Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jsi.2017.002

XÂY DỰNG HỆ HỖ TRỢ GIẢI TOÁN ĐẠI SỐ TUYẾN TÍNH TRÊN CƠ SỞ TRI THỨC GỒM CÁC MIỀN TRI THỨC PHỐI HỢP

Nguyễn Đình Hiển, Đỗ Văn Nhơn và Phạm Thi Vương
Đại học Công nghệ Thông tin, Đại học Quốc gia – Hồ Chí Minh

Thông tin chung:
Ngày nhận bài: 15/09/2017
Ngày nhận bài sửa: 10/10/2017
Ngày duyệt đăng: 20/10/2017

Title:
Design an intelligent problem solver in linear algebra based on knowledge base included collaborative knowledge domains

Từ khóa:
Biểu diễn tri thức, công nghệ tri thức, hệ giải bài toán thông minh, suy diễn tự động.

Keywords:
Automated reasoning, intelligent problem solver, knowledge engineering, knowledge representation

ABSTRACT

Application knowledge representation for intelligent systems is a development trend in education, especially in science technology engineering and math education. In the mathematical foundation of higher education, linear algebra is an important course. This course includes the knowledge about matrices, linear equations systems, and vector spaces. In this paper, a method for representing the knowledge domain about linear algebra is proposed. It includes three sub-domains: matrices, linear equations systems, and vector spaces. Each domain is represented by model of computational objects knowledge base. These sub-domains have been researched to combine their knowledge for solving the classes of problems in linear algebra. Based on this knowledge base, an intelligent problem solver for this course in technical universities has been built. This program can solve common exercises. Its solutions are readable, step-by-step, and alike human method.

TÓM TẮT

Hiện nay, việc ứng dụng các phương pháp biểu diễn tri thức trong xây dựng các hệ thống giáo dục thông minh đang là một trong những xu thế phát triển, đặc biệt là trong giáo dục về STEM. Trong kiến thức toán cơ sở ở bậc đại học và cao đẳng, Đại số tuyến tính là một môn học rất quan trọng. Các kiến thức về ma trận, hệ phương trình tuyến tính và không gian vector là các kiến thức toán học nền tảng cho sinh viên. Trong bài báo này, chúng tôi sẽ nghiên cứu và đề xuất một mô hình biểu diễn tri thức Đại số tuyến tính. Miền tri thức này sẽ được phân thành ba miền tri thức: tri thức về ma trận, tri thức về hệ phương trình tuyến tính và tri thức về không gian vector. Trên cơ sở các miền tri thức này, chúng tôi nghiên cứu việc phối hợp các miền tri thức để giải quyết các lớp bài toán trong kiến thức về Đại số tuyến tính. Từ đó, chúng tôi xây dựng một hệ hỗ trợ giải toán tự động môn Đại số tuyến tính ở chương trình toán cao cấp bậc đại học cho khối ngành kỹ thuật. Chương trình có thể giải được các dạng bài tập thường gặp trong quá trình học. Lời giải chương trình rõ ràng, từng bước, tương tự như cách giải của con người.

Trích dẫn: Nguyễn Đình Hiển, Đỗ Văn Nhơn và Phạm Thi Vương, 2017. Xây dựng hệ hỗ trợ giải toán đại số tuyến tính trên cơ sở tri thức gồm các miền tri thức phối hợp. Tạp chí Khoa học Trường Đại học Cần Thơ. Số chuyên đề: Công nghệ thông tin: 10-18.

1 GIỚI THIỆU

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: frame-based, 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 (Computational Objects Knowledge Base - COKB) là một ontology được xây dựng theo tiếp cận hướng đối tượng (Do, 2010, 2015). Trong mô hình COKB, các đối tượng có cấu trúc và các hành vi nhất định. Đồng thời, mô hình COKB có thể biểu diễn các dạng tri thức khác nhau như quan hệ, toán tử, hàm. Mô hình COKB đã được ứng dụng trong các miền tri thức khác nhau, tuy nhiên mô hình COKB chưa thể giải quyết được vấn đề phối hợp giữa các miền tri thức trong một hệ thống.

Đại số tuyến tính là môn học đại cương nền tảng của nhiều trường đại học hiện nay. Môn học đề cập tới các kiến thức như ma trận, hệ phương trình tuyến tính, không gian vector (Anton and Rorres, 2010; Đỗ Công Khanh *và ctv.,* 2012). Hiện nay, đã có nhiều phần mềm hỗ trợ giải bài tập cho kiến thức này, tuy nhiên chúng đều không đáp ứng được yêu cầu cho một hệ thống hỗ trợ học tập:

Phần mềm Quickmath (2017) và MathSolver (2017) là các phần mềm giải được một số dạng toán về Đại số tuyến tính. Các chương trình này chỉ giải được các dạng toán cơ bản về ma trận như tính định thức, tìm ma trận nghịch đảo, chuyển vị, các phép toán đơn giản trên ma trận.

Bên cạnh đó, các hệ thống website hỗ trợ giải toán như: Mathway (2017), Symbolab (2017) có khả năng giải quyết các bài toán do người dùng nhập vào với lời giải từng bước, tự nhiên. Tuy nhiên, tri thức trong các hệ thống này chủ yếu được đặc tả theo dạng frame, do đó hệ thống chỉ có thể giải được các bài toán đơn giản, không giải được các bài toán đòi hỏi phải vận dụng các kiến thức chuyên sâu của tri thức.

