

Đại Học Quốc Gia TP.HCM
Đại Học Công Nghệ Thông Tin



BÁO CÁO CHUYÊN ĐỀ
BIỂU DIỄN TRI THỨC VÀ ỨNG DỤNG

ĐỀ TÀI:

MÔ HÌNH COKB VÀ ỨNG DỤNG VÀO
GIẢI TOÁN HÌNH HỌC

GVHD:

Người thực hiện:

Lớp:

PGS.TS. ĐỖ VĂN NHƠN

Trần Quốc Cường

Cao học khóa 8

Mã số: CH1301082

NHA TRANG – 2014

MỤC LỤC



MỤC LỤC.....	1
PHẦN 1.MÔ HÌNH COKB.....	3
Khái niệm về đối tượng tính toán COKB.....	3
Mô hình tri thức về các đối tượng tính toán.....	5
Tổ chức cơ sở tri thức COKB.....	10
Giải toán trên đối tượng tính toán.....	11
PHẦN 2: ÁP DỤNG MÔ HÌNH COKB GIẢI TOÁN HÌNH HỌC PHẪNG.....	15
I.Giới thiệu.....	15
Thiết kế hệ cơ sở tri thức cho miền tri thức hình học phẳng.....	15
Thiết kế bộ suy diễn tự động của chương trình.....	17
PHẦN 3: LẬP TRÌNH ỨNG DỤNG COKB GIẢI TOÁN HÌNH HỌC PHẪNG.....	19
I.Giới thiệu:.....	19
II.Tạo package đọc File:.....	19
III.Code xử lý chính của chương trình:.....	22
IV.Kết quả chương trình:.....	28
V.Hướng dẫn sử dụng chương trình:.....	29
KẾT LUẬN.....	31
TÀI LIỆU THAM KHẢO.....	33

LỜI MỞ ĐẦU



Trong khoa học về trí tuệ nhân tạo, có nhiều phương pháp để biểu diễn tri thức nhưng những phương pháp này lại không hiệu quả trong việc biểu diễn và suy luận trên các tri thức phức tạp. Bên cạnh đó, các phương pháp suy diễn cũng đóng một vai trò quan trọng trong các hệ cơ sở tri thức, nhưng những phương pháp suy diễn hiện nay vẫn còn mang tính khái quát cao, chưa thể mô phỏng được lối tư duy của con người. Trong thực tế, khi giải quyết một bài toán, chúng ta thường không tìm ngay một lời giải mới mà trước tiên ta sẽ tìm những bài toán liên quan với bài toán ấy để từ đó có cách giải quyết phù hợp. Mô hình mẫu COKB, một hướng tiếp cận hiện đại, đã và đang được nghiên cứu phát triển do khả năng ứng dụng của nó trong việc biểu diễn các tri thức. Mô hình COKB là mô hình có thể sử dụng rất hiệu quả trong việc thiết kế các hệ cơ sở tri thức phức tạp, như các miền tri thức về Hình học, Giải tích, Vật lý...

Trong phạm vi của bài thu hoạch nhỏ này, em sẽ trình bày khái niệm về mô hình COKB từ đó ứng dụng mô hình này trong việc xây dựng chương trình giải toán hình học phẳng cấp THCS.

Qua đây, em cũng xin được gửi lời cảm ơn đến Phó Giáo Sư - Tiến sỹ Đỗ Văn Nhơn, người đã tận tâm truyền đạt những kiến thức nền tảng cơ bản cho em về môn học “Biểu diễn tri thức và ứng dụng”. Bên cạnh đó tôi cũng xin chân thành cảm ơn toàn thể các bạn bè học viên trong lớp đã tận tình giúp đỡ cho tôi trong những thời điểm khó khăn khi tìm hiểu tiểu luận này.

PHẦN 1. MÔ HÌNH COKB

Khái niệm về đối tượng tính toán COKB

1. Giới thiệu

Trong nhiều vấn đề giải toán dựa trên tri thức ta thường đề cập đến các đối tượng khác nhau và mỗi đối tượng có cấu trúc bao gồm một số thuộc tính với những quan hệ nhất định giúp ta thực hiện sự suy diễn, tính toán.

Cấu trúc đối tượng trên một số hành vi giải toán nhất định để tạo ra một đối tượng.

Nhiều bài toán khác nhau có thể được biểu diễn dưới dạng mạng các đối tượng. Cách biểu diễn này có thể được áp dụng một cách có hiệu quả trong các hệ giải toán, chẳng hạn như các hệ giải các bài toán hình học.

