, trong khi sum lấy ba đối số nguyên và trả về true nếu thành phần thứ ba bằng tổng thành phần thứ nhất và thứ hai, ngược lại nó trả về false.

# Chương 3. CÀI ĐẶT DEMO VÀ ỨNG DỤNG

## 3.1 Công cụ thực hiện

Yêu cầu phần cứng: Laptop Core i5 64bit, 2.2 GHz, Ram 4GB dùng để xây dựng và chạy phần mềm.

Yêu cầu phần mềm:

* Microsoft Visual Studio Professional 2013 để xây dựng phần mềm.
* Bộ thư viện SWI – Prolog cho Windows 64 bit.
* Notepad để xây dựng CSDL cho chương trình.

## 3.2. Đặt vấn đề bài toán

Xây dựng chương trình tư vấn hướng nghiệp là giúp người sử dụng có thể xác định được những công việc phù hợp với mình trong tương lai. Và để tìm kiếm nghề nghiệp thích hợp, chương trình cần người sử dụng cung cấp các thông tin về năng khiếu, sở thích và tiềm năng của họ. Chương trình sẽ đưa ra những câu hỏi dưới dạng Yes – No question cho người sử dụng để thực hiện việc thu thập thông tin cần thiết. Sau khi trả lời câu hỏi, chương trình sẽ đưa ra được những kết luận nghề nghiệp phù hợp với sở thích, năng khiếu và tiềm năng của người sử dụng. Chẳng hạn như:

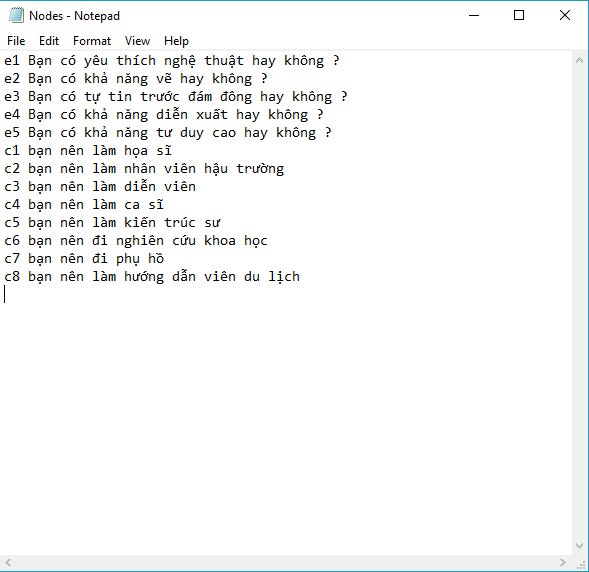
* Để xác định người sử dụng muốn làm công việc đi theo thiên hướng nghệ thuật hay thực tế, khoa học, ta có các câu hỏi sau:
* Bạn có yêu thích nghệ thuật không?
* Trong tương lai, bạn có muốn một công việc thực tế khoa học hay không?
* Trong quan điểm của bạn, những giá trị thực tế và khoa học có quan trọng hơn giá trị về tinh thần hay không?
* Để xác định tiềm năng của người sử dụng, ta có thể đặt các câu hỏi sau:
* Bạn có khả năng nói trước đám đông không ?
* Bạn có khả năng tư duy cao hay không?
* Bạn cảm thấy dễ dàng giải quyết những câu hỏi hóc búa ít người giải được ?
* Theo bản thân, bạn nhận thấy mình là một người tự tin hay không ?
* Để xác định người sử dụng có các năng khiếu gì, chúng ta cần đặt các câu hỏi cho từng loại năng khiếu, chẳng hạn như:
* Bạn có khả năng vẽ hay không?
* Bạn có khả năng diễn xuất hay không?