Trong bài báo này, chúng tôi sẽ trình bày một phương pháp biểu diễn cho dạng tri thức gồm nhiều miền kiến thức, trong đó mỗi miền kiến thức được đặc tả theo mô hình COKB. Tri thức Đại số tuyến tính bao gồm các miền tri thức về ma trận ($\mathcal{K}_1$), tri thức về hệ phương trình tuyến tính ($\mathcal{K}_2$) và tri thức về không gian vector ($\mathcal{K}_3$). Các miền tri thức này sẽ được phối hợp với nhau trong tổng thể tri thức Đại số tuyến tính để giải quyết các lớp bài toán trong kiến thức Đại số tuyến tính, đặc biệt là các bài toán trong quá trình giải cần phải sử dụng kiến thức từ nhiều miền kiến thức. Trên cơ sở đó, tri thức Đại số tuyến tính và các lớp bài toán được đặc tả để thiết kế hệ hỗ trợ giải bài tập tự động cho môn học. Chương trình cho lời giải từng bước và tương tự như cách giải của người học.

2 THIẾT KẾ CƠ SỞ TRI THỨC ĐẠI SỐ TUYẾN TÍNH

2.1 Mô hình tri thức đối tượng tính toán

Mô hình tri thức các đối tượng tính toán (Computational Object Knowledge Base - COKB) là một bộ gồm 6 thành phần:

$\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules)

Trong đó, C là tập hợp các khái niệm, H là tập các quan hệ *is–a* giữa các khái niệm trong C, và R là tập các quan hệ khác. Ops là tập hợp các toán tử giữa các khái niệm trong C. Funcs là tập hợp các hàm. Rules là tập các luật của tri thức (Do, 2015).

Mỗi khái niệm c $\in$ C là một lớp các đối tượng, có tập thể hiện I_c, và khái niệm c có cấu trúc: (*Attr, Facts, EqObj, RulObj*). Trong đó, *Attr* là tập các thuộc tính của khái niệm c, *Facts* là tập các sự kiện nội tại giữa các thuộc tính trong khái niệm, *EqObj* là các quan hệ dưới dạng đẳng thức giữa các thuộc tính và *RulObj* là các luật dẫn nội tại của khái niệm. Mỗi đối tượng trong c có các hành vi để giải quyết các lớp vấn đề bên trong của nó.

Tập Rules được phân thành các tập luật:

Rules = $\text{Rule}_{\text{deduce}} \cup \text{Rule}_{\text{generate}} \cup$

$\text{Rule}_{\text{equivalent}} \cup \text{Rule}_{\text{equation}}$

- r $\in \text{Rule}_{\text{deduce}}$: là một luật dẫn, có dạng: $u(r) \Rightarrow v(r)$ với u(r), v(r) là các tập sự kiện

- r $\in \text{Rule}_{\text{generate}}$ là luật dẫn phát sinh một đối tượng mới, có dạng: $u(r) \rightarrow v(r)$

với u(r), v(r) là các tập sự kiện và thỏa điều kiện: $\exists o, o \in v(r)$ và $o \notin u(r)$

- r $\in \text{Rule}_{\text{equivalent}}$ là luật tương đương, có dạng: $h(r), u(r) \leftrightarrow v(r)$

với h(r), u(r), v(r) là các tập sự kiện và thỏa điều kiện: $h(r) \sqcup u(r) \rightarrow v(r)$, và $h(r) \sqcup v(r) \rightarrow u(r)$ đều đúng

- r $\in \text{Rule}_{\text{equation}}$ là một luật dạng đẳng thức, có dạng: $g(o_1, o_2, \ldots, o_k) = h(x_1, x_2, \ldots, x_p)$

với o_i, x_i là các đối tượng và g, h là các biểu thức giữa các đối tượng.

Tri thức $\mathcal{K}$ được mô hình theo dạng COKB có thể được thu gọn một số thành phần, chẳng hạn như: mô hình COKB chỉ có các tri thức dạng quan hệ (C, H, R, Rules), mô hình COKB gồm các tri thức quan hệ và toán tử (C, R, Ops, Rules) (Nguyễn Đình Hiển và Đỗ Văn Nhơn, 2014), mô hình COKB gồm các tri thức dạng hàm và toán tử (C, R, Ops, Funcs, Rules) (Do *et al.* 2015). Thông qua việc nghiên cứu và phân tích miền tri thức trong thực tế, chúng ta có thể xác định được dạng mô hình cần thiết để biểu diễn cho miền tri thức đó.

2.2 Cơ sở tri thức Đại số tuyến tính

Trong thực tế, tri thức $\mathcal{K}$ có thể được phân thành các miền tri thức con $\mathcal{K}_i$ (i=1,2,3…) với cấu trúc mỗi miền tri thức được đặc tả theo mô hình COKB. Mỗi miền tri thức $\mathcal{K}_i$ có thể được biểu diễn bằng mô hình $M(\mathcal{K}_i)$. Bên cạnh đó, giữa các miền tri thức con này có các mối quan hệ lẫn nhau, do đó các mô hình $\{M(\mathcal{K}_i)\}$ cùng với các quan hệ giữa chúng được đặc tả hình thành nên mô hình $M(\mathcal{K})$ biểu diễn cho tri thức $\mathcal{K}$ trong thực tế.