2. Định nghĩa

Một đối tượng tính toán là đối tượng O có cấu trúc gồm:

- Một danh sách các thuộc tính $\text{Attr}(O) = x_1, x_2, \dots, x_n$ trong đó mỗi thuộc tính lấy giá trị trong một miền xác định nhất định, và giữa các thuộc tính ta có các quan hệ thể hiện qua các sự kiện, các luật suy diễn hay các công thức tính toán.
- Các hành vi liên quan đến sự suy diễn và tính toán trên các thuộc tính của đối tượng hay trên các sự kiện như:

Xác định bao đóng của một tập hợp thuộc tính $A \subseteq \text{Attr}(O)$.

Xác định tính giải được của bài toán suy diễn tính toán có dạng $A \vdash B$ với $A \subseteq \text{Attr}(O)$ và $B \subseteq \text{Attr}(O)$.

Thực hiện các tính toán.

Xem xét tính xác định của đối tượng, hay của một sự kiện.

Ví dụ:

Cấu trúc tam giác gồm các yếu tố như : 3 cạnh a, b, c ; 3 góc tương ứng với 3 cạnh : α, β, γ ; 3 đường cao tương ứng : h_a, h_b, h_c ; diện tích S của tam giác, v.v ... cùng với các công thức liên hệ giữa chúng sẽ trở thành một đối tượng tính toán khi ta tích hợp cấu trúc này với các hành vi xử lý liên quan đến việc giải bài toán tam giác cũng như các hành vi xem xét một sự kiện nào đó liên quan đến các thuộc tính hay chính bản thân đối tượng.

3. Mô hình cho một đối tượng tính toán

Một đối tượng tính toán có thể được mô hình bởi bộ:

(Attrs, F, Facts, Rules)

Attrs là tập hợp các thuộc tính của đối tượng,

F là tập hợp các quan hệ suy diễn tính toán,

Facts là tập hợp các tính chất hay các sự kiện vốn có của đối tượng, và

Rules là tập hợp các luật suy diễn trên các sự kiện liên quan đến các thuộc tính cũng như liên quan đến bản thân đối tượng.

Ví dụ: Đối tượng “TAM_GIAC” được biểu diễn theo mô hình trên gồm có:

- Attrs = $\{GocA, GocB, GocC, a, b, c, h_a, h_b, h_c, m_a, m_b, m_c, p_a, p_b, p_c, S, p, R, r, r_a, r_b, r_c\}$
- F = $\{GocA + GocB + GocC = \pi, a \cdot \sin(GocB) = b \cdot \sin(GocA), a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos(GocA), \dots\}$
- Facts =

- Rules = {GocA = GocB} {a = b}, {a = b} {GocA = GocB},
{GocA=pi/2} {a^2 = b^2+c^2, b c}, ...

Xét một loại Com-object, với cấu trúc như sau:

(Attrs, F, Facts, Rules)

Cho trước (gt) A Attrs, Xác định B Attrs.

Ký hiệu vấn đề là: A B.

Algorithm:

GD 1: Tìm một lời giải **Solution** dựa trên suy diễn tiến; Solution có dạng danh sách các quan hệ suy diễn tính toán hay các luật được áp dụng.

GD 2: Thực hiện loại bước thừa trong **Solution** để được **Solution cuối cùng**.

Mô hình tri thức về các đối tượng tính toán

4. Giới thiệu

Mỗi loại đối tượng tính toán khi xét riêng biệt chỉ thể hiện được một phần tri thức có tính chất cục bộ trong ứng dụng trong khi kiến thức của con người về một lĩnh vực hay một phạm vi kiến thức nào đó thường bao gồm các khái niệm về các loại đối tượng khác nhau với những mối quan hệ v những thành phần khác liên quan.

Ví dụ: cạnh a của một tam giác là một thuộc tính của đối tượng tam giác, khi xét như một đối tượng độc lập thì nó l một “đoạn thẳng”, là một loại đối tượng có những luật riêng của nó.

Để có mot mô hình biểu diễn tri thức rộng hơn có thể sử dụng trong việc xây dựng một hệ cơ sở tri thức và giải toán về các loại đối tượng khác nhau ta cần phải xem xét khái niệm đối tượng tính toán trong một hệ thống khái niệm các đối

tượng cùng với các loại sự kiện, các loại quan hệ khác nhau và các dạng luật liên quan đến chúng.

Mô hình tri thức về các đối tượng tính toán là mô hình cho một dạng cơ sở tri thức bao gồm các khái niệm về các đối tượng có cấu trúc cùng với các loại quan hệ và các công thức tính toán liên quan.