## 3.3. Quy trình xử lý bài toán

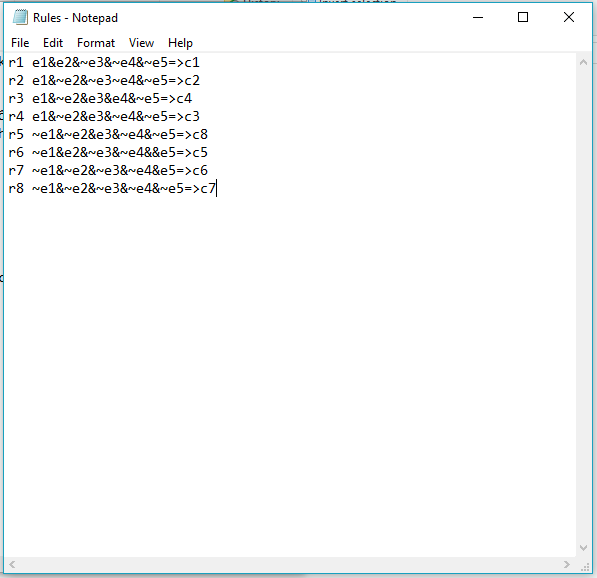
Thu thập các ngành nghề đang được quan tâm để xây dựng các sự kiện trong phần kết luận của các luật.

Căn cứ vào ngành nghề có được, chúng ta sẽ xác định được những yếu tố cần thiết cho công việc đó. Thông qua đó chúng ta xây dựng được các luật cho chương trình. Cũng từ việc căn cứ vào nghề nghiệp, chúng ta có thể khái quát được toàn bộ các sự kiện ở phần giả thiết trong CSDL chính là các yếu tố cần thiết cho tất cả các công việc mà chương trình tư vấn.

Sau khi xác định được các sự kiện và các luật, chúng ta có thể bắt tay vào xây dựng CSDL trong Prolog. Nhưng với quy mô hạn chế của một chương trình phục vụ cho mục đính học tập, nghiên cứu, tìm hiểu, chúng em sẽ xây dung CSDL trên một file văn bản để truyền trực tiếp vào chương trình không thông qua việc nạp dữ liệu giữa C# và Prolog.



Hình 3.1. Các sự kiện của chương trình



Hình 3.2. Các luật của chương trình

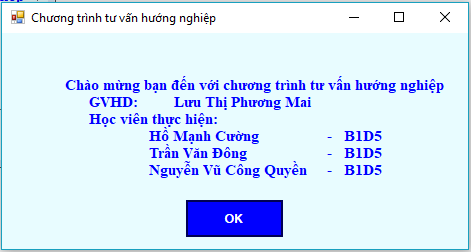
Các yếu tố cần thiết cho các công việc đã đề cập ở bên trên cũng chính là các thông tin chương trình quan tâm để có thể thực hiện công việc của mình. Vì thế chương trình sẽ yêu cầu người dùng cung cấp các thông tin trên bằng cách đặt ra câu hỏi và người dùng sẽ cung cấp các thông tin đó bằng cách khẳng định hay phủ định nôi dung câu hỏi. Từ các thông tin được cung cấp, chương trình sẽ đối chiếu với các sự kiện và các luật trong CSDL để đưa ra câu trả lời.

Bên cạnh đó, chúng ta sẽ cần phải xây dựng các chức năng thêm, sửa, xóa các yếu tố, nghề nghiệp, các luật để có thể tư vấn chính xác,kịp thời,phù hợp với tình hình, nhu cầu và xu hướng việc làm của xã hội.

## 

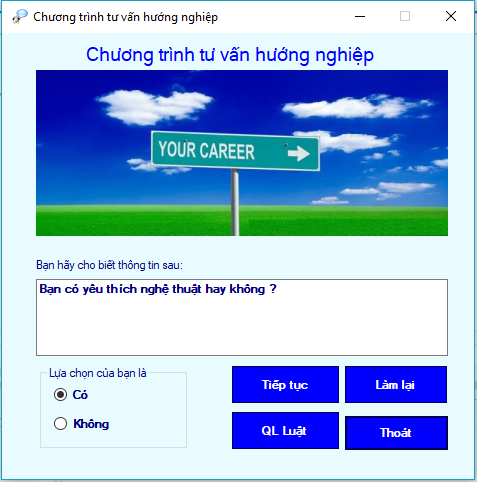
## 3.4. Demo chương trình

Bước 1: Khởi động chương trình.