Đối với miền tri thức Đại số tuyến tính trong chương trình Toán cao cấp bậc Đại học, tri thức này sẽ được phân thành ba miền tri thức con: tri thức về ma trận ($\mathcal{K}_1$), tri thức về hệ phương trình tuyến tính ($\mathcal{K}_2$) và tri thức về không gian vector ($\mathcal{K}_3$). Mỗi miền tri thức này có thể được biểu diễn theo dạng mô hình COKB như sau:

2.2.1 $\mathcal{K}_1$ – Cơ sở tri thức ma trận:

$\mathcal{K}_1 = (C_1, H_1, R_1, Ops_1, Funcs_1, Rules_1)$

C₁ - là tập các khái niệm về ma trận và vector

C_1 = {MATRAN, MATRANVUONG, MATRANCHEO, VECTOR,...}

Ví dụ 2.1.1: Cấu trúc khái niệm MATRAN

Attr := {m, n, a[m][n], rank}

m: $\mathbb{N}$ // số dòng

n: $\mathbb{N}$ // số cột

rank: $\mathbb{N}$ // hạng ma trận

a[m][n]: $\mathbb{R}$ // Mảng 2 chiều giá trị các phần tử của ma trận

F := $\varnothing$ Eqbj := $\varnothing$

RulObj :={ r1: {m = n} $\rightarrow$ {this: MATRANVUONG}

Cấu trúc khái niệm:

MATRANVUONG::MATRAN (Ma trận vuông là một ma trận)

Att := MATRAN.Attr $\cup$ {inv, diag, det, dx}

diag: Boolean // chéo hóa được

inv: Boolean // tính khả nghịch

det: R // định thức

dx: Boolean // tính đối xứng

Facts:= MATRAN.Facts $\cup$ { m = n, dx= 0}

EqObj= MATRAN.EqObj

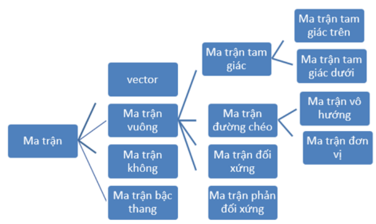
RulObj:= MATRAN.RulObj $\cup$

{ r1: $\forall$ i, j, $1 \le i \le n$, $1 \le j \le n$: a[i][j] = a[j][i] $\rightarrow$ dx = 1

r2: det # 0 $\rightarrow$ inv = 1

r3: $\forall$ i, j, $1 \le i < j \le n$: a[i][j] = 0 $\rightarrow$ this: Ma trận tam giác trên}

H₁ – các quan hệ is-a giữa các khái niệm ma trận và vector.



Hình 1: Quan hệ is-a giữa các khái niệm ma trận và vector

R₁ – Các quan hệ giữa các khái niệm trong **C₁**

R_1 = {Bằng nhau, Tương đương dòng, Tương đương cột, vector riêng, trị riêng}

Ví dụ 2.1.2:

+ *Bằng nhau* (=) $\subseteq I_{MATRAN} \times I_{MATRAN}$: Quan hệ bằng nhau giữa hai ma trận

* Tính chất: phản xạ, đối xứng, bắc cầu.

+ *Tương đương dòng* $\subseteq I_{MATRAN} \times I_{MATRAN}$: Quan hệ tương đương dòng giữa hai ma trận

Ops₁ – Các toán tử giữa các khái niệm ma trận và vector: $Ops_1 = O^1 \cup O^2$

Trong đó: O^1 là các toán tử một ngôi như các phép toán chuyển vị (^{T}) trên ma trận, phép toán nghịch đảo trên ma trận vuông ($^{-1}$)

O^2 là tập các phép toán 2 ngôi trên khái niệm ma trận và vector như: các phép toán cộng, trừ, nhân hai ma trận, hai vector.

$Funcs_1$ – Tập các hàm biến đổi sơ cấp trên ma trận

$Funcs_1$ = {Hoanvidong, NhanDong,

ThayTheDong,…}

Ví dụ 2.1.3:

Hoanvidong: $I_{MATRAN} \times \mathbb{N} \times \mathbb{N} \rightarrow I_{MATRAN}$

$(A, i, j) \mapsto B$

Ma trận B được tạo thành từ việc hoán vị dòng i và dòng j của ma trận A

ThayTheDong: $I_{MATRAN} \times \mathbb{N} \times \mathbb{R} \times \mathbb{N} \rightarrow I_{MATRAN}$ $(A, i, k, j) \mapsto B$

Ma trận B đươc tạo thành từ việc thay thế dòng i của A bằng giá trị của dòng i cộng với k lần dòng j ($k \neq 0$)

$Rules_1$ – Tập các luật cuả tri thức ma trận

Các luật dạng luật dẫn - $Rule_{deduce}$:

Rule 1.1: {A:MATRANVUONG, $\exists k \in \mathbb{R}$, $\exists x$, $\exists y$, $x \neq y$, $\forall j$: A.a[x][j] = k*A.a[y][j]}

$\rightarrow$ {A.det= 0}

+ Các luật phát sinh đối tượng - $Rule_{generate}$:

Rule 1.2: {A: MATRANVUONG, A.diag = 1}

$\rightarrow$ $\exists$D: MATRANCHEO, S: MATRANVUONG, S.inv = 1, D.n = S.n = A.n:

$A = S^{-1}.D.S$

Các luật tương đương - $Rule_{equivalent}$:

Rule 1.3: {A: MATRAN, B: MATRAN}

A tương đương dòng D $\leftrightarrow \exists[f_1,..f_n]$: dãy các phép biến đổi sơ cấp dòng, $f_n(\ldots(f_1(A))\ldots) = B$

Các luật dạng đẳng thức - $Rule_{equation}$:

