

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH



BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM HỌC 2023-2024

**NGHIÊN CỨU ONTOLOGY VÀ MÔ HÌNH TRI THỨC CÁC ĐỐI TƯỢNG
TÍNH TOÁN COKB ĐỂ XÂY DỰNG HỆ THỐNG HỖ TRỢ GIẢI TOÁN
THÔNG MINH - ÁP DỤNG VÀO HỌC PHẦN TOÁN RỜI RẠC DÀNH
CHO SINH VIÊN CÔNG NGHỆ THÔNG TIN TẠI PHÂN HIỆU**

Sinh viên thực hiện

- | | | |
|------------------------------|------------------------------|---------------------------|
| 1. Phạm Thị Ngọc Oanh | Lớp: Công nghệ thông tin K63 | Khoa: Công nghệ thông tin |
| 2. Hà Văn Dũng | Lớp: Công nghệ thông tin K63 | Khoa: Công nghệ thông tin |
| 3. Đỗ Văn Thành Được | Lớp: Công nghệ thông tin K63 | Khoa: Công nghệ thông tin |
| 4. Phan Quang Phát | Lớp: Công nghệ thông tin K61 | Khoa: Công nghệ thông tin |
| 5. Võ Công Sinh | Lớp: Công nghệ thông tin K63 | Khoa: Công nghệ thông tin |

Người hướng dẫn: TH.S. Trần Phong Nhã

TPHCM, 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH



BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN
NĂM HỌC 2023-2024

**NGHIÊN CỨU ONTOLOGY VÀ MÔ HÌNH TRI THỨC CÁC ĐỐI TƯỢNG
TÍNH TOÁN COKB ĐỂ XÂY DỰNG HỆ THỐNG HỖ TRỢ GIẢI TOÁN
THÔNG MINH - ÁP DỤNG VÀO HỌC PHẦN TOÁN RỜI RẠC DÀNH CHO
SINH VIÊN CÔNG NGHỆ THÔNG TIN TẠI PHÂN HIỆU**

Sinh viên thực hiện

- | | | |
|------------------------------|--------------------------|---------------|
| 1. Phạm Thị Ngọc Oanh | Giới tính: Nữ | Dân tộc: Kinh |
| Lớp: Công Nghệ Thông Tin K63 | Khoa Công Nghệ thông tin | Năm thứ: 2/4 |
| 2. Hà Văn Dũng | Giới tính: Nam | Dân tộc: Kinh |
| Lớp: Công Nghệ Thông Tin K63 | Khoa Công Nghệ thông tin | Năm thứ: 2/4 |
| 3. Đỗ Văn Thành Được | Giới tính: Nam | Dân tộc: Kinh |
| Lớp: Công Nghệ Thông Tin K63 | Khoa Công Nghệ thông tin | Năm thứ: 2/4 |
| 4. Phan Quang Phát | Giới tính: Nam | Dân tộc: Kinh |
| Lớp: Công Nghệ Thông Tin K61 | Khoa Công Nghệ thông tin | Năm thứ: 4/4 |
| 5. Võ Công Sinh | Giới tính: Nam | Dân tộc: Kinh |
| Lớp: Công Nghệ Thông Tin K63 | Khoa Công Nghệ thông tin | Năm thứ: 2/4 |

Người hướng dẫn: TH.S. Trần Phong Nhã

TPHCM2024

MỤC LỤC

DANH MỤC NHỮNG TỪ VIẾT TẮT	iii
DANH MỤC HÌNH ẢNH	iii
MỞ ĐẦU	iv
TỔNG QUAN	vi
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	1
1.1 Giới thiệu về Ontology	1
1.1.1 Khái niệm Ontology	1
1.1.2 Các thành phần của Ontology	1
1.1.3 Các bước xây dựng Ontology	3
1.2 Giới thiệu về mô hình COKB	6
1.2.1 Định nghĩa	6
1.2.2 Ngôn ngữ đặc tả cho mô hình COKB	7
1.2.3 Thành phần tri thức hàm	7
1.2.4 Các loại sự kiện trong mô hình COKB	7
1.3 Công nghệ và ngôn ngữ lập trình	8
1.3.1 React.js	8
1.3.2 Spring boot	9
1.3.3 Lập trình ký hiệu(Symbolic)	9
1.3.4 Ngôn ngữ Maple	10
1.3.5 Cơ sở dữ liệu (MySQL)	10
1.4 Hệ thống giải toán thông minh	11
1.4.1 Mô hình hệ luật dẫn	11
1.4.2 Cơ chế suy luận trên hệ luật dẫn	12
CHƯƠNG 2: MÔ HÌNH BÀI TOÁN	13
2.1 Mô hình bài toán	13
2.2 Thuật giải tìm lời giải cho bài toán tổng quát	14
2.4 Ứng dụng	15
2.4.1 Thiết kế hệ cơ sở tri thức	16

2.4.2 Thiết kế động cơ suy diễn của hệ thống.....	16
CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG	17
3.1 Mục tiêu triển khai thử nghiệm	17
3.2 Thiết kế ứng dụng.....	17
3.2.1 Thiết kế chức năng ứng dụng	17
3.3.2 Thiết kế kiến trúc hệ thống.....	18
3.4 Kết quả hiển thị	18
KẾT LUẬN	23
TÀI LIỆU THAM KHẢO.....	25

DANH MỤC NHỮNG TỪ VIẾT TẮT

TỪ VIẾT TẮT	TỪ VIẾT ĐẦY ĐỦ	Ý NGHĨA
COKB	Computational Objects Knowledge Base	Cơ sở tri thức các đối tượng tính toán
Com-Object	Computational Object	Đối tượng tính toán
STEM	Science Technology Engineering and Math Education	Là một phương pháp giáo dục đặc biệt tập trung vào việc phát triển kỹ năng và kiến thức trong các lĩnh vực khoa học, công nghệ, kỹ thuật và toán học

DANH MỤC HÌNH ẢNH

Hình 1-1: Ví dụ minh họa biểu diễn tri thức trong Ontology	1
Hình 1-2: Sơ đồ tổ chức cơ sở tri thức theo mô hình COKB	7
Hình 2-1: Mô tả đồ thị	13
Hình 3-1: Mô tả thiết kế kiến trúc hệ thống	18
Hình 3-2: Mô tả tìm kiếm theo từng chương	19
Hình 3-3: Mô tả tìm kiếm theo từ khóa - tìm kiếm theo chiều rộng	20
Hình 3-4: Mô tả tìm kiếm bằng chatbot- tìm kiếm chiều sâu	20
Hình 3-5: Mô tả chức năng xem bài viết	21
Hình 3-6: Mô tả nội dung của 1 bài viết	21
Hình 3-7: Mô tả diễn đàn câu hỏi	22
Hình 3-8: Mô tả comment của người bình luận	22

MỞ ĐẦU

Trong thời đại của sự phát triển không ngừng của công nghệ thông tin và trí tuệ nhân tạo, việc áp dụng các phương pháp nghiên cứu ontology và mô hình tri thức đối với các đối tượng tính toán COKB (các khái niệm, các quan hệ, các hành động và các nguyên tắc tri thức) đã trở thành xu hướng không thể phủ nhận trong việc xây dựng các hệ thống hỗ trợ giải toán thông minh. Trong ngữ cảnh của giáo dục đại học, việc áp dụng và phát triển các hệ thống thông minh này trở nên cực kỳ quan trọng, đặc biệt là trong học phần Toán rời rạc - một phần không thể thiếu trong chương trình đào tạo của sinh viên ngành Công nghệ thông tin.

Sự phức tạp của các vấn đề trong Toán rời rạc thường đòi hỏi sự hiểu biết sâu sắc về các khái niệm cơ bản và mối quan hệ giữa chúng. Điều này không chỉ đòi hỏi sự ghi nhớ mà còn yêu cầu khả năng áp dụng, phân tích và suy luận logic. Trong bối cảnh này, việc sử dụng các công nghệ thông minh dựa trên ontology và mô hình tri thức có thể giúp sinh viên hiểu và áp dụng Toán rời rạc một cách hiệu quả hơn.

Lợi ích của việc áp dụng ontology và mô hình tri thức trong học phần Toán rời rạc:

- Hỗ trợ giải quyết vấn đề: Các hệ thống thông minh có thể cung cấp hỗ trợ trong việc giải quyết các bài toán phức tạp thông qua việc phân tích và áp dụng các nguyên tắc tri thức. Từ đó người học không chỉ học được cách giải quyết vấn đề mà còn phát triển kỹ năng suy luận logic và tư duy sáng tạo.
- Cung cấp đáp án và hướng dẫn giải cho người học, từ đó người học không cần phải xác minh các lời giải xung quanh (trên mạng, bạn bè,...) xem có đúng hay không

Thách thức và cơ hội:

- Thách thức về độ phức tạp: Xây dựng và triển khai các hệ thống thông minh dựa trên ontology và mô hình tri thức đòi hỏi sự đầu tư lớn về thời gian, công sức và tài nguyên. Việc phát triển một ontology phù hợp và mô hình tri thức chính xác đòi hỏi sự hiểu biết sâu rộng về lĩnh vực cụ thể cũng như kỹ năng kỹ thuật.
- Cơ hội trong việc cá nhân hóa học phần: Các hệ thống thông minh có thể được tùy chỉnh để phù hợp với nhu cầu học tập cụ thể của từng sinh viên. Việc này giúp tăng cường sự hứng thú và hiệu quả học tập bằng cách cung cấp nội dung phù hợp và phương pháp học tập linh hoạt.
- Tạo ra môi trường học tập tích cực: Các hệ thống hỗ trợ giải toán thông minh có thể tạo ra một môi trường học tập tích cực, thúc đẩy sự tương tác và hợp tác giữa sinh viên

và giảng viên. Việc phát triển các công cụ giáo dục dựa trên công nghệ này có thể thúc đẩy sự hòa nhập và phát triển cá nhân của sinh viên.

Bài viết này sẽ tập trung vào khả năng ứng dụng của nghiên cứu ontology và mô hình tri thức trong việc xây dựng các hệ thống hỗ trợ giải toán thông minh, đặc biệt là áp dụng vào học phần Toán rời rạc dành cho sinh viên Công nghệ thông tin Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh. Chúng em sẽ đi sâu vào những lợi ích mà việc áp dụng các phương pháp này mang lại, cùng với những thách thức và cơ hội trong việc triển khai thực tế.

TỔNG QUAN

Tổng quan về Ontology

- Thuật ngữ “Ontology” đã xuất hiện từ rất sớm. Trong cuốn sách “Siêu hình” (Metaphysics) của mình, Aristotle đã định nghĩa: “Ontology là một nhánh của triết học, liên quan đến sự tồn tại và bản chất các sự vật trong thực tế”.
- Trong ngành khoa học máy tính và khoa học thông tin, Ontology mang ý nghĩa là các loại vật và quan hệ giữa chúng trong một hệ thống hay ngữ cảnh cần quan tâm. Các loại vật này còn được gọi là khái niệm, thuật ngữ hay từ vựng có thể được sử dụng trong một lĩnh vực chuyên môn nào đó. Ontology cũng có thể hiểu là một ngôn ngữ hay một tập các quy tắc được dùng để xây dựng một hệ thống Ontology. Một hệ thống Ontology định nghĩa một tập các từ vựng mang tính phổ biến trong lĩnh vực chuyên môn nào đó và quan hệ giữa chúng. Sự định nghĩa này có thể được hiểu bởi cả con người lẫn máy tính. Một cách khái quát, có thể hiểu Ontology là "một biểu diễn của sự khái niệm hoá chung được chia sẻ" của một miền hay lĩnh vực nhất định. Nó cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, Ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một lĩnh vực mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán khác.
- Tuy nhiên Ontology vẫn có 1 số hạn chế như sau
 - Chỉ biểu diễn được một khía cạnh nào đó của kiến thức thực tế. Tri thức thực tế cần được biểu diễn thường rất đa dạng, bao gồm các khái niệm từ đơn giản đến có cấu trúc phức tạp, các hệ thức tính toán với những quy luật nhất định, các liên hệ đa dạng bao gồm cả định tính lẫn định lượng, các luật dẫn và các heuristics...
 - Các mô hình này chưa đáp ứng được việc biểu diễn tri thức bao hàm nhiều dạng thông tin và sự kiện khác nhau trong thực tế.
 - Cơ sở tri thức và hệ thống suy diễn chưa thể hiện sự độc lập rõ ràng, chưa chú trọng đến thiết kế và xây dựng hệ thống suy diễn chuyên biệt hoạt động dựa trên nền cơ sở tri thức.
 - Chưa có một ngôn ngữ đặc tả qui ước tổng quát.