5. Mô hình

Ta gọi một mô hình tri thức về các đối tượng tính toán, viết tắt là một mô hình COKB (Computational Objects Knowledge Base), là một hệ thống (**C, H, R, Ops, Rules**) gồm:

- Một tập hợp C các khái niệm về các đối tượng tính toán
- Mỗi khái niệm là một loại đối tượng tính toán có cấu trúc và được phân mức theo sự thiết lập của cấu trúc đối tượng, gồm: biến thực, đối tượng cơ bản, đối tượng mức 1 và đối tượng mức 2.
- Các đối tượng cơ bản có cấu trúc rỗng hoặc có cấu trúc gồm một số thuộc tính thuộc kiểu thực. Các đối tượng loại này có thể làm nền cho sự thiết lập các đối tượng ở mức cao hơn.
- Các đối tượng tính toán mức 1 có một thuộc tính loại <real> và có thể được thiết lập từ một danh sách nền các đối tượng cơ bản.

Các đối tượng tính toán mức 2 có các thuộc tính loại real và các thuộc tính thuộc loại đối tượng mức 1, và đối tượng có thể được thiết lập trên một danh sách nền các đối tượng cơ bản.

- Một tập hợp H các quan hệ phân cấp giữa các loại đối tượng

Trên tập hợp C ta có một quan hệ phân cấp theo đó có thể có một số khái niệm là sự **đặc biệt hóa** của các khái niệm khác, chẳng hạn như một tam giác cân

cũng là một tam giác, một hình bình hành cũng là một tứ giác. Có thể nói rằng H là một biểu đồ Hasse khi xem quan hệ phân cấp trên là một quan hệ thứ tự trên C .

- Một tập hợp R các khái niệm về các loại quan hệ trên các loại đối tượng

Mỗi quan hệ được xác định bởi <tên quan hệ> và các loại đối tượng của quan hệ, và quan hệ có thể có một số tính chất trong các tính chất sau đây: tính chất phản xạ, tính chất đối xứng, tính chất phản xứng và tính chất bắc cầu.

- Một tập hợp Ops các toán tử

Các toán tử cho ta một số phép toán trên các biến thực cũng như trên các đối tượng, chẳng hạn các phép toán số học và tính toán trên các đối tượng đoạn và góc tương tự như đối với các biến thực.

- Một tập hợp $Rules$ gồm các luật

Các luật thể hiện các tri thức mang tính phổ quát trên các khái niệm và các loại **sự kiện** khác nhau. Mỗi luật cho ta một qui tắc suy luận để đi đến các sự kiện mới từ các sự kiện nào đó, và về mặt cấu trúc nó gồm 2 thành phần chính là: phần giả thiết của luật và phần kết luận của luật. Phần giả thiết và phần kết luận đều là các tập hợp sự kiện trên các đối tượng nhất định.

Một luật r có thể được mô hình dưới dạng:

$$r : sk_1, sk_2, \dots, sk_n \quad sk_1, sk_2, \dots, sk_m$$

* Phân loại sự kiện:

Mỗi sự kiện là một phát biểu khẳng định một tính chất về một hay một số đối tượng tính toán. Ở đây chúng ta xem xét 6 loại sự kiện khác nhau như sau:

Loại 1: Phát biểu về loại (hay tính chất) của một đối tượng. Ví dụ: Ob là một tam giác.

Loại 2: Phát biểu về tính xác định của một đối tượng (các thuộc tính coi như đã biết) hay của một thuộc tính. Ví dụ: Giả sử đoạn AB trong tam giác ABC được cho trước.

Loại 3: Phát biểu về sự xác định của một thuộc tính hay một đối tượng thông qua một biểu thức hằng.

Ví dụ: đoạn $AB = 2 \cdot m^2 + 1$ (với m được cho trước), góc $ABC = \pi / 3$.

Loại 4: Sự kiện về sự bằng nhau giữa một đối tượng hay một thuộc tính với một đối tượng hay một thuộc tính khác.

Ví dụ: thuộc tính a của đối tượng Ob thuộc loại tam giác = đoạn CD, đối tượng Ob1 = đối tượng Ob2.

Loại 5: Sự kiện về sự phụ thuộc của một đối tượng hay của một thuộc tính theo những đối tượng hay các thuộc tính khác thông qua một công thức tính toán.

Ví dụ: $O1.a = O2.a + 2 \cdot O2.b$

Loại 6: Sự kiện về một quan hệ trên các đối tượng hay trên các thuộc tính của các đối tượng.

Ví dụ: đoạn AB song song với đoạn CD, điểm M thuộc đoạn AB.