Rule 1.4: A, B: MATRAN, A.n = B.m,

$$(A.B)^T = B^T.A^T$$

$\mathcal{K}_2$ – Cơ sở tri thức hệ phương trình tuyến tính

Tri thức hệ phương trình tuyến bậc nhất được biểu diễn bằng bộ gồm 4 thành phần:

$\mathcal{K}_2 = (C_2, H_2, R_2, Rules_2)$

C_2 - Tập các khái niệm về phương trình, hệ phương trình tuyến tính

C_2 = {PHUONGTRINH, HEPHUONGTRINH, HECRAMER}

Ví dụ 2.2.1: Cấu trúc khái niệm HEPHUONGTRINH:

Attr = {m, n, pt[m], Nghiem, Matranheso, Matranbosung**}**

m: $\mathbb{N}$ // số phương trình

n: $\mathbb{N}$ // số ẩn

pt[m]: PHUONGTRINH // Dãy các

phương trình

Nghiem:=

$$\left\{(b_1,...b_n) \in \mathbb{R}^n \mid \forall i = \overline{1,m}: \sum_{j=1}^{n} pt[i].a[j]*b_j = pt[i].a[n+1]\right\}$$

Matranheso: MATRAN [m,n]

Matranbosung: MATRAN [m, n+1]

Facts:={ $\forall$ i , pt[i].n = n}

EqObj:={$\forall$ i, j, $1 \le i \le m$, $1 \le j \le n$:

Matranheso[i , j] = pt[i].a …..}

RulObj:={

r1: Matranheso.rank = Matranbosung.rank= n

$\rightarrow$ card(Nghiem) = 1 …..}

H_2 – các quan hệ is-a giữa các khái niệm hệ phương trình

R_2 – Các quan hệ giữa các khái niệm trong **C_2**

R_2 = {Tương đương}

Tương đương ($\Leftrightarrow$) $\subseteq I_{HEPHUONGTRINH} \times I_{HEPHUONGTRINH}$: Quan hệ tương đương giữa hai hệ phương trình tuyến tính bậc nhất.

* Tính chất: phản xạ, đối xứng, bắc cầu.

$Rules_2$ – Tập các luật cuả tri thức về hệ phương trình tuyến tính bậc nhất

Các luật tương đương - $Rule_{equivalent}$:

Rule 2.1**:** A, B: HEPHUONGTRINH

A tương đương B $\leftrightarrow$ A.Nghiem = B.Nghiem

Rule 2.2: A, B: HEPHUONGTRINH

A tương đương B $\leftrightarrow$ (A.Matranbosung *tương đương dòng* B.Matranbosung)

Tri thức $\mathcal{K}_1$ và $\mathcal{K}_2$ liên hệ với nhau thông qua sự biến đổi tương đương giữa tập nghiệm của hai hệ phương trình và tương đương giữa hai ma trận bổ sung biểu diễn cho hai hệ phương trình tương ứng. Sự biến đổi này sẽ chuyển lớp bài toán trên hệ phương trình tuyến tính bậc nhất ($\mathcal{K}_2$) thành bài toán trên miền tri thức về ma trận ($\mathcal{K}_1$).

2.2.2 $\mathcal{K}_3$ – Cơ sở tri thức không gian vector

Tri thức không gian vector trong bài báo này được hiểu là các không gian vector cảm sinh trong không gian $\mathbb{R}^n$. Miền tri thức được biểu diễn bằng bộ gồm 5 thành phần:

$\mathcal{K}_3$ = (C_3, R_3, Ops_3, $Funcs_3$, $Rules_3$)

C_3 - Tập các khái niệm về không gian vector

C_3 = {VECTOR, KG_VECTOR}

Ví dụ 2.3.1: Cấu trúc khái niệm KG_VECTOR (không gian vector):

Attr := {dim, L}

dim: $\mathbb{N}$ // số chiều

$L \subseteq I_{VECTOR}$

Facts:= $\varnothing$ EqObj:= $\varnothing$

RulObj:={

r1: $\{\forall u, v \in L, \forall k \in \mathbb{R}\} \rightarrow \{ku + v \in L\}$

r2: $\{\forall u \in L\} \rightarrow \exists\, v \in L : u + v = 0\}$ }

R_3 – Các quan hệ giữa các khái niệm trong C_3

R_3 = {Thuộc, không gian con, Cơ sở,

Tập sinh, Độc lập tuyến tính,… }

Ví dụ 2.3.2:

Không gian con ($\leq$) $\subseteq I_{KG\_VECTOR} \times I_{KG\_VECTOR}$: Quan hệ không gian con giữa hai không gian vector.

Cơ sở $\subseteq I_{2^{I_{VECTOR}}} \times I_{KG\_VECTOR}$: Quan hệ một tập vector là cơ sở của một không gian vector.

Độc lập tuyến tính $\in I^k_{VECTOR}$: Quan hệ độc lập tuyến tính giữa k vector.