Để giải quyết những vấn đề tồn tại nêu trên mà một mô hình Ontology mới được gọi là mô hình tri thức các đối tượng tính toán được gọi là Ontology-COKB (Computational Objects Knowledge Base)

Hệ thống giải toán thông minh (intelligent problem solver)

- Việc xây dựng các hệ thống thông minh trong giáo dục về toán học và khoa học công nghệ (Science Technology Engineering and Math Education - STEM) có ý nghĩa lớn trong lĩnh vực giáo dục (Noy and McGuinness, 2013). Biểu diễn tri thức đóng vai trò quan trọng trong việc thiết kế các hệ cơ sở tri thức và động cơ suy diễn trong các hệ thống thông minh. Hiện nay, có nhiều phương pháp biểu diễn đã được nghiên cứu và ứng dụng trong các miền tri thức khác nhau như: framebased, mạng ngữ nghĩa, đồ thị khái niệm (Harmelen et al. 2008.). Bên cạnh một số mô hình tri thức được xây dựng chặt chẽ về lý thuyết (Aladova and Plotkin, 2017) thì logic mô tả là một dạng ngôn ngữ hình thức để biểu diễn tri thức (Baader et al., 2017). Logic mô tả được sử dụng trong việc xây dựng các web ontology. Tuy nhiên, các phương pháp này không thể biểu diễn đầy đủ các miền tri thức và rất khó ứng dụng trong việc xây dựng hệ thống ứng dụng trong thực tế, đặc biệt là ứng dụng trong lĩnh vực giáo dục STEM.

- Trong lĩnh vực giáo dục, hệ thống giải bài tập thông minh (intelligent problem solver) phải có một cơ sở tri thức đầy đủ để có thể hướng dẫn người học trong quá trình học, đặc biệt là trong việc giải quyết các bài toán. Người dùng chỉ cần khai báo các giả thiết và kết luận của bài toán theo một dạng ngôn ngữ đặc tả nhất định (Do, 2012). Sau khi đặc tả bài toán, người dùng có thể yêu cầu chương trình giải các bài toán đó hoặc đưa ra những hướng dẫn để giúp người dùng có thể giải quyết bài toán đó.

Mô hình tri thức các đối tượng tính toán- Ontology COKB

- Mô hình COKB được xây dựng theo cách tiếp cận hướng đối tượng, trong đó kết hợp các kỹ thuật biểu diễn cấu trúc, biểu diễn thủ tục, biểu diễn mạng, biểu diễn theo đối tượng và các kỹ thuật tính toán symbolic trên máy tính. Mô hình COKB được ký hiệu bởi bộ sáu (C, H, R, Ops, Funcs, Rules) thể hiện sáu dạng tri thức trong thực tế gồm: (1) Tập khái niệm về các đối tượng tính toán, (2) Biểu đồ H thể hiện quan hệ phân cấp trên các loại đối tượng, (3) Tập hợp R các quan hệ trên các loại đối tượng, (4) Tập hợp Ops thể hiện tri thức về các toán tử, (5) Tập hợp Funcs gồm các hàm và (6) Tập hợp Rules gồm các luật.

- Những ưu điểm của COKB so với các phương pháp biểu diễn truyền thống:

- Có cấu trúc tường minh và bộ sự kiện rất gần với tri thức thực tế.

- Có khả năng mô hình hóa được phần lớn các cơ sở tri thức thực tế, trên cơ sở đó thiết kế được các thuật giải tổng quát mô phỏng hành vi, suy luận giải quyết vấn đề dựa trên tri thức của con người.
 - Có đầy đủ ngôn ngữ khai báo và ngôn ngữ đặc tả một cách tự nhiên.
- Mô hình COKB đã mô hình hóa được rất nhiều cơ sở tri thức thực tế, biểu diễn được khá đầy đủ cho một miền tri thức thực của con người như Đại số tuyến tính, Hình học phẳng, Vật lý điện một chiều, điện xoay chiều ... Từ khi ra đời đến nay, mô hình COKB ngày càng được hoàn thiện và phát triển gắn với nhiều cơ sở tri thức khác nhau, đây là minh chứng thể hiện giá trị của mô hình COKB về cả mặt khoa học lẫn ứng dụng trong thực tế.
- Các phương pháp biểu diễn tri thức truyền thống đã được ứng dụng nhiều. Tuy nhiên, các phương pháp đó chưa đủ mạnh để xây dựng các chương trình tự động hay biểu diễn cơ sở tri thức thuộc các lĩnh vực tri thức khác nhau. Mô hình tri thức các đối tượng tính toán được xây dựng theo các tiếp cận hướng đối tượng kết hợp với các kỹ thuật lập trình tính toán hình thức. Theo hướng này sẽ cho phép mô hình được các bài toán cũng như thiết kế các giải thuật.
- Mô hình tri thức các đối tượng tính toán, viết tắt là COKB (Computational Objects Knowledge Base) đề xuất trong [1], là một hệ thống gồm 6 thành phần: (C,H,R,Ops,Funcs,Rules) trong đó mỗi thành phần sẽ được mô tả trong chương cơ sở lý thuyết. Tuy nhiên trong chương cơ sở lý thuyết sẽ không có thành phần H vì theo quan điểm của chúng em thì thành phần H trong đề tài này không có tác dụng nên chúng em sẽ không nói chi tiết về nó.

Lý do chọn đề tài

- Việc sử dụng các trang web giải toán thông minh đang trở thành một phần không thể thiếu trong quá trình học tập và giải quyết vấn đề của sinh viên, học sinh và người học mọi lứa tuổi. Những công cụ này không chỉ là một phương tiện tiện lợi để giải quyết các bài toán phức tạp mà còn mang lại nhiều lợi ích khác mà không thể phủ nhận.
- Trước hết, sự tiện lợi của các trang web giải toán thông minh không thể phủ nhận. Với một cú nhấp chuột, người dùng có thể truy cập vào một nguồn thông tin toán học không giới hạn và nhận được câu trả lời chính xác trong tích tắc. Thay vì phải tốn nhiều thời gian để giải quyết các bài toán phức tạp bằng tay, họ có thể dành thời gian hơn cho việc hiểu và ứng dụng kiến thức.

- Lợi ích tiếp theo của việc sử dụng các trang web giải toán thông minh là khả năng hỗ trợ học tập. Các công cụ này thường cung cấp các giải pháp chi tiết cho từng bước trong quá trình giải quyết bài toán. Điều này giúp người học hiểu rõ hơn về cách giải quyết một vấn đề cụ thể và cung cấp cơ sở cho việc học hỏi và phát triển kiến thức toán học của họ.
- Ngoài ra, các trang web giải toán thông minh cũng giúp kiểm tra đáp án và cung cấp phản hồi ngay lập tức. Điều này giúp người dùng kiểm tra xem họ đã giải quyết bài toán đúng hay sai và từ đó cải thiện kỹ năng giải toán của mình.
- Không chỉ có ích trong lĩnh vực học tập, các trang web giải toán thông minh cũng có thể được áp dụng vào thực tế. Chúng có thể giúp người dùng giải quyết các vấn đề phức tạp trong nhiều lĩnh vực như kinh tế, khoa học, công nghệ, và nhiều lĩnh vực khác.
- Tóm lại, việc sử dụng các trang web giải toán thông minh mang lại nhiều lợi ích không chỉ trong việc giải quyết các bài toán mà còn trong việc hỗ trợ học tập, cải thiện kỹ năng và ứng dụng kiến thức vào thực tế. Tuy nhiên, việc hiểu và áp dụng kiến thức mới vẫn là yếu tố quan trọng nhất trong quá trình học tập và phát triển cá nhân.

Mục tiêu của đề tài:

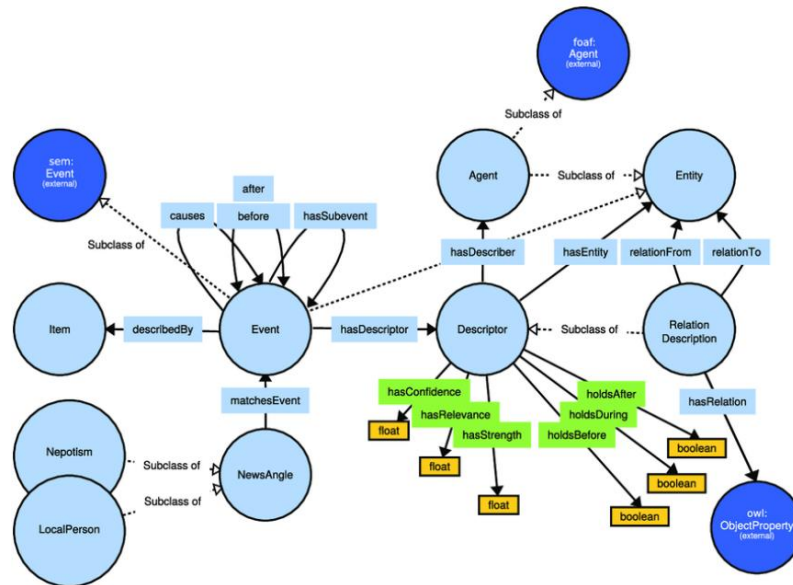
- Tập trung nghiên cứu Ontology và mô hình biểu diễn tri thức từ đó áp dụng vào thực tiễn cụ thể hơn là áp dụng giải toán của học phần toán rời rạc
- Xây dựng một trang web giải bài để minh họa cho kết quả nghiên cứu. Ngoài ra chúng em còn xây dựng những bài blog liên quan học phần toán rời rạc và xây dựng diễn đàn để người học (sinh viên) thoải mái trao đổi ý kiến với nhau.
- Xây dựng chức năng tra cứu theo keyword để tìm ra đáp án khi muốn tra cứu tài liệu lí thuyết.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Giới thiệu về Ontology

1.1.1 Khái niệm Ontology

- Ontology là một hình thức mô tả chính qui và tường minh các khái niệm được chia sẻ. Một ontology được thể hiện bằng các khái niệm và mối quan hệ giữa các khái niệm. Ontology là một trong các hình thức biểu diễn tri thức rộng rãi và tiên tiến nhất hiện nay. Với hình thức biểu diễn tri thức này, mô hình của các khái niệm và mối quan hệ giữa các khái niệm trong miền tri thức cho phép tri thức có thể được chia sẻ giữa các ứng dụng.



Hình 1-1: Ví dụ minh họa biểu diễn tri thức trong Ontology

1.1.2 Các thành phần của Ontology

Ontology bao gồm các thành phần: thực thể (Individuals), lớp (Classes), thuộc tính (Properties) và quan hệ (Relations).

➤ Thực thể

- Thực thể là các thành phần cơ bản của ontology. Các thực thể là các đối tượng cụ thể, ví dụ về số nguyên, phép toán, đồ thị, hàm số,...

- Ví dụ về các thực thể:

+ Số nguyên : -1,2,3,....

+ Phép toán : cộng, nhân, chia, ...