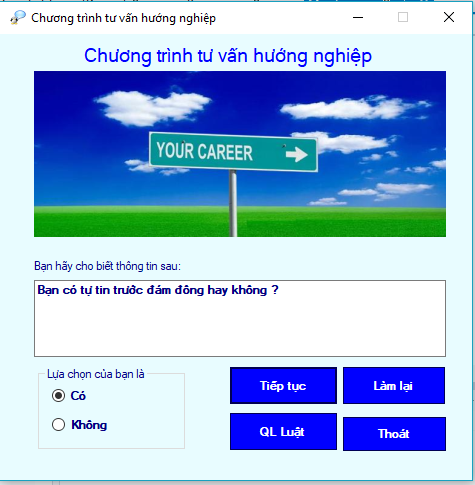


Hình 3.3. Giao diện khởi động của chương trình

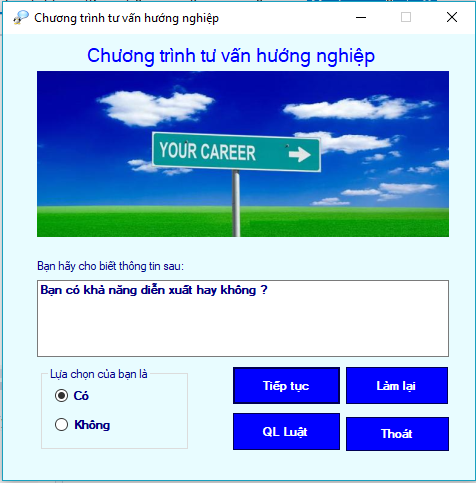
Bước 2: Lựa chọn câu trả lời và nhấn nút tiếp tục để cung cấp thông tin cho chương trình cho đến khi chương trình đưa ra kết luận hoặc nhấn nút làm lại để bắt đầu lại từ đầu.



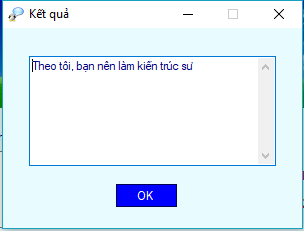
Hình 3.4. Giao diện chính của chương trình



Hình 3.5. Giao diện chính của chương trình

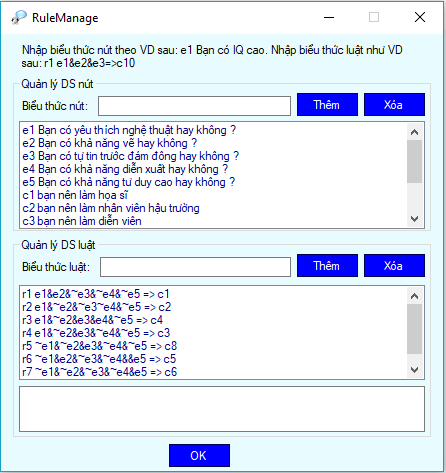


Hình 3.6. Giao diện chính của chương trình



Hình 3.7. Chương trình đưa ra kết luận

Nhấn nút QL luật để có thể thực hiện việc thêm bớt, sửa xóa các sự kiện à luật của chương trình.



Hình 3.8. Giao diện quản lý luật của chương trình

# 

# Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1. Kết quả đạt được

Qua quá trình nghiên cứu đề tài này, chúng em đã giới thiệu một cách tương đối đầy đủ về CSDL suy diễn, về ý nghĩa cũng như ứng dung của CSDL suy diễn vào thực tế thông qua ngôn ngữ logic; đồng thời giới thiệu một ngôn ngữ lập trình để biểu diễn cơ sở tri thức bằng logic mô tả, đó là ngôn ngữ Prolog. Đề tài cũng nêu khả năng thực hiện truy vấn thông qua ngôn ngữ Datalog, biến đổi các luật Datalog thành các mô tả khái niệm trong logic mô tả. Dựa trên những tìm hiểu và nghiên cứu trong suốt thời gian vừa qua, chúng em xây dựng một phần mềm đơn giản ứng dụng CSDL suy diễn trong thực tế, đó là chương trình “Tư vấn hướng nghiệp”.