Tập sinh $\subseteq I_{2^{I_{VECTOR}}} \times I_{KG\_VECTOR}$: Quan hệ một tập vector là tập sinh của không gian vector

Ops_3 – Toán tử giữa hai vactor và hai không gian vector: $Ops_3 = Ops_{VECTOR} \cup \{\oplus\}$

Với Ops_{VECTOR} là tập các phép toán giữa hai vector và $\oplus$ là phép toán tổng trực tiếp của hai không gian vector con. Ta có định nghĩa của phép toán $\oplus$ như sau:

F_1, F_2, V: KG_VECTOR

$_1 \leq V, F_2 \leq V, F_1.L \cap F_2.L = \varnothing$

$\oplus: F_1 \times F_2 \rightarrow V$

$(x_1, x_2) \longmapsto x = x_1 + x_2$

$Funcs_3$ – Tập các hàm về tọa độ trong không gian vector

$Funcs_3$ = {Matrantoado, Toado}

Với *Matrantoado* là họ hàm xác định ma trận chuyển tọa độ trong không gian vector V và *Toado* là họ hàm xác định tọa độ của một vector trong không gian vector V. Ta có định nghĩa của các hàm như sau:

V: KG_VECTOR

Matrantoado$_V$: $2^{I_{VECTOR}} \times 2^{I_{VECTOR}} \rightarrow I_{MATRANVUONG}$

(B1, B2) $\longmapsto$ M

Trong không gian vector V, hàm xác định ma trận chuyển tọa độ theo cơ sở B1 sang cơ sở B2

Toado$_V$: $I_{VECTOR} \times 2^{I_{VECTOR}} \rightarrow I_{VECTOR}$

(v, B) $\longmapsto$ v’

Trong không gian vector V, hàm xác định tọa độ vector v theo cơ sở B.

$Rules_3$ – Tập các luật cuả tri thức không gian vector

+ Các luật dạng luật dẫn - $Rule_{deduce}$:

Rule 3.1: {B: $2^{I_{VECTOR}}$,V: KG_VECTOR, B cơ sở V} $\rightarrow$ V.dim = |B|

+ Các luật tương đương - $Rule_{equivalent}$:

Rule 3.2: {W, V: KG_VECTOR}

$W \leq V \leftrightarrow W.L \subset V.L$

Rule 3.3: {S: $2^{I_{VECTOR}}$, S = {$e_1, e_2, \ldots, e_k$} }

S độc lập tuyến tính $\leftrightarrow$ ({$a_1e_1 + \ldots + a_ke_k = 0$}

$\leftrightarrow$ {$a_1 = a_2 = \ldots = a_m = 0$})

Rule 3.4: {V: KG_VECTOR, B: $2^{I_{VECTOR}}$, B = {$e_1, e_2, \ldots, e_{V.dim}$} }

B cơ sở V $\leftrightarrow$ (B độc lập tuyến tính) AND

(B tập sinh V)

\+ Các luật dạng đẳng thức - $\text{Rule}_{\text{equation}}$:

Rule 3.5**:** {V: KG_VECTOR, B1, B2: $2^{I_{\text{VECTOR}}}$, B1 cơ sở V, B2 cơ sở V}

$$\text{Matrantoado}_V(B_2, B_1) = \text{Matrantoado}_V(B_1, B_2)^{-1}$$

Rule 3.6: {V: KG_VECTOR, *v*: VECTOR,

B1, B2: $2^{I_{\text{VECTOR}}}$, B1 cơ sở V, B2 cơ sở V }

$$\text{Toado}_V(v, B_2) =$$

$$\text{Matrantoado}_V(B_2, B_1).\text{Toado}_V(v, B_1)$$

Từ kết quả "Tập hợp tất cả các nghiệm của một hệ phương trình tuyến tính bậc nhất theo n ẩn số là một không gian vector con của $\mathbb{R}^n$", ta có tri thức $\mathcal{K}_3$ liên hệ với tri thức $\mathcal{K}_2$ thông qua việc tìm tập sinh của một không gian vector và chứng minh sự độc lập tuyến tính của một hệ vector trong tri thức $\mathcal{K}_3$ bằng cách đưa về việc giải một hệ phương trình trong miền tri thức $\mathcal{K}_2$.

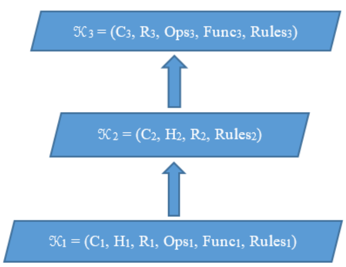
2.2.3 Mô hình tri thức đại số tuyến tính:

Cơ sở tri thức Đại số tuyến tính được đặc tả gồm 2 thành phần: **(K, Connect)**

Trong đó, **K** = {$\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$} là tập các miền tri thức con của đại số tuyến tính.

Connet: là tập các luật liên kết giữa các miền tri thức.

Tri thức K_2 có thể được chuyển về tri thức K_1 thông qua luật rule 2.2 và tri thức K_3 được chuyển về tri thức K_2 thông qua các luật rule 3.3 và rule 3.4.



Hình 2: Sơ đồ liên hệ giữa các miền tri thức $\mathcal{K}_1$, $\mathcal{K}_2$ và $\mathcal{K}_3$

3 CÁC LỚP BÀI TOÁN VÀ THUẬT GIẢI TRÊN MIỀN TRI THỨC ĐẠI SỐ TUYẾN TÍNH

Trên mô hình tri thức Đại số tuyến tính (K, Connect), các lớp bài toán có thể chia thành 2 dạng là các lớp bài trên một miền tri thức con và các lớp bài toán vận dụng sự phối hợp kiến thức giữa các miền tri thức con.

3.1 Bài toán trên mô hình COKB

Các miền tri thức con của tri thức Đại số tuyến tính được đặc tả dưới dạng mô hình COKB, do đó các lớp bài toán trên miền tri thức con này có thể được mô hình và giải quyết theo mô hình bài toán trên mô hình COKB. Các lớp bài toán trên mô hình COKB có hai loại.