➤ Lớp

- Lớp là các nhóm hay tập hợp các đối tượng có chung một số thuộc tính. Lớp có thể bao gồm các lớp con, các thực thể hoặc bao gồm cả hai. Một lớp có thể gộp vào hoặc bị gộp vào các lớp khác. Một lớp gộp vào lớp khác được gọi là lớp con (subclass) của lớp gộp.

Ví dụ :

+ Lớp số thực có lớp con là lớp số nguyên.

+ Lớp đồ thị có lớp con là lớp đồ thị có hướng, lớp đồ thị vô hướng.

➤ Thuộc tính

- Thuộc tính được sử dụng để chỉ ra mối quan hệ giữa các thực thể (thuộc tính đối tượng) hoặc giữa thực thể và kiểu dữ liệu được mô tả (thuộc tính kiểu dữ liệu).

+ Ví dụ: Đồ thị có hướng

- Thuộc tính đối tượng "có hướng" có thể được sử dụng để chỉ ra mối quan hệ về hướng di chuyển giữa các đỉnh trong đồ thị có hướng. Ví dụ, trong một mạng lưới giao thông, thuộc tính "có hướng" có thể mô tả hướng đi của các con đường từ một điểm đến một điểm khác.
- Thuộc tính kiểu dữ liệu với một đồ thị có hướng có thể bao gồm các thuộc tính như tên đỉnh, trọng số của các cạnh, v.v. Giá trị của một thuộc tính dữ liệu có thể có các kiểu dữ liệu khác nhau như kiểu chuỗi, kiểu số... Ví dụ, một đỉnh có thể có thuộc tính "tên" để xác định tên của đỉnh đó, và một cạnh có thể có thuộc tính "trọng số" để biểu diễn trọng số của cạnh đó.

Trong trường hợp này, thuộc tính được sử dụng để mô tả các đặc điểm hoặc mối quan hệ giữa các thực thể trong đồ thị, bao gồm cả các thuộc tính đối tượng và thuộc tính kiểu dữ liệu.

➤ Mối quan hệ

- Mối quan hệ được sử dụng để xác định sự liên quan của đối tượng này với một đối tượng khác bằng các thuộc tính tương ứng. Cùng với đó là tập hợp các quan hệ mô tả ngữ nghĩa của miền như quan hệ đồng nghĩa, trái nghĩa và các quan hệ khác. Điểm mạnh của Ontology là khả năng mô tả các mối quan hệ. Tập hợp các kiểu quan hệ được sử dụng và hệ thống phân cấp của chúng mô tả sức mạnh biểu đạt của ngôn ngữ mà Ontology được thể hiện.

+ Ví dụ: Đồ thị không có hướng

- Mối quan hệ "kề cận" giữa các đỉnh trong một đồ thị có thể mô tả sự liên kết giữa các đỉnh mà không có hướng. Ví dụ, trong một đồ thị biểu diễn một mạng lưới xã hội, mối quan hệ "kề cận" có thể mô tả mối quan hệ bạn bè giữa các người dùng.

- Mỗi quan hệ "có thứ bậc" có thể được sử dụng để mô tả mức độ liên kết giữa các đỉnh trong một cây, ví dụ như cây genealogical. Ví dụ, mỗi quan hệ "có thứ bậc" giữa các nút trong một cây có thể mô tả mối quan hệ cha con.

1.1.3 Các bước xây dựng Ontology

Để xây dựng một Ontology chúng ta sẽ tham khảo quy trình gồm 7 bước do nhóm nghiên cứu của đại học Stanford đưa ra.

✓ Bước 1: Xác định lĩnh vực và phạm vi của Ontology

Bước đầu tiên sẽ xác định lĩnh vực và phạm vi của Ontology:

Lĩnh vực mà Ontology sẽ đề cập đến là gì?

Mục đích của việc xây dựng và ứng dụng của Ontology?

Ontology cần trả lời cho những câu hỏi truy vấn thông tin gì?

Đối tượng sử dụng và khai thác Ontology?

✓ Bước 2: Xem xét việc sử dụng lại các Ontology có sẵn

Xem xét các Ontology hiện có và kiểm tra xem có thể kế thừa và mở rộng hay không. Nhiều Ontology đã có sẵn có thể được kế thừa và nhập vào môi trường phát triển Ontology mà chúng ta sử dụng.

✓ Bước 3: Liệt kê các thuật ngữ quan trọng trong Ontology

Ở bước này chúng ta cần liệt kê danh sách các thuật ngữ muốn đề đưa ra và giải thích cho người dùng.

Các thuật ngữ chúng ta muốn đề cập đến là gì?

Các thuật ngữ đó có những thuộc tính gì?

Chúng ta muốn nói gì về những thuật ngữ đó.

Ví dụ: Các thuật ngữ quan trọng liên quan đến đồ thị sẽ bao gồm tên đồ thị, số đỉnh, số cạnh, có hướng hay vô hướng, trọng số, ... Ban đầu, điều quan trọng là có được một danh sách đầy đủ các thuật ngữ mà không phải lo về sự trùng lặp giữa các khái niệm mà chúng đại diện, mối quan hệ giữa các thuật ngữ hoặc bất kỳ thuộc tính nào mà các khái niệm có thể có, hoặc liệu các khái niệm có phải là lớp hay thuộc tính.

✓ Bước 4: Xác định các lớp và phân cấp của các lớp

Một số cách tiếp cận để phát triển các phân cấp của các lớp:

- Tiếp cận theo hướng từ trên xuống top-down: Quá trình phát triển từ trên xuống bắt đầu với việc xác định các khái niệm chung nhất trong lĩnh vực và sau đó sẽ tiếp tục cụ thể hóa các khái niệm. Ví dụ, chúng ta có thể bắt đầu với việc tạo các lớp cho các

khái niệm chung về toán rời rạc. Sau đó, chúng ta cụ thể hóa lớp toán rời rạc bằng cách tạo ra một số phân lớp của nó: Đồ thị, cạnh, đỉnh,....

- Tiếp cận theo hướng từ dưới lên bottom-up: Quá trình phát triển bắt đầu từ các lớp cụ thể nhất, sau đó thực hiện nhóm các lớp này thành lớp tổng quát hơn là các lớp cha. Ví dụ, chúng ta bắt đầu bằng cách xác định các lớp Đỉnh, Cạnh,.... Sau đó là lớp cha của các lớp Đỉnh, Cạnh,... là lớp Đồ thị
- Tiếp cận theo hướng kết hợp: Quá trình phát triển kết hợp là sự kết hợp của các phương pháp bottom-up và top-down: Trước tiên, chúng ta xác định xem các khái niệm nổi bật, sau đó khái quát hóa và chi tiết hóa chúng một cách thích hợp. Chúng ta có thể bắt đầu với một vài khái niệm cấp cao nhất và một vài khái niệm cụ thể. Sau đó, chúng ta có thể liên hệ chúng với một khái niệm cấp trung.

Không có phương pháp nào trong số ba phương pháp trên được khẳng định tốt hơn các phương pháp còn lại. Nếu nhà phát triển Ontology có xu hướng nghĩ về phân loại chung nhất trước tiên, thì cách tiếp cận từ trên xuống có thể phù hợp hơn. Nếu nhà phát triển muốn bắt đầu bằng cách dựa trên các ví dụ cụ thể, thì cách tiếp cận từ dưới lên có thể phù hợp hơn.

✓ Bước 5: Xác định các thuộc tính của các lớp

Sau khi xác định các lớp, chúng ta phải mô tả cấu trúc các khái niệm. Chúng ta đã chọn các lớp từ danh sách các thuật ngữ đã tạo ở Bước 3. Hầu hết các thuật ngữ còn lại có thể là thuộc tính của các lớp này. Các thuật ngữ này bao gồm, ví dụ: tên đồ thị, bậc đồ thị, số đỉnh, số cạnh, có hướng hay vô hướng, trọng số.

Đối với mỗi thuộc tính trong danh sách, chúng ta phải xác định nó mô tả lớp nào. Các thuộc tính này trở thành các thuộc tính gắn vào các lớp. Ví dụ thuộc tính “bậc đồ thị” gắn vào lớp “Đồ thị” nhưng không gắn vào lớp “Mệnh đề”.

Vì vậy, ngoài các thuộc tính chúng ta đã xác định trước đó, chúng ta cần thêm các thuộc tính sau trong lớp “Đồ thị”: tên đồ thị, bậc đồ thị, số đỉnh, số cạnh, có hướng hay vô hướng, trọng số, ...

Tất cả các lớp con của một lớp đều kế thừa thuộc tính của lớp đó do vậy một thuộc tính cần được đính kèm ở lớp chung nhất có thể có thuộc tính đó.

✓ Bước 6: Xác định giới hạn của các thuộc tính

Thuộc tính có thể có các khía cạnh khác nhau để mô tả loại giá trị, giá trị cho phép, số lượng giá trị (số lượng) và các đặc điểm khác của giá trị mà thuộc tính có thể nhận. Ví dụ: thuộc

tính “tên gọi” được giới hạn kiểu dữ liệu dạng chuỗi (string) còn thuộc tính “bậc của đỉnh” được giới hạn kiểu dữ liệu dạng số nguyên (integer).

Tập hợp thuộc tính

Tập hợp thuộc tính định nghĩa số lượng giá trị mà một thuộc tính có thể có. Một số hệ thống chỉ phân biệt giữa tập hợp đơn (cho phép nhiều nhất một giá trị) và tập hợp đa (cho phép bất kỳ số lượng giá trị nào).

Loại thuộc tính giá trị:

Một ràng buộc kiểu giá trị mô tả những loại giá trị nào có thể điền vào thuộc tính. Dưới đây là danh sách các loại giá trị phổ biến:

String là kiểu giá trị đơn giản nhất được sử dụng cho các thuộc tính, chẳng hạn như tên gọi một đồ thị.

Number (Float, Integer, ...) mô tả các thuộc tính có giá trị số. Ví dụ, bậc đồ thị có thể có kiểu giá trị integer.

Boolean là kiểu giá trị có - không. Ví dụ: nếu chọn “Bể bơi” là một trong những tiện ích để đánh giá khách sạn thì các khách sạn có thể “có” hoặc “không có” tiện ích này

Enumerated chỉ rõ một danh sách các giá trị được phép cho thuộc tính. Ví dụ thuộc tính “độ phức tạp” có thể nhận một trong ba giá trị có thể có: tốt, trung bình, kém.

Các thuộc tính kiểu Instance cho phép xác định mối quan hệ giữa các thực thể. Các thuộc tính có kiểu giá trị Instance cũng phải xác định danh sách các lớp được phép mà các thực thể thuộc vào.

Domain và range của một thuộc tính:

Các lớp được phép gán cho các thuộc tính thường được gọi là một phạm vi của một thuộc tính. Một số hệ thống cho phép giới hạn phạm vi của thuộc tính khi thuộc tính được gán cho một lớp cụ thể. Các lớp mà một thuộc tính được gán vào hoặc một lớp thuộc vùng mà thuộc tính mô tả, được gọi là miền của thuộc tính đó.

Các quy tắc cơ bản để xác định miền và phạm vi thuộc tính tương tự nhau: Khi xác định miền hoặc phạm vi cho thuộc tính, hãy tìm các lớp hoặc lớp chung nhất có thể tương ứng là miền hoặc phạm vi cho thuộc tính. Mặt khác, không xác định miền và phạm vi quá chung chung.

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính bao gồm một lớp và lớp con của nó, hãy xóa phạm vi và miền trên lớp con đó

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính chứa tất cả các lớp con của một lớp, nhưng không phải chính lớp đó, thì phạm vi chỉ nên chứa cả lớp cha chứ không cần chứa từng lớp con.