6. Ví dụ p dụng

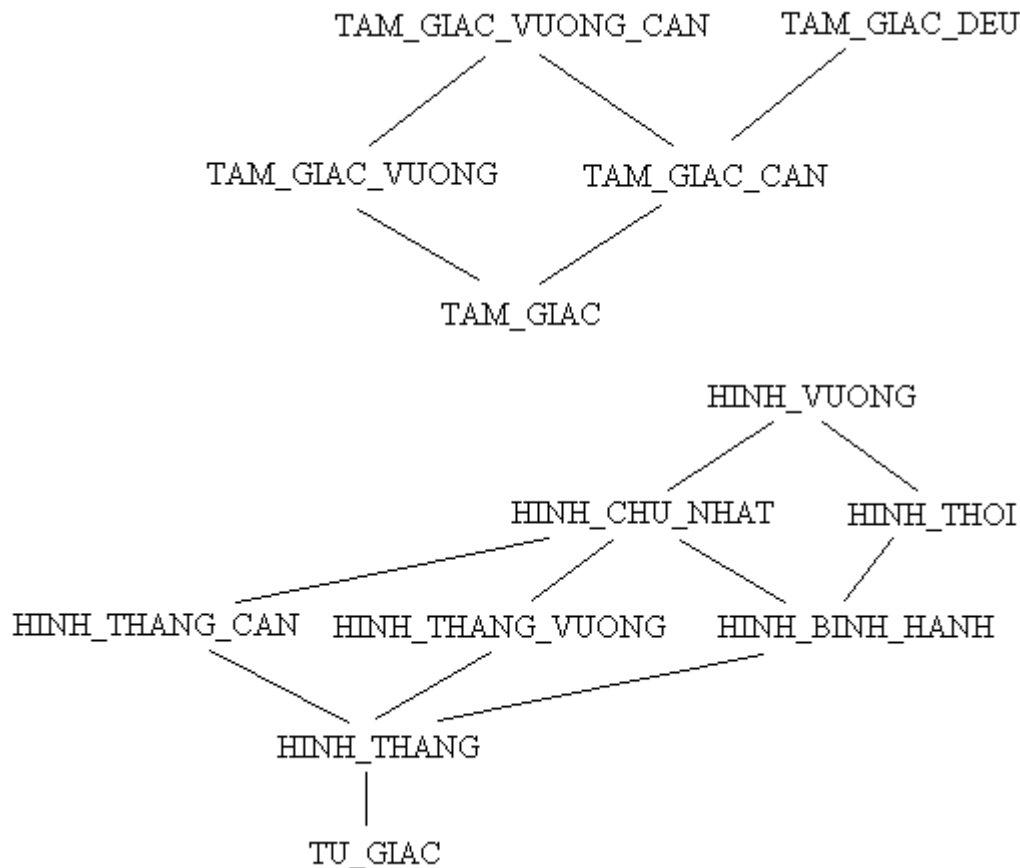
Phần kiến thức về các tam giác và các tứ giác trong hình học phẳng có thể được biểu diễn theo mô hình tri thức về các đối tượng tính toán như dưới đây.

- Các khái niệm về các đối tượng gồm:

- o Điểm, đường thẳng
- o Đoạn thẳng, tia
- o Góc.
- o Các loại tam giác và các loại tứ giác.

- Các quan hệ phân cấp giữa các loại đối tượng:

Giữa các khái niệm về các loại tam giác và các loại tứ giác có các quan hệ phân cấp theo sự đặc biệt hóa của các khái niệm, được thể hiện bởi các biểu đồ sau đây:



- Các quan hệ giữa các khái niệm bao gồm các loại quan hệ như:
 - o Quan hệ thuộc về của 1 điểm đối với một đoạn thẳng.
 - o Quan hệ trung điểm của một điểm đối với một đoạn thẳng.
 - o Quan hệ song song giữa 2 đoạn thẳng.
 - o Quan hệ vuông góc giữa 2 đoạn thẳng.
 - o Quan hệ bằng nhau giữa 2 tam giác.
- Các toán tử:

Các toán tử số học và các hàm sơ cấp cũng áp dụng đối với các đối tượng loại “đoạn thẳng” và các đối tượng loại “góc”.

- Các luật:

Các luật thể hiện các định lý hay qui tắc suy diễn trên các loại sự kiện khác nhau.

Ví dụ: Một tam giác ABC có 2 cạnh AB và AC bằng nhau thì tam giác là tam giác cân tại A. Với 3 đoạn thẳng a, b và c, nếu $a \parallel b$ và $a \perp c$ thì ta có $b \perp c$.

Tổ chức cơ sở tri thức COKB