3.1.1 Bài toán trên đối tượng

Mỗi đối tượng Obj trong tri thức COKB có các hành vi để giải quyết các vấn đề nội tại của nó. Bài toán trên đối tượng Obj có dạng: A → B, với A, B là các sự kiện trên các thuộc tính thuộc Obj.Attrs. Bài toán A → B được giải dựa trên các luật trong Obj.EqObj và O.RulObj và các quan hệ trong Obj.Facts. Các mục tiêu của bài toán trên đối tượng là:

Xác định giá trị các thuộc tính chưa biết từ các thuộc tính đã biết.

Xác định quan hệ giữa các thuộc tính trong đối tượng.

Cho biết quá trình suy diễn bên trong của nó để xác định các thuộc tính và quan hệ giữa các thuộc tính.

3.1.2 Bài toán trên mô hình COKB

Mô hình bài toán trên tri thức dạng COKB có dạng: (O, F) → G với O là tập các đối tượng, F là tập các sự kiện giữa các đối tượng và G là mục tiêu của bài toán.

Các mục tiêu của bài toán có thể là: xác định một đối tượng hoặc giá trị các thuộc tính của đối tượng; xác định quan hệ giữa các đối tượng; tính giá trị và rút gọn một biểu thức giữa các đối tượng; tính giá trị hàm giữa các đối tượng (Do *et al.*, 2015; Nguyễn Đình Hiển và Đỗ Văn Nhơn, 2014).

Thuật giải 3.1: Thuật giải này được xây dựng dựa trên chiến lược suy diễn tiến kết hợp với các heuristic, cùng với việc sử dụng tri thức Bài toán mẫu trong quá trình suy luận. Thuật giải này đã được trình bày trong (Do, 2010, 2015).

Bài toán trên mô hình COKB có thể đặc tả và giải được các bài toán trên từng miền tri thức con của Đại số tuyến tính. Các bài toán này trong quá trình giải chỉ cần dùng kiến thức của miền tri thức đó, chẳng hạn như: các bài toán về tính toán biểu thức ma trận, biến đổi ma trận; chứng minh sự tương đương của hai hệ phương trình (không dùng ma trận); chứng minh không gian con... Tuy nhiên, các lớp bài toán trên mô hình COKB không giải được các bài toán đòi hỏi cần phải vận dụng phối hợp kiến thức của nhiều miền tri thức con để giải quyết vấn đề.

3.2 Bài toán liên hệ giữa các miền tri thức con trong tri thức Đại số tuyến tính

Trong mục này, chúng ta sẽ nghiên cứu các lớp bài toán liên hệ giữa các miền tri thức con trong tri thức Đại số tuyến tính. Các lớp bài toán này có dạng: **(O, F) → G**

Với O là tập các đối tượng tính toán, F là các sự kiện giữa các đối tượng và G là mục tiêu của bài toán. Bài toán này được phân thành 2 loại:

Loại 1: Giả thiết bài toán có chứa các đối tượng thuộc các miền tri thức $\mathcal{K}_p$ và $\mathcal{K}_q$ (p, q =1,2,3; q < p)

$$\forall o \in O, \exists c_o \in \mathcal{K}_p.C_p \cup \mathcal{K}_q.C_q : o \in I_{c_o}$$

Loại 2: Giả thiết bài toán chỉ có các đối tượng thuộc cùng một miền tri thức $\mathcal{K}_p$ (p=1,2,3)

$$\forall o \in O, \exists c_o \in \mathcal{K}_p.C_p : o \in I_{c_o}$$

Các loại bài toán này thuộc các miền tri thức K_2 và K_3 mà trong quá trình giải cần phải vận dụng kiến thức thuộc miền tri thức thấp hơn.

3.2.1 Thuật giải cho lớp bài toán thuộc loại 1:

Trên tri thức Đại số tuyến tính (K, Connect), tìm lời giải cho bài toán P = (O, F) → G có chứa các đối tượng thuộc các miền tri thức $\mathcal{K}_p$ và $\mathcal{K}_q$ (p, q =1,2,3; q < p)

Input: P = (O, F) → G thỏa điều kiện

$$\forall o \in O, \exists c_o \in \mathcal{K}_p.C_p \cup \mathcal{K}_q.C_q : o \in I_{c_o}$$

Output: Lời giải bài toán P

Thuật giải 3.2:

B.1: Khởi tạo giá trị các biến:

Known:= F // tập các sự kiện đã suy luận được

Sol:=[ ] // danh sách các bước giải của bài toán

B.2: Phân loại (O, F) theo từng miền tri thức:

O_p, F_p: Tập các đối tượng và tập các sự kiện giữa các đối tượng thuộc miền tri thức $\mathcal{K}_p$.

O_q, F_q: Tập các đối tượng và tập các sự kiện giữa các đối tượng thuộc miền tri thức $\mathcal{K}_q$.

F_{pq}: các sự kiện giữa các đối tượng thuộc các miền tri thức khác nhau.

B.3: Sử dụng thuật giải 3.1 cho tri thức $\mathcal{K}_p$ để phát sinh các sự kiện mới dựa trên (O_p, F_p).

Cập nhật *Known* và *Sol*;

B.4: Sử dụng thuật giải 3.1 cho tri thức $\mathcal{K}_q$ để phát sinh các sự kiện mới dựa trên (O_q, F_q)

Cập nhật *Known* và *Sol*;

B.5: Sử dụng luật liên kết miền tri thức $\mathcal{K}_p$ và $\mathcal{K}_q$ trong **Connect** để phát sinh ra các đối tượng mới ở miền tri thức $\mathcal{K}_q$.