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính chứa tất cả trừ một vài lớp con của một lớp cha, hãy xem xét liệu lớp cha có thể đưa ra một định nghĩa phạm vi phù hợp hơn hay không

Trong các hệ thống mà việc gán một thuộc tính vào một lớp cũng giống như việc thêm lớp đó vào miền của thuộc tính: Một mặt, chúng ta nên cố gắng làm cho nó càng tổng quát càng tốt. Mặt khác, chúng ta phải đảm bảo rằng mỗi lớp mà chúng ta gán thuộc tính thực sự có thể có thuộc tính mà thuộc tính đó đại diện.

✓ Bước 7: Tạo các thể hiện / thực thể.

Bước cuối cùng là tạo các thực thể của các lớp trong hệ thống phân cấp. Việc xác định một thực thể riêng lẻ của một lớp yêu cầu: (1) chọn một lớp, (2) tạo một thực thể riêng của lớp đó và (3) điền vào các giá trị thuộc tính. Ví dụ: tạo một thực thể “Dạng mệnh đề” thuộc lớp “Cơ sở logic”. Thực thể này có các giá trị thuộc tính sau được xác định:

- Tên gọi: Dạng mệnh đề

- Định nghĩa: là một biểu thức được cấu tạo từ:

+ Các mệnh đề (các hằng mệnh đề)

+ Các biến mệnh đề p, q, r, \dots , tức là các biến lấy giá trị là các mệnh đề nào đó

+ Các phép toán $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ và dấu đóng mở ngoặc $()$.

+ Ví dụ: $E(p,q) = \neg(\neg p \vee q)$ $F(p,q,r) = (p \wedge q) \rightarrow \neg(q \vee r)$

1.2 Giới thiệu về mô hình COKB

1.2.1 Định nghĩa

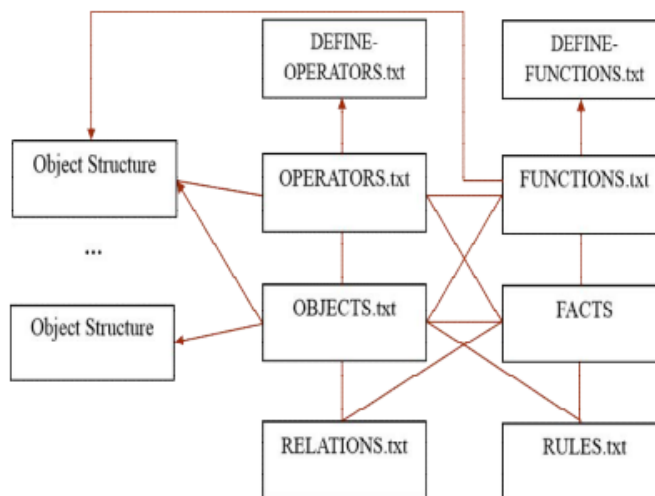
- Mô hình COKB là một cấu trúc gồm 5 thành phần tri thức sau:

$(C, R, Ops, Funcs, Rules)$

Trong đó:

- C là tập khái niệm các đối tượng tính toán thu hẹp (NCom-Objects).
- R là tập các quan hệ nhị phân trên các khái niệm.
- Ops là tập các toán tử 2 ngôi.
- $Funcs$ là tập các hàm
- $Rules$ là tập các luật suy diễn

Dưới đây là sơ đồ tổ chức cơ sở tri thức theo mô hình COKB



Hình 1-2: Sơ đồ tổ chức cơ sở tri thức theo mô hình COKB

1.2.2 Ngôn ngữ đặc tả cho mô hình COKB

- Ngôn ngữ đặc tả cho mô hình COKB được xây dựng từ các thành phần sau:

- Tập các ký tự: chữ, số và ký tự đặc biệt
- Từ vựng: từ khóa, tên
- Các kiểu dữ liệu: kiểu cơ sở và các kiểu cấu trúc.
- Biểu thức và câu, các câu lệnh
- Cú pháp đặc tả các thành phần của mô hình COKB

- Dưới đây, là một số cấu trúc định nghĩa cho các biểu thức, các đối tượng tính toán, các quan hệ tính toán, các sự kiện, và các hàm.

1.2.3 Thành phần tri thức hàm

- Trong mô hình COKB thì thành phần tri thức hàm chưa được làm rõ mà chỉ đưa ra đặc tả cơ bản về hai dạng hàm là: hàm cơ sở và hàm tự định nghĩa, chẳng hạn như các hàm lấy vế trái của một đẳng thức: lhs(eq), lấy giá trị tuyệt đối của một số: abs(a)... là các hàm cơ sở, còn các hàm kiểm tra một đỉnh có thuộc đồ thị hay không: hàm hasVertex(G,v), hay hàm thêm một cạnh vào đồ thị: hàm addEdge(G,e) là các hàm tự định nghĩa.

1.2.4 Các loại sự kiện trong mô hình COKB

Trong mô hình COKB, có tất cả 11 loại sự kiện được mô tả chi tiết ở bên dưới:

- **Sự kiện 1:** Thông tin về loại của đối tượng
- **Sự kiện 2:** Tính xác định của đối tượng hay thuộc tính của đối tượng

- **Sự kiện 3:** Tính xác định của đối tượng hay thuộc tính của đối tượng thông qua một biểu thức hằng
- **Sự kiện 4:** Sự bằng nhau của đối tượng hay thuộc tính của đối tượng với một đối tượng hay một thuộc tính khác
- **Sự kiện 5:** Sự phụ thuộc của một đối tượng với một đối tượng khác thông qua một biểu thức tính toán
- **Sự kiện 6:** Quan hệ của các đối tượng hay thuộc tính của đối tượng
- **Sự kiện 7:** Tính xác định của một hàm
- **Sự kiện 8:** Tính xác định của một hàm thông qua một biểu thức hằng
- **Sự kiện 9:** Sự bằng nhau của một đối tượng với một hàm
- **Sự kiện 10:** Sự bằng nhau giữa một hàm với một hàm khác
- **Sự kiện 11:** Sự phụ thuộc của một hàm với các hàm/đối tượng khác thông qua một công thức tính toán

Các sự kiện từ loại 7 đến loại 11 là các sự kiện liên quan đến thành phần tri thức hàm mà ta đã làm đề cập ở phần b. Vấn đề suy diễn và tìm kiếm, cũng như là tính toán trên thành phần hàm sẽ được làm rõ ở các mục tiếp theo.

1.3 Công nghệ và ngôn ngữ lập trình

1.3.1 React.js

- **Khái niệm:** ReactJS là một thư viện JavaScript phổ biến được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web. Được phát triển bởi Facebook, ReactJS cho phép các nhà phát triển xây dựng các thành phần giao diện tái sử dụng và tương tác dễ dàng.

Một số điểm nổi bật của ReactJS bao gồm:

- **Component-Based:** ReactJS sử dụng cấu trúc dựa trên thành phần, cho phép phân chia giao diện thành các thành phần nhỏ độc lập, dễ quản lý và tái sử dụng.

- **Virtual DOM:** ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Thay vì cập nhật trực tiếp DOM khi có thay đổi, React tạo ra một bản sao của DOM (Virtual DOM) và chỉ cập nhật những phần thay đổi, giúp giảm thiểu số lần render trên DOM thực sự.

- **JSX:** ReactJS sử dụng JSX (JavaScript XML) để viết các thành phần giao diện. JSX cho phép viết HTML trong JavaScript, làm cho mã nguồn trở nên dễ đọc và dễ hiểu hơn.

Unidirectional Data Flow: ReactJS thúc đẩy luồng dữ liệu một chiều, từ component cha đến component con. Điều này giúp dễ dàng theo dõi dữ liệu và làm cho ứng dụng dễ bảo trì hơn.

- **Ecosystem:** ReactJS có một hệ sinh thái phong phú với nhiều thư viện và công cụ hỗ trợ như Redux, React Router, Styled Components, và nhiều thứ khác để giúp phát triển ứng dụng hiệu quả hơn.

- ReactJS đã trở thành một trong những công nghệ phổ biến nhất cho phát triển ứng dụng web hiện đại nhờ vào tính linh hoạt, hiệu suất và cộng đồng hỗ trợ mạnh mẽ.

1.3.2 Spring boot

- **Khái niệm:** Spring Boot là một framework của Java được sử dụng để xây dựng ứng dụng web nhanh chóng và dễ dàng. Nó cung cấp một cách tiếp cận linh hoạt và mạnh mẽ để triển khai ứng dụng Java mà không cần nhiều cấu hình phức tạp.

- Spring Boot tích hợp sẵn các thư viện và cấu hình phổ biến để giảm bớt sự phức tạp trong việc triển khai ứng dụng Java, giúp nhà phát triển tập trung vào việc viết mã logic chính của ứng dụng mà không phải lo lắng về việc cấu hình và tích hợp các thành phần khác nhau. Điều này giúp tăng hiệu suất phát triển và giảm thiểu thời gian triển khai.

- **Spring Boot cung cấp nhiều tính năng và chức năng mạnh mẽ để phát triển ứng dụng web và dịch vụ. Dưới đây là một số chức năng quan trọng của Spring Boot:**

- **Tích hợp nhanh chóng:** Spring Boot giúp bạn bắt đầu với dự án mới một cách nhanh chóng bằng cách cung cấp các cấu hình mặc định hợp lý và tự động cấu hình môi trường cho ứng dụng.
- **Tự cấu hình:** Spring Boot sử dụng các chuẩn cấu hình mặc định để tự động cấu hình ứng dụng của bạn mà không cần phải thực hiện nhiều công việc cấu hình thủ công.
- **Dependency Injection (DI):** Spring Boot sử dụng Dependency Injection (DI) để quản lý và kết nối các thành phần của ứng dụng, giúp tạo ra mã dễ đọc, linh hoạt và dễ bảo trì.
- **Embedded Server:** Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty hoặc Undertow, giúp bạn triển khai ứng dụng một cách dễ dàng mà không cần cài đặt và cấu hình máy chủ bên ngoài.
- **Spring Data và ORM:** Spring Boot tích hợp các tính năng của Spring Data JPA và các công cụ ORM như Hibernate, giúp bạn dễ dàng thao tác với cơ sở dữ liệu.
- **Spring Security:** Spring Boot cung cấp tích hợp sẵn với Spring Security để bảo vệ ứng dụng của bạn bằng cách xác thực người dùng, quản lý phiên và kiểm soát quyền truy cập.
- **Spring MVC:** Spring Boot hỗ trợ Spring MVC để xây dựng các ứng dụng web RESTful và MVC dễ dàng.
- **Cấu hình bên ngoài:** Spring Boot hỗ trợ cấu hình bên ngoài thông qua các tệp cấu hình như application.properties hoặc application.yml, giúp bạn tinh chỉnh ứng dụng mà không cần phải thay đổi mã nguồn.

1.3.3 Lập trình ký hiệu(Symbolic)

- **Khái niệm:** Lập trình ký hiệu (symbolic programming) là một phương pháp lập trình mà trong đó các biểu thức và công thức được biểu diễn dưới dạng ký hiệu hoặc biểu thức toán học thay vì giá trị cụ thể. Các biểu thức này có thể chứa các biến, hàm, toán tử và các ký hiệu khác thay vì giá trị số hoặc chuỗi.