## 4.2. Các mặt chưa làm được

Tuy nhiên, ngoài những công việc đã làm được, chúng em xây dựng đề tài này mới mang tính lý thuyết. Do hạn chế về thời gian cũng như tài liệu tham khảo, và còn hạn chế về tri thức, chúng em chỉ mới xây dựng được chương trình DEMO đơn giản, nên chắc còn nhiều thiếu sót, hạn chế. Chương trình DEMO của chúng em còn chưa hoàn thiện về nhiều chức năng như cung cấp CSDL trong các lĩnh vực, cung cấp thông tin các ngành nghề phổ biến hiện nay cùng với thống kê, phân tích, đánh giá tình hình việc làm trong nhiều năm gần đây, đưa ra các chỉ số đánh giá chất lượng việc làm tăng, giảm, giá trị trung bình,…vv. Kết quả chương trình chưa cụ thể, chi tiết, độ chính xác còn có những hạn chế. Vì vậy, chúng em dự định trong tương lai sẽ đi sâu hơn, hoàn thiện chương trình để xây dựng được một chương trình ứng dụng thực tế với quy mô lớn hơn, thực tiễn hơn.

## 4.3. Phương hướng phát triển

Chương trình tư vấn hướng nghiệp là phần mềm có tính thực tiễn cao. Hiện tại phần mềm của chúng em còn nhiều thiếu sót, cần xây dựng và phát triển thêm để phần mềm được hoàn thiện hơn và phát triển đầy đủ các chức năng.

Phương hướng phát triển của nhóm sẽ là đi nghiên cứu từ lý thuyết cho đến thực tiễn, sẽ tiến tới phát triển phần mềm theo quy mô hơn, chuyên sâu hơn cùng với đó là phát triển phần mền quản lý rộng hơn với các quy trình tư vẫn hướng nghiệp phân loại cho từng đối tượng trong quy mô ngày càng ngày được mở rộng, đi kèm cùng với chất lượng, tính ứng dụng ngày càng được nâng cao.

- Xây dựng, thiết lập một số chức năng mới như: thu thập, thống kê, cung cấp thông tin các ngành nghề phổ biến hiện nay, xây dựng bảng biểu, chỉ số, thông số đánh giá chất lượng nghề, việc làm để tư vấn phù hợp cho người sử dụng.

- Mở rộng phạm vi của đề tài áp dụng cho nhiều đối tượng với phạm vi, không gian và thời gian lớn hơn.

- Thiết kế phần mềm đẹp hơn, tiện dụng hơn, có nhiều chức năng hơn và sinh động hơn.

- Chỉnh sữa, thu thập cơ sở dữ liệu cho chính xác với thực tế và sẽ nâng cấp dần phần mềm để đáp ứng được những yêu cầu đề ra.

Chương trình tạo ra có một số ưu điểm, tuy nhiên do còn thiếu kinh nghiệm và hạn chế nhất định về kiến thức nên sản phẩm vẫn còn một số thiếu sót. Chúng em rất mong nhận được sự góp ý từ các thầy cô để chương trình được hoàn thiện hơn. Chúng em xin bày tỏ lòng biết ơn đến cô Lưu Thị Phương Mai đã trực tiếp giúp đỡ, truyền đạt kiến thức quý báu và tận tình hướng dẫn của cô.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

[1] Đinh Nguyễn Anh Dũng, “Biểu diễn tri thức”(2001). NXB Thống Kê.

[2] Vũ Đức Thi, ”Cơ sở dữ liệu”(2004). NXB Thống kê.

[3] Phan Huy Khánh, “Lập trình logic trong Prolog”(2004), NXB Đại học quốc gia Hà Nội.

[4] Lê Mạnh Thạnh, “Giáo trình Datalog”(2004).

**Danh mục các Website tham khảo:**

[1] http:// en.wikipedia.org/wiki/Deductive\_database (6/4/2018)

[2] <https://text.123doc.org/document/3299867-nghien-cuu-co-so-du-lieu-suy-dien-voi-phan-mem-des-va-ung-dung.htm>

[3] <https://123doc.org/document/2429056-luan-van-thac-si-cong-nghe-thong-tin-nghien-cuu-co-so-du-lieu-suy-dien-va-ung-dung.htm>