O_{pq}: tập các đối tượng mới được phát sinh

B.6: Sử dụng các đối tượng trong tập O_{pq} kết hợp với các sự kiện trong F_{pq} và *Known* chuyển bài toán P về bài toán P' trong tri thức $\mathcal{K}_q$.

B.7: Giải bài toán P' trong miền tri thức $\mathcal{K}_q$ bằng thuật giải 3.1.

B.8: Nếu bài toán P' giải được

Xuất *Sol* là lời giải của bài toán P

Nếu bài toán P' không giải được

Xuất "Không tìm thấy lời giải bài toán"

Thuật giải cho lớp bài toán thuộc loại 2:

Trên tri thức Đại số tuyến tính (K, Connect), tìm lời giải cho bài toán P = (O, F) → G chỉ có các đối tượng thuộc cùng một miền tri thức $\mathcal{K}_p$ (p=1,2,3)

Input: P = (O, F) → G thỏa điều kiện

$$\forall o \in O, \exists c_o \in \mathcal{K}_p.C_p : o \in I_{c_o}$$

Output: Lời giải bài toán P

Thuật giải 3.3:

B.1: Khởi tạo giá trị các biến:

Known:= F // tập các sự kiện đã suy luận được

Sol:=[ ] // danh sách các bước giải của bài toán

B.2: Sử dụng thuật giải 3.1 cho tri thức $\mathcal{K}_p$ để phát sinh các sự kiện mới dựa trên (O, F).

Cập nhật *Known* và *Sol* ;

B.3: Sử dụng các luật liên kết trong Connect để xác định miền tri thức $\mathcal{K}_q$ liên hệ với $\mathcal{K}_p$ (q<p).

Phát sinh các đối tượng thuộc $\mathcal{K}_q$ liên kết với $\mathcal{K}_p$

O_q: tập các đối tượng mới được phát sinh

B.4: Sử dụng các đối tượng trong tập O_q kết hợp với các sự kiện trong *Known* chuyển bài toán P về bài toán P' trong miền tri thức $\mathcal{K}_q$.

B.5: Giải bài toán P' trong miền tri thức $\mathcal{K}_q$ bằng thuật giải 3.1

Cập nhật *Known* và *Sol*.

B.6: Nếu tìm được mục tiêu bài toán P'

Chuyển về kết quả trên tri thức Kp

Cập nhật Known và Sol.

Xuất Sol là lời giải của bài toán P

B.7: Nếu chưa tìm được mục tiêu bài toán P' Chuyển bài toán P' trên miền tri thức Kq thành bài toán P'' trên miền tri thức $\mathcal{K}_r$ (r < q)

B.8: Giải bài toán P'' trong miền tri thức $\mathcal{K}_r$ bằng thuật giải 3.1

Cập nhật *Known* và *Sol*.

B.9: Nếu tìm được mục tiêu bài toán P'' Chuyển về kết quả trên tri thức Kq

Chuyển về kết quả trên tri thức Kp

Xuất Sol là lời giải của bài toán P

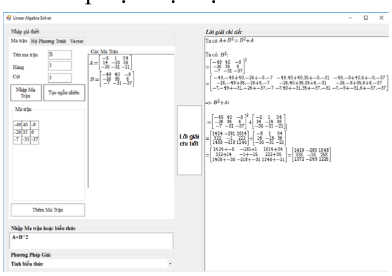
Nếu bài toán P' không giải được

Xuất "Không tìm thấy lời giải bài toán"

4 KẾT QUẢ THỰC NGHIỆM

4.1 Hệ giải bài tập Đại số tuyến tính

Chương trình giải toán tự động kiến thức Đại số tuyến tính có thể giải các bài toán trong các miền tri thức về ma trận, hệ phương trình tuyến tính bậc nhất và không gian vector một cách hiệu quả và nhanh chóng. Chương trình rất hữu ích cho các sinh viên học tập môn học ở chương trình đại học, đặc biệt là chương trình toán dành cho các khối ngành kỹ thuật. Trong bài báo này, sử dụng mô hình phối hợp các miền tri thức dạng COKB, cơ sở tri thức Đại số tuyến tính trong (Anton and Rorres, 2010; Đỗ Công Khanh *và ctv.*, 2012) đã được đặc tả trong mục 2.2 và các lớp bài toán đã được xây dựng và giải quyết trong mục 3. Từ đó, chúng tôi xây dựng chương trình có thể giải được các bài tập trong kiến thức Đại số tuyến tính trong chương trình toán cao cấp bậc Đại học.



Hình 3: Giao diện của chương trình

4.2 Kết quả thử nghiệm chương trình

Chương trình đã được thử nghiệm trên các bài tập trong sách (Đỗ Công Khanh *và ctv.*, 2012). Lời giải các bài tập tương tự như cách giải của sinh viên, phù hợp với kiến thức của người học. Ví dụ về lời giải của các bài tập:

Ví dụ 4.1: Tìm cơ sở của không gian S sinh bởi các vector: $u_1 = [1, 2, 3]$, $u_2 = [4, 5, 6]$, $u_3 = (7, 8, 9)$, $u_4 = [10, 11, 12]$

+ Đặc tả bài toán:

O := {S: KG_VECTOR,

u_1, u_2, u_3, u_4: VECTOR }

F: = {$u_1 = [1, 2, 3]$, $u_2 = [4, 5, 6]$, $u_3 = [7, 8, 9]$, $u_4 = [10, 11, 12]$,