- **Chức năng của lập trình ký hiệu (Symbolic programming):**

- **Thực hiện tính toán kí hiệu:** Lập trình kí hiệu cho phép bạn thực hiện các phép tính toán trên biểu thức kí hiệu thay vì giá trị cụ thể. Điều này hữu ích trong các tác vụ như tích hợp, đạo hàm, tích phân và giải phương trình.
- **Xử lý dữ liệu kí hiệu:** Bằng cách sử dụng biểu thức kí hiệu, bạn có thể thực hiện các thao tác xử lý dữ liệu phức tạp mà không cần chuyển đổi dữ liệu thành dạng số học hoặc chuỗi.
- **Mô hình hóa vấn đề phức tạp:** Lập trình kí hiệu cho phép bạn mô hình hóa vấn đề theo cách mô tả và trừu tượng hóa chúng bằng cách sử dụng kí hiệu và biểu thức toán học, giúp dễ dàng hiểu và giải quyết vấn đề.
- **Tích hợp với hệ thống thông minh:** Trong lĩnh vực trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên, lập trình kí hiệu được sử dụng để biểu diễn và xử lý thông tin một cách linh hoạt và mạnh mẽ.
- **Phát triển phần mềm toán học:** Lập trình kí hiệu là một công cụ quan trọng trong việc phát triển các phần mềm toán học và khoa học, cho phép người dùng thực hiện các phép tính và phân tích phức tạp một cách hiệu quả.
- **Tính linh hoạt và tái sử dụng:** Biểu thức kí hiệu có thể được tái sử dụng trong nhiều ngữ cảnh khác nhau mà không cần phải thay đổi chúng, giúp tăng cường tính linh hoạt và tái sử dụng trong mã nguồn của bạn.

1.3.4 Ngôn ngữ Maple

- **Mối liên hệ giữa ngôn ngữ Maple với lập trình kí hiệu(symbolic programming):** Ngôn ngữ Maple có mối liên hệ sâu sắc với lập trình kí hiệu (symbolic programming). Đúng như tên gọi, lập trình kí hiệu là một phương pháp lập trình mà trong đó các biểu thức và công thức được biểu diễn dưới dạng kí hiệu hoặc biểu thức toán học thay vì giá trị cụ thể. Các biểu thức này có thể chứa các biến, hàm, toán tử và các kí hiệu khác thay vì giá trị số hoặc chuỗi.

- Maple được thiết kế và phát triển chủ yếu để hỗ trợ việc lập trình kí hiệu trong toán học và kỹ thuật. Các tính năng và chức năng của Maple hướng tới việc thực hiện tính toán và xử lý thông tin dưới dạng kí hiệu và biểu thức toán học, thay vì chỉ làm việc với giá trị cụ thể. Các chức năng của Maple đều ưu việt hơn lập trình Symbolic nhưng Maple không có tính linh hoạt và tái sử dụng, không có tính phổ quát, tính năng mở rộng.

1.3.5 Cơ sở dữ liệu (MySQL)

- **Khái niệm:** MySQL là một hệ quản trị cơ sở dữ liệu (Hệ quản trị cơ sở dữ liệu quan hệ - RDBMS) phổ biến và mạnh mẽ, được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp.

- **Những điểm lưu ý của MySQL:**

- **Mô hình quan hệ:** MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, có nghĩa là dữ liệu được tổ chức thành các bảng có mối quan hệ với nhau thông qua các khóa ngoại.
- **Ngôn ngữ truy vấn cấp cao (High-Level Query Language):** MySQL sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để tương tác và truy vấn dữ liệu từ cơ sở dữ liệu.

- **Tính năng đa nền tảng:** MySQL có sẵn cho nhiều hệ điều hành phổ biến như Linux, Windows và macOS, giúp cho việc triển khai trên nhiều môi trường một cách dễ dàng.
- **Tính bảo mật:** MySQL cung cấp các tính năng bảo mật như phân quyền người dùng, mã hóa dữ liệu và kiểm soát truy cập để bảo vệ dữ liệu.
- **Cơ chế sao lưu và phục hồi:** MySQL hỗ trợ các công cụ sao lưu và phục hồi dữ liệu, giúp đảm bảo an toàn và tin cậy cho dữ liệu.
- **Tính nhất quán và đồng nhất:** MySQL cung cấp các tính năng như giao thức ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính nhất quán và đồng nhất của dữ liệu.
- **Cộng đồng lớn và hỗ trợ rộng rãi:** MySQL có một cộng đồng lớn và nhiều nguồn tài liệu hỗ trợ, bao gồm tài liệu chính thức, diễn đàn và các tài liệu hướng dẫn từ các nguồn bên ngoài.
- **Phiên bản mã nguồn mở:** MySQL có hai phiên bản: Community Edition (phiên bản cộng đồng, miễn phí và mã nguồn mở) và Enterprise Edition (phiên bản doanh nghiệp, đi kèm với các tính năng và hỗ trợ cao cấp từ Oracle).

1.4 Hệ thống giải toán thông minh

Trong đề tài này chúng em sử dụng hệ thống giải toán thông minh ứng dụng trong trí tuệ nhân tạo như hệ luật dẫn. Ứng dụng lập trình symbolic giúp cho việc thiết kế các hệ thống này trở nên tiện lợi hơn, đồng thời chúng sẽ giải quyết được những bài toán có độ phức tạp cao, mang tính thực tiễn

1.4.1 Mô hình hệ luật dẫn

Phương pháp biểu diễn tri thức bằng luật đánh dấu được phát minh bởi Newell và Simon trong lúc họ đang cố gắng xây dựng một hệ giải bài toán tổng quát. Đây là một kiểu biểu diễn tri thức có cấu trúc bằng luật. Ý tưởng cơ bản là tri thức có thể được cấu trúc bằng một cặp điều kiện và hành động: "NẾU điều kiện xảy ra THÌ hành động sẽ được thực hiện". Chẳng hạn: NẾU đèn giao thông đỏ THÌ bạn không được đi thẳng, NẾU máy tính đã mở mà không khởi động được THÌ kiểm tra nguồn điện, vv...

Một cách tổng quát luật đánh dấu có dạng như sau:

$$P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$$

Tùy vào các vấn đề đang quan tâm mà luật đánh dấu có nhiều ý nghĩa hay cấu trúc khác nhau:

- Trong logic vị từ: P_1, P_2, \dots, P_n, Q là những biểu thức logic
- Trong ngôn ngữ lập trình, mỗi một luật sinh là một câu lệnh

$$\text{IF}(P_1 \text{ AND } P_2 \text{ AND } \dots \text{ AND } P_n) \text{ THEN } Q$$

- Trong lý thuyết hiểu ngôn ngữ tự nhiên, mỗi luật sinh là một phép dịch:

ONE \rightarrow một

TWO \rightarrow hai.

JANUARY \rightarrow tháng một.

Mô hình hệ luật dẫn gồm có hai thành phần như sau

(Facts, Rules)

Trong đó:

(1) Tập các sự kiện **Facts**:

$$F = \{f_1, f_2, \dots, f_n\}$$

(2) Tập các quy tắc **Rules** áp dụng trên các sự kiện có dạng như sau:

$$f_1 \wedge f_2 \wedge \dots \wedge f_i \rightarrow q \text{ trong đó, các } f_i, q \text{ đều thuộc } F$$

Ví dụ 1: 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):

$$R1: A \rightarrow C$$

$$R2: B \rightarrow D$$

$$R3: H \rightarrow A$$

$$R4: E \wedge G \rightarrow C$$

$$R5: E \wedge K \rightarrow B$$

$$R6: D \wedge E \wedge K \rightarrow C$$

$$R7: G \wedge K \wedge F \rightarrow A$$

1.4.2 Cơ chế suy luận trên hệ luật dẫn

Có 2 loại suy diễn là suy diễn tiến và suy diễn lùi. Tuy nhiên trong đề tài nghiên cứu khoa học này thì chúng em chỉ sử dụng suy diễn tiến

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.

Ví dụ: Với hệ luật dẫn trong ví dụ 1 trên ta có:

Sự kiện ban đầu: H, K

$$R3: H \rightarrow A \{A, H, K\}$$

$$R1: A \rightarrow E \{A, E, H, K\}$$

$$R5: E \wedge K \rightarrow B \{A, B, E, H, K\}$$

$$R2: B \rightarrow D \{A, B, D, E, H, K\}$$

$$R6: D \wedge E \wedge K \rightarrow C \{A, B, C, D, E, H, K\}$$

CHƯƠNG 2: MÔ HÌNH BÀI TOÁN

2.1 Mô hình bài toán

a. Định nghĩa

Trong mô hình COKB, một bài toán P được biểu diễn theo cấu trúc như sau:

$$P = (T, O, F) \rightarrow G$$

Trong đó: T là tập các tham số thực (nếu có); O là tập các NCom-Objects; F là tập các sự kiện giữa các NCom-Objects; bộ (T, O, F) là giả thiết của bài toán, được ký hiệu là H; G là mục tiêu của bài toán.

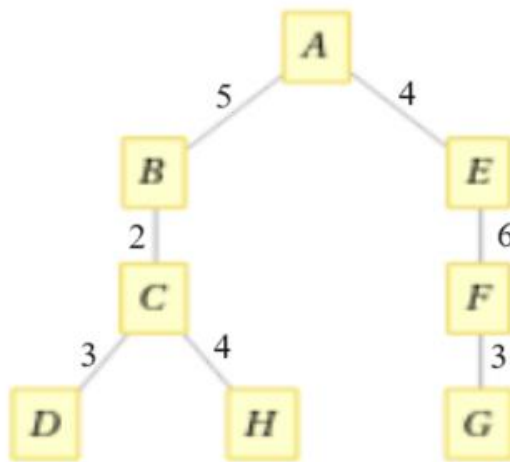
Mục tiêu G của bài toán có thể là:

- Xác định một hoặc nhiều đối tượng.
- Xác định một hoặc nhiều thuộc tính của đối tượng.
- Xác định quan hệ giữa các loại đối tượng
- Tính giá trị của một hàm thông qua các đối tượng/thuộc tính.
- Xác định giá trị của một hoặc nhiều

Để làm rõ các thành phần trong mô hình bài toán ở trên cũng như là cách mô hình một bài toán trong thực tế thực hiện như thế nào? Ta sẽ nêu một ví dụ cụ thể như ở phần dưới.

b. Ví dụ

Cho đồ thị có hình dạng như sau. Tìm cây khung nhỏ nhất của đồ thị



Hình 2-1: Mô tả đồ thị

Trong bài toán này, chúng ta có:

Tập hợp tham số (T):

Trong trường hợp này, không có tham số nào cụ thể được đề cập. Vì vậy, tập tham số (T) là rỗng.

Tập hợp các đối tượng (O):

Đối tượng được định nghĩa bao gồm các đỉnh của đồ thị và các cạnh của đồ thị:

- Các đỉnh: A, B, C, D, E, F, G, H.
- Các cạnh: [A, B], [B, C], [C, D], [A, E], [E, F], [F, G], [C, H].

Tập hợp các sự kiện giữa các đối tượng (F):

Các sự kiện được định nghĩa thông qua các trọng số của các cạnh:

- Trọng số của cạnh [A, B]: 5.
- Trọng số của cạnh [B, C]: 2.
- Trọng số của cạnh [C, D]: 3.
- Trọng số của cạnh [A, E]: 4.
- Trọng số của cạnh [E, F]: 6.
- Trọng số của cạnh [F, G]: 3.
- Trọng số của cạnh [C, H]: 4.

Mục tiêu của bài toán (G):

Tìm ra cây khung nhỏ nhất bằng cách sử dụng thuật toán Kruskal.

Để tiếp cận và giải quyết bài toán đồ thị mà ta đang xem xét, ta bắt đầu bằng việc định nghĩa các thành phần cơ bản nhất của đồ thị, đó là các đỉnh và các cạnh kèm theo trọng số của chúng. Đây là những giả thiết cơ bản và cần thiết để khởi động quá trình giải quyết vấn đề.

Tiếp theo, ta áp dụng một loạt các phép suy diễn, mỗi phép nhằm phát hiện và thêm vào các kết quả mới từ những dữ liệu có sẵn. Trong quá trình này, ta mở rộng liên tục tập hợp các biến đã được giải thích rõ, cho tới khi toàn bộ mục tiêu của bài toán được xác định.

2.2 Thuật giải tìm lời giải cho bài toán tổng quát

B1: Thu thập giả thiết và mục tiêu của bài toán

B2: Solution ← []; #khởi tạo lời giải rỗng

B3: Phân tích giả thiết

Kind Fact(); phân loại sự kiện

Unify Fact(); # hợp nhất sự kiện

B4: do{

Deduce Objects(); #Suy diễn trong đối tượng

```

    Deduce Funcs(); #Suy diễn trên hàm
  }
B5: if  $G \subseteq H$  then
    (Solution_found  $\leftarrow$  true; goto B8;)
  else
    Solution_found  $\leftarrow$  false;
B6: do{
    Hold H
    Chọn  $f \in F$ :
    while (not Solution found và (f found)) do(
      if (sinh sự kiện mới từ việc dùng f trong H) then (
         $H \leftarrow H \cup M(f)$ : #cập nhật lại giả thiết
        Đưa f vào tập lời giải Solution;
      ) #end if
      if( $G \subseteq H$ ) then (Solution_found true; goto B8;)
        Chọn sự kiện mới  $f \in F$ ;
      } // kết thúc vòng lặp while
    } while (Solution found hoặc ( $H = \text{Hold}$ ))
B7: if (not Solution found) then
    Kết luận: Không tìm thấy lời giải, kết thúc:
  else
B8: Lời giải của bài toán là: Solution:
    Thu gọn lời giải.
B9: Thể hiện lời giải

```

Quá trình suy diễn trên thành phần hàm tại bước 4 trong thuật toán ở trên khá phức tạp. Quá trình này được thể hiện ở thuật toán bên dưới:

2.4 Ứng dụng

Trong phần này, chúng em sẽ giới thiệu một ví dụ về việc ứng dụng vào bài toán lý thuyết đồ thị trong học phần toán rời rạc. Chương trình này được phát triển dựa trên nền tảng mô hình COKB và đã được lập trình bằng ngôn ngữ Maple. Các bước thiết kế chi tiết của chương trình sẽ được trình bày trong các mục sau.

2.4.1 Thiết kế hệ cơ sở tri thức

- Dựa trên mô hình COKB, cơ sở tri thức về lý thuyết đồ thị được tổ chức thành 2 thành phần như sau:
- + Tập C: gồm loại khái niệm như: “EDGE”, “GRAPH”, “UNDIRECTEDGRAPH” và “DIRECTEDGRAPH”.
- + Tập Funcs chứa các hàm như: thêm đỉnh/cạnh vào đồ thị, kiểm tra đỉnh/cạnh có thuộc đồ thị hay không,....
- Ví dụ về tập tin EDGE.txt

```
begin_object: EDGE[A, B];

A: char;
B: char;

begin_variables
  weight: integer;
end_variables

begin_construct_relations
end_construct_relations

begin_properties
  directed: boolean;
  EDGE[V1,V2] = EDGE[V2,V1];
end_properties

begin_computation_relations
end_computation_relations

begin_rules
end_rules

begin_functions
end_functions

end_object
```

2.4.2 Thiết kế động cơ suy diễn của hệ thống

Trong ứng dụng này chúng ta sẽ sử dụng chiến lược suy diễn tiến, được mô phỏng theo tư duy suy diễn diễn dịch của con người. Chúng ta có thể hiểu như sau: từ tập sự kiện đã biết, sử dụng các luật suy diễn trong miền tri thức để suy ra hay sinh ra các sự kiện mới, quá trình này tiếp diễn đến khi đạt được mục tiêu của bài toán hoặc là đến khi bị bế tắc. Do đó, quá trình suy luận diễn nhanh hơn, hiệu quả hơn.

CHƯƠNG 3: TRIỂN KHAI ỨNG DỤNG

3.1 Mục tiêu triển khai thử nghiệm

- Mục tiêu của việc triển khai thử nghiệm nghiên cứu ontology và mô hình tri thức các đối tượng tính toán COKB là xây dựng một hệ thống hỗ trợ giải toán thông minh, đặc biệt áp dụng vào học phần Toán rời rạc dành cho sinh viên Công nghệ thông tin tại Phân hiệu. Đây là một bước tiền quan trọng trong việc cải thiện chất lượng giáo dục và nâng cao hiệu suất học tập của sinh viên.
- Trong nghiên cứu này, việc sử dụng ontology và mô hình tri thức COKB nhằm tạo ra một cơ sở kiến thức cấu trúc và logic, giúp hệ thống hiểu và phân tích được bài toán, từ đó đề xuất các phương pháp giải quyết phù hợp. Điều này giúp cho sinh viên không chỉ có được cái nhìn tổng quan về bài toán mà còn được hỗ trợ trong quá trình suy luận và giải quyết vấn đề.
- Mục tiêu cuối cùng của dự án là tạo ra một hệ thống thông minh, có khả năng đề xuất và giải quyết các bài toán toán học rời rạc một cách hiệu quả, từ đó giúp sinh viên phát triển kỹ năng tư duy logic, khả năng suy luận và giải quyết vấn đề. Đồng thời, hệ thống cũng giúp giảng viên trong việc dạy và hỗ trợ sinh viên trong quá trình học tập, tạo ra môi trường học tập tích cực và sáng tạo.

3.2 Thiết kế ứng dụng

3.2.1 Thiết kế chức năng ứng dụng

- Chức năng xem lý thuyết toán rời rạc theo từng chương

Trang web hiển thị nội dung lý thuyết toán rời rạc theo từng chương, danh mục bài học cụ thể, từ đó người dùng có thể chọn vào mục lý thuyết cần xem và sẽ hiển thị các nội dung cần xem.

- Chức năng tìm kiếm lý thuyết theo từ khóa – Tìm kiếm theo chiều rộng

Người dùng gõ từ khóa cần tìm kiếm vào ô tìm kiếm (ví dụ: “mệnh đề”). Hệ thống sẽ chuyển từ khóa này thành câu lệnh sql để truy vấn cơ sở dữ liệu và hiển thị thông tin các kết quả có chứa từ khóa này.

- Chức năng tìm kiếm bằng chatbot – Tìm kiếm theo chiều sâu

Người dùng có thể tương tác với chatbot thông qua hộp thoại và hỏi những từ khóa về kiến thức họ đang cần tìm kiếm (Ví dụ: mệnh đề là gì?). Từ các từ khóa cung cấp, hệ thống sẽ gọi đến APIs mà chatbot đã được training và trả về kết quả. Kết quả sẽ trả về những kết quả tương tự với từ khóa và những phần lý thuyết có liên quan đến từ khóa đó.

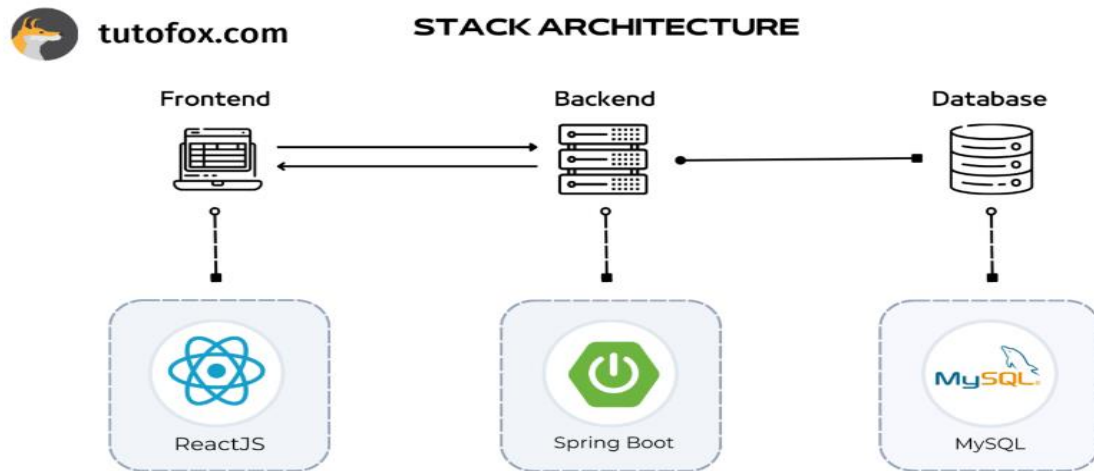
- Chức năng xem các bài viết, blogs

Người dùng có thể xem các bài viết, blogs nổi bật liên quan đến lý thuyết hoặc bài tập mà họ quan tâm

- Chức năng trao đổi học tập, và hỏi đáp cho sinh viên

Sinh viên có thể đặt câu hỏi thắc mắc về bất kỳ vấn đề nào, và thảo luận trên cộng đồng giữa các sinh viên với nhau

3.3.2 Thiết kế kiến trúc hệ thống



Hình 3-1: Mô tả thiết kế kiến trúc hệ thống

- Frontend (ReactJS): Trong phần này, ReactJS được sử dụng để xây dựng giao diện người dùng, quản lý các thành phần UI và trạng thái của ứng dụng.
- Backend (Spring Boot): Phần này của hệ thống là nơi triển khai các endpoint API RESTful để xử lý yêu cầu từ frontend. Nó cũng chịu trách nhiệm cho việc xác thực và phân quyền, xử lý logic nghiệp vụ, và tương tác với cơ sở dữ liệu.
- Giao tiếp giữa Frontend và Backend: Giao tiếp giữa frontend và backend được thực hiện thông qua các yêu cầu HTTP, ví dụ như GET, POST, PUT, DELETE để truy cập và cập nhật dữ liệu.
- Cơ sở dữ liệu: Dữ liệu được lưu trữ và quản lý trong cơ sở dữ liệu như MySQL. Backend sử dụng một lớp trung gian để tương tác với cơ sở dữ liệu, đảm bảo an toàn và hiệu suất.

3.4 Kết quả hiển thị

- Chức năng xem lý thuyết toán rời rạc theo từng chương

UTC2 - Discrete Math

[Home](#)
[Theory](#)
[Posts](#)
[Discussion Forum](#)
[About Us](#)
[Get in Touch](#)

Hi Admin

Enter your keyword?

Search

CHƯƠNG I: CƠ SỞ LOGIC

CHƯƠNG II: ĐẠI SỐ BOOLE

CHƯƠNG III: PHƯƠNG PHÁP ĐẾM

CHƯƠNG IV: QUAN HỆ

CHƯƠNG V: LÝ THUYẾT ĐỒ THỊ

KIẾN THỨC KHÁC

Bài 1: Mệnh đề

1. Định nghĩa:

Mệnh đề là một khẳng định có giá trị chân lý xác định, đúng hoặc sai.

Câu hỏi, câu cảm thán, mệnh lệnh... không là mệnh đề.

Ví dụ:

- Đại học CNTT trực thuộc ĐHQG TP.HCM.