{ u_1, u_2, u_3, u_4} tập sinh S }

G:= {Tìm: B: $2^{I_{VECTOR}}$, B cơ sở S}

+ Lời giải của chương trình:

Bước 1: Do {u_1, u_2, u_3, u_4} tập sinh S, ta có ma trận:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

Bước 2: Biến đổi A về dạng ma trận bậc thang:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \xrightarrow{\substack{(2)\leftarrow(2)-4.(1) \\ (3)\leftarrow(3)-7.(1) \\ (4)\leftarrow(4)-10.(1)}} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \\ 0 & -9 & -18 \end{bmatrix}$$

$$\xrightarrow{\substack{(2)\leftarrow-1/3.(2) \\ (3)\leftarrow-1/6.(3) \\ (4)\leftarrow-1/9.(4)}} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 0 & -1 & -2 \\ 0 & -1 & -2 \end{bmatrix} \xrightarrow{\substack{(3)\leftarrow(3)-(2) \\ (4)\leftarrow(4)-(2)}} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Bước 3: Ma trận bậc thang có 2 dòng khác 0:

[1 2 3] và [0 -1 -2]

Bước 4:

Đặt B := {$v_1 = [1, 2, 3]$, $v_2 = [0, -1, -2]$}

B là cơ sở của S.

Bảng: Kết quả việc giải các bài tập thử nghiệm trong sách

Kiến thức	Số bài tập thử nghiệm	Số bài tập giải được
Ma trận – Vector	65	58
Hệ phương trình tuyến tính	39	38
Không gian vector	67	53

5 KẾT LUẬN

Trong bài báo này, chúng tôi đã xây dựng một phương pháp biểu diễn cho nhiều miền tri thức phối hợp nhau, trong đó mỗi miền tri thức được biểu diễn dưới dạng mô hình tri thức các đối tượng tính toán (COKB). Các miền tri thức con của kiến thức Đại số tuyến tính: ma trận, hệ phương trình tuyến tính và không gian vector được biểu diễn bằng các dạng của mô hình COKB. Trên cơ sở đó, chúng được phối hợp để hình thành nên cơ sở tri thức Đại số tuyến tính. Trên cơ sở mô hình tri thức phối hợp, chúng tôi đã đề xuất các lớp bài toán cho miền tri thức phối hợp và xây dựng thuật giải để giải các lớp bài toán này.

Từ đó, hệ giải bài tập tự động kiến thức Đại số tuyến tính được xây dựng. Chương trình được thử nghiệm trên các bài toán trong sách (Đỗ Công Khanh *và ctv.*, 2012) và cho lời giải từng bước, tương tự cách giải của sinh viên.

Trong nghiên cứu tiếp theo, chúng tôi sẽ tiếp tục nghiên cứu việc bổ sung thêm kiến thức ánh xạ tuyến tính trong hệ thống. Đồng thời, xây dựng mô hình tri thức phối hợp trên các miền tri thức tổng quát để từ đó thiết kế hệ thống các agent phối hợp trong việc xử lý các miền tri thức riêng biệt.

TÀI LIỆU THAM KHẢO

Noy, N. and McGuinness, D. (Eds), 2013. Final Report on the 2013 NSF Workshop on Research Challenges and Opportunities in Knowledge Representation, National Science Foundation Workshop Report.

Baader, F., Horrocks, I., Lutz, C., Sattler, U., 2017. An Introduction to Description Logic, Cambridge Univ. Press, 255 pages.

Harmelen, F., Lifschitz, V., Porter, B., 2008. Handbook of Knowledge Representation, Elsevier.

Do, V.N., 2012. Intelligent Problem Solvers in Education: Design Method and Applications, Intelligent Systems, In: Vladimir M. Koleshko (Ed.), Intelligent systems, InTech, pp. 121-148.

Aladova, E., Plotkin, T., 2017. Logically automorphically equivalent knowledge bases, arXiv:1707.01027v1

Do, V.N., 2015. Ontology COKB for knowledge representation and reasoning in designing knowledge-based systems. Communications in Computer and Information Science (CCIS), vol. 513, Springer, pp. 101-118.

Do, V.N., Nguyen, D.H., Mai, T.T., 2015. Reasoning Method on Knowledge about Functions and Operators, International Journal of Advanced Computer Science and Applications (IJACSA), 6(6): 156 – 168.

Anton, H., Rorres, C., 2010. Elementary Linear Algebra, 10th edition, John Wiley & Sons.

Đỗ Công Khanh, Nguyễn Thu Hằng, Ngô Thu Lương, 2012. Toán cao cấp đại số tuyến tính. NXB ĐHQG-HCM

Nguyễn Đình Hiển, Đỗ Văn Nhơn, 2014. Mô hình tri thức toán tử và Ứng dụng xây dựng hệ hỗ trợ giải bài toán thông minh, Tạp chí Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 52 (4D): 60-76.

Do, V.N., 2010. Model for Knowledge Bases of Computational Objects, International Journal of Computer Science Issues, Vol. 7, Issue 3, No 8, pp. 11-20.

Quickmath, 2017. https://quickmath.com, Accessed on 01 October 2017.

MathSolver, 2017. https://play.google.com/store/apps/details?id=com.emulestudio.math&hl=en, Accessed on 01 October 2017.

Mathway, 2017. https://mathway.com/, Accessed on 01 October 2017.

Symbolab, 2017. http://symbolab.com/, Accessed on 01 October 2017.