Bài 1: Khái niệm đồ thị

Bài 2: Các loại đồ thị

Bài 3: Bậc của đồ thị

Bài 4: Biểu diễn đồ thị

Bài 5: Tính liên thông trong đồ thị

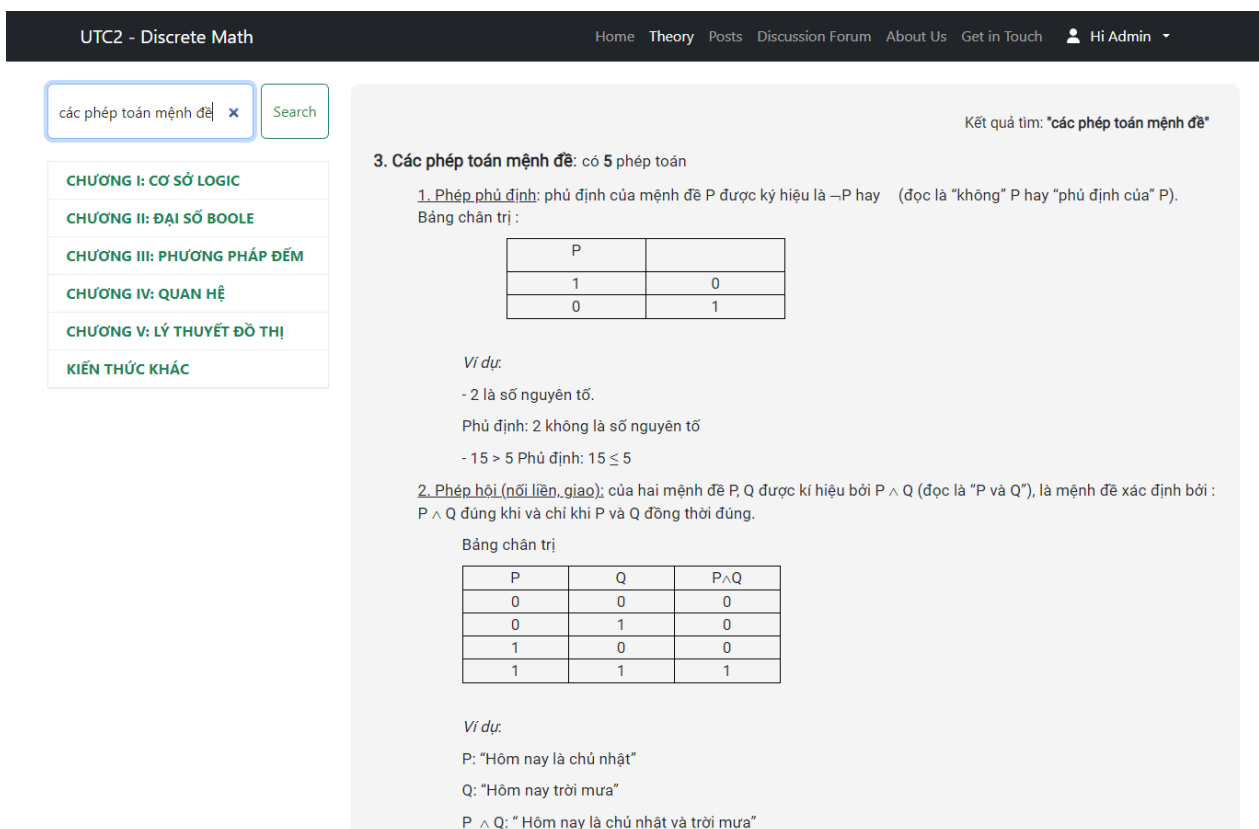
Bài 6: Chu trình - Chu trình Euler - Hamilton

Bài 7: Khung cây của đồ thị

Bài 8: Tìm đường đi ngắn nhất

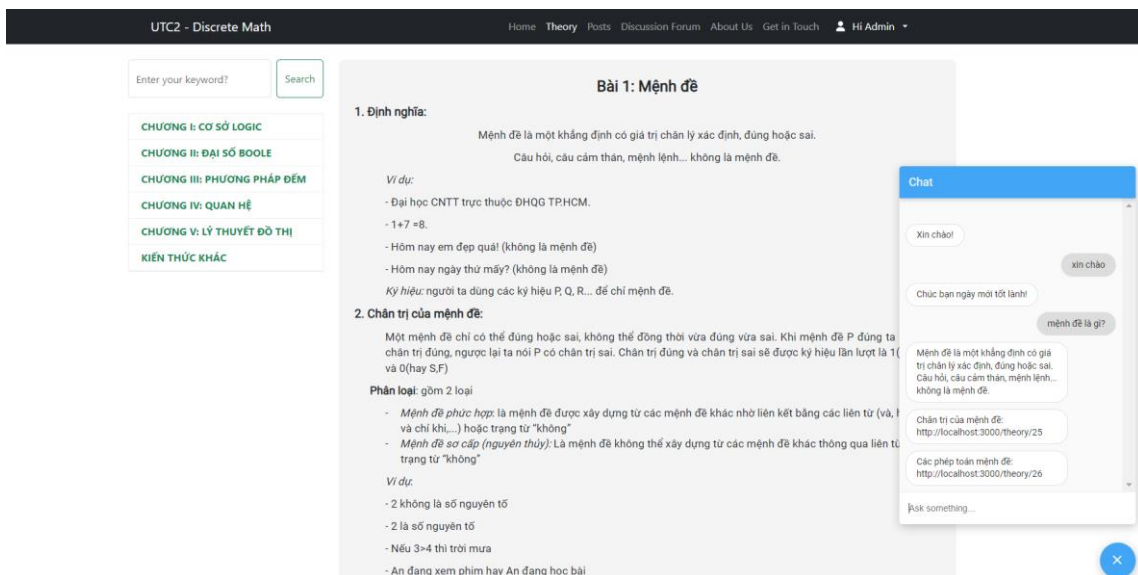
Hình 3-2: Mô tả tìm kiếm theo từng chương

- Chức năng tìm kiếm lý thuyết theo từ khóa – Tìm kiếm theo chiều rộng



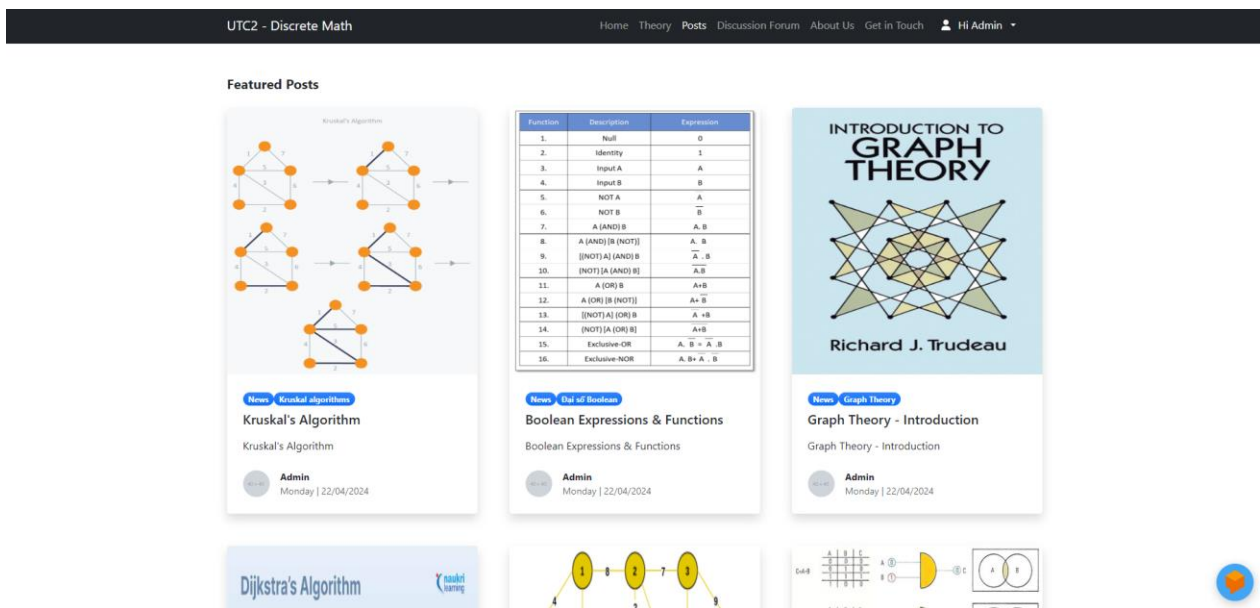
Hình 3-3: Mô tả tìm kiếm theo từ khóa - tìm kiếm theo chiều rộng

- Chức năng tìm kiếm bằng chatbot – Tìm kiếm theo chiều sâu

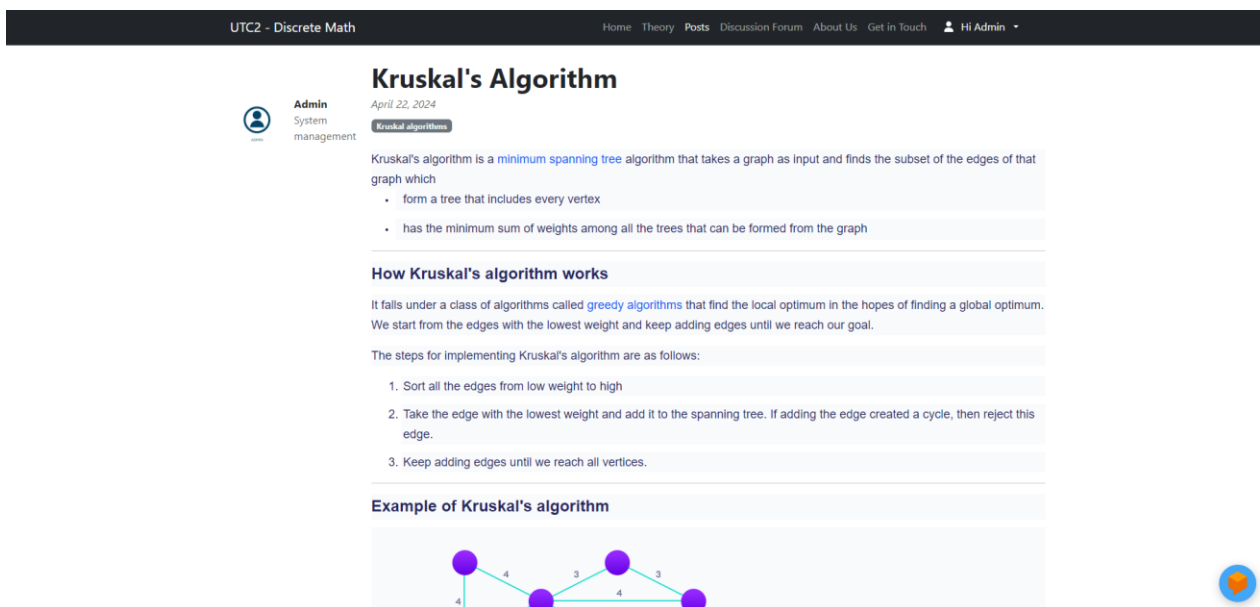


Hình 3-4: Mô tả tìm kiếm bằng chatbot- tìm kiếm chiều sâu

- Chức năng xem các bài viết, blogs

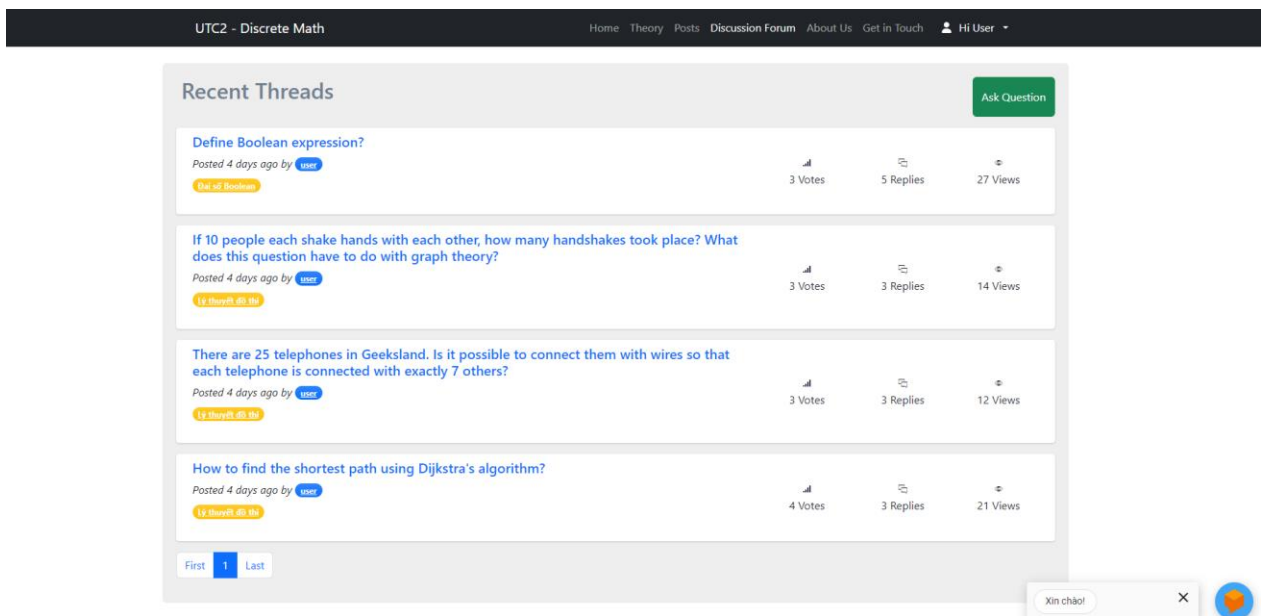


Hình 3-5: Mô tả chức năng xem bài viết

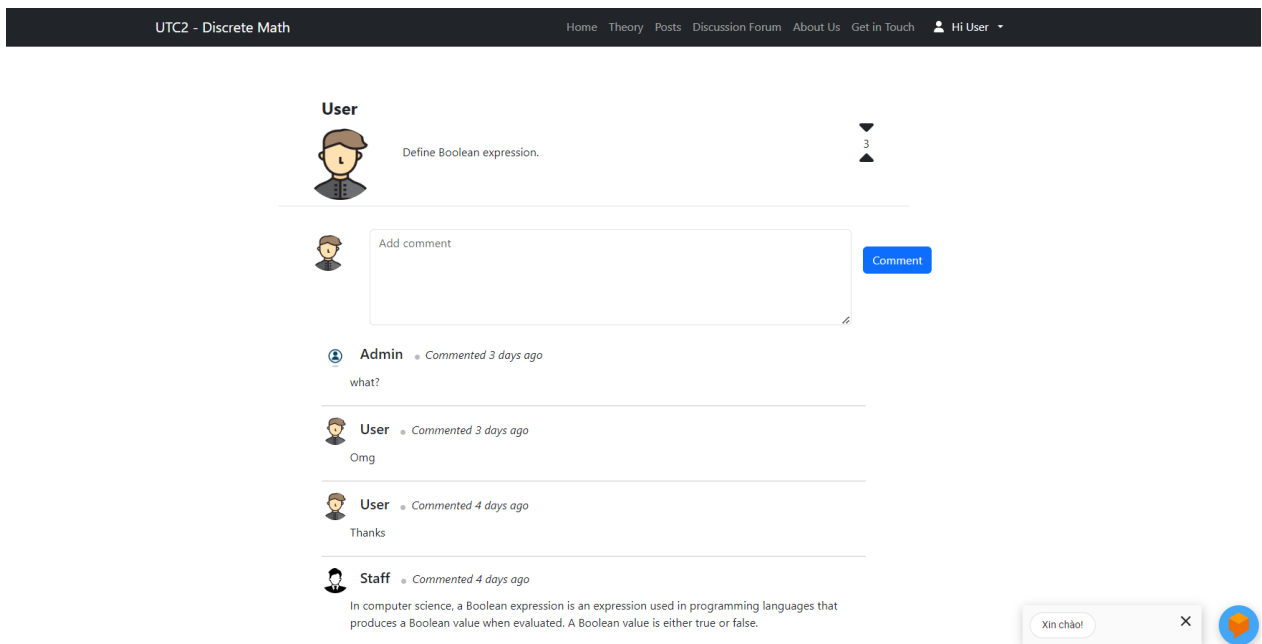


Hình 3-6: Mô tả nội dung của 1 bài viết

- Chức năng trao đổi học tập, và hỏi đáp cho sinh viên



Hình 3-7: Mô tả diễn đàn câu hỏi



Hình 3-8: Mô tả comment của người bình luận

KẾT LUẬN

1. Kết quả đạt được:

- Trong nghiên cứu này, chúng em đã khám phá và áp dụng ontology và mô hình tri thức các đối tượng tính toán COKB để xây dựng một hệ thống hỗ trợ giải toán thông minh, đặc biệt là trong học phần Toán Rời Rạc dành cho sinh viên Công nghệ Thông tin tại Phân hiệu.
- Ontology đã được chứng minh là một công cụ hữu ích trong việc mô hình hóa tri thức và mối quan hệ giữa các khái niệm trong một lĩnh vực cụ thể. Bằng cách xây dựng một ontology cho Toán Rời Rạc, chúng em đã có thể biểu diễn các khái niệm toán học và mối quan hệ giữa chúng một cách rõ ràng và có cấu trúc.
- Bằng cách kết hợp ontology và mô hình tri thức COKB, chúng em đã xây dựng một hệ thống hỗ trợ giải toán thông minh, giúp sinh viên hiểu và giải quyết các bài toán Toán Rời Rạc một cách hiệu quả hơn. Hệ thống này không chỉ cung cấp các tài nguyên học liệu và bài tập thực hành mà còn cung cấp khả năng hỗ trợ trực tiếp trong quá trình giải quyết bài toán.
- Kết hợp với React.js và Spring Boot chúng em đã thành công xây dựng một trang website đầy đủ các tính năng như tìm kiếm, đặt câu hỏi trên diễn đàn, coi các bài vlog trên trang website.
- Nhìn chung, nghiên cứu này đóng góp một phần nhỏ vào việc nâng cao chất lượng giáo dục toán học và cung cấp một công cụ hữu ích cho việc học tập và giảng dạy Toán Rời Rạc trong ngành Công nghệ Thông tin.

2. Hạn chế

- Mặc dù đề tài này mang lại nhiều lợi ích và tiềm năng trong việc cải thiện quá trình học và giảng dạy Toán Rời Rạc, nhưng cũng tồn tại một số hạn chế cần được lưu ý:
 - **Phụ thuộc vào chất lượng dữ liệu:** Hiệu suất của hệ thống phụ thuộc nhiều vào chất lượng của dữ liệu đầu vào và sự chính xác của ontology được xây dựng. Nếu dữ liệu không chính xác hoặc thiếu sót, có thể dẫn đến kết quả không chính xác. Vì vậy chúng ta cần xây dựng dữ liệu chất lượng nhất có thể
 - **Khả năng mở rộng:** Một trong những thách thức lớn của hệ thống này là khả năng mở rộng để bao gồm các khái niệm mới và cập nhật thông tin trong thời gian. Việc quản lý và duy trì ontology có thể trở nên phức tạp khi hệ thống mở rộng ra nhiều lĩnh vực và chủ đề khác nhau. Mở rộng càng nhiều thì độ chính xác của hệ thống càng giảm vì tập mẫu lớn dễ sai sót trong quá trình vận hành.
 - **Độ khó đặc tả tri thức :** khi đặc tả tri thức thì cần đảm bảo tính chính xác và đầy đủ của thông tin được biểu diễn. Việc hiểu rõ và phân loại các khái niệm, mối quan hệ và thuộc tính trong lĩnh vực cụ thể đòi hỏi sự kiên nhẫn và kiến thức sâu rộng về lĩnh vực đó.
 - **Khả năng triển khai và sử dụng thực tế:** Việc triển khai hệ thống và tích hợp vào môi trường học tập thực tế có thể đòi hỏi nhiều công sức và tài nguyên. Sự

hỗ trợ từ phía nhà trường và sự chấp nhận từ phía sinh viên cũng là yếu tố quan trọng để thành công của đề tài.

3. Hướng phát triển

- Trong tương lai, nghiên cứu về ontology và mô hình tri thức của các đối tượng tính toán COKB để xây dựng hệ thống hỗ trợ giải toán thông minh, sẽ mang lại nhiều lợi ích đối với cả giảng viên và sinh viên.
- Một hướng phát triển tiềm năng là mở rộng phạm vi áp dụng của hệ thống để bao gồm các môn học khác trong ngành Công nghệ Thông tin. Việc áp dụng ontology và mô hình tri thức vào các môn học khác như Cấu trúc dữ liệu và giải thuật, Lập trình logic, hoặc Cơ sở dữ liệu có thể giúp cung cấp các công cụ hỗ trợ học tập hiệu quả và thú vị cho sinh viên.
- Hơn nữa, việc tích hợp các công nghệ mới như trí tuệ nhân tạo (AI) và học máy (ML) vào hệ thống cũng là một hướng đi tiềm năng. Sử dụng các thuật toán máy học để phân tích và dự đoán nhu cầu học tập của sinh viên có thể giúp cá nhân hóa trải nghiệm học tập và cung cấp phản hồi cá nhân cho từng sinh viên, từ đó tăng cường hiệu suất học tập và sự tiến bộ của sinh viên.
- Ngoài ra, việc tăng cường sự tương tác giữa hệ thống và sinh viên cũng là một hướng đi quan trọng. Phát triển giao diện người dùng thân thiện và dễ sử dụng, cùng với việc cung cấp các tính năng tương tác như hỏi đáp tự động, đánh giá hiệu suất học tập và tạo ra các bài tập và câu hỏi tùy chỉnh có thể giúp tạo ra một môi trường học tập động, linh hoạt và thú vị.

Tóm lại, việc phát triển và mở rộng nghiên cứu về ontology và mô hình tri thức COKB để xây dựng hệ thống hỗ trợ giải toán thông minh trong học phần Toán Rời Rạc là một hướng đi tiềm năng, đem lại nhiều lợi ích cho việc giảng dạy và học tập trong ngành Công nghệ Thông tin tại các Phân hiệu.

TÀI LIỆU THAM KHẢO

- [1]. David Poole, Alan Mackworth (2023), Artificial Intelligence – Foundation of Computational Agents 3rd Edition, Cambridge University Press Publ, Cambridge, UK.
- [2] N. Do, H.D. Nguyen, T.T. Mai, “Intelligent Educational Software in Discrete Mathematics and Graph Theory”, Proceedings of 17th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2018), Granada, Spain, Frontiers in Artificial Intelligence and Applications, vol. 303, pp. 925 - 938, Sept. 2018.
- [3]. Đỗ, Văn Nhơn (2016), Biểu diễn tri thức và suy luận, Đại học Quốc Gia, TP.HCM
- [4]. Nguyễn, Đình Hiền & Hồ, Long Vân & Nguyễn, Thị Ngọc Diễm (2013), Tài liệu hướng dẫn thực hành Lập trình Symbolic trong trí tuệ nhân tạo, Đại học Quốc Gia, TP.HCM
- [5]. Đỗ, Văn Nhơn (2023), Bài giảng cao học chuyên đề Biểu diễn tri thức và ứng dụng, Trường Đại học Công nghệ thông tin, TP.HCM.
- [6]. Paul J. Deitel, Harvey Deitel(2017), Java How to Program, Early Objects, 11th Edition, Pearson, Cambridge, UK.
- [7]. Nhơn Van Do (2012), “Intelligent Problem Solvers in Education: Design Method and Applications”, Intelligent Systems, Prof. Vladimir M. Koleshko (Ed.), ISBN: 978-953-51-0054-6, InTech Publ, pp. 121-148.
- [8]. Kathy Sierra, Bert Bates, Trisha Gee (2022), “Head First Java – 3rd Edition”, O’Reilly Media, Inc., United States.
- [9]. Kirupa Chinnathambi (2018), A Hands-On Guide to Building Web Applications Using React and Redux, O’Reilly Media, Sebastopol, California, Hoa Kỳ
- [10]. LCF Publishing & Jamie Chan (2018), SQL: Learn SQL (using MySQL) in One Day and Learn It Well, Amazon Digital Services LLC – Kdp, United States
- [11]. Craig Walls(2016), Spring Boot in Action, Manning Publications, Shelter Island, New York, Hoa Kỳ.

[12]. Hien D. Nguyen, Linh Nguyen , Nha P. Tran , Van-Thanh Nguyen-Le, Sang Vu,
Design an Intelligent Problem Solver in Mathematics based on Integrated-Knowledge
Model

[13] Hien D. Nguyen , Nhon V. Do, Nha P. Tran , Xuan Hau Pham, and Vuong T. Pham
,Some Criteria of the Knowledge Representation Method for an Intelligent Problem
Solver in STEM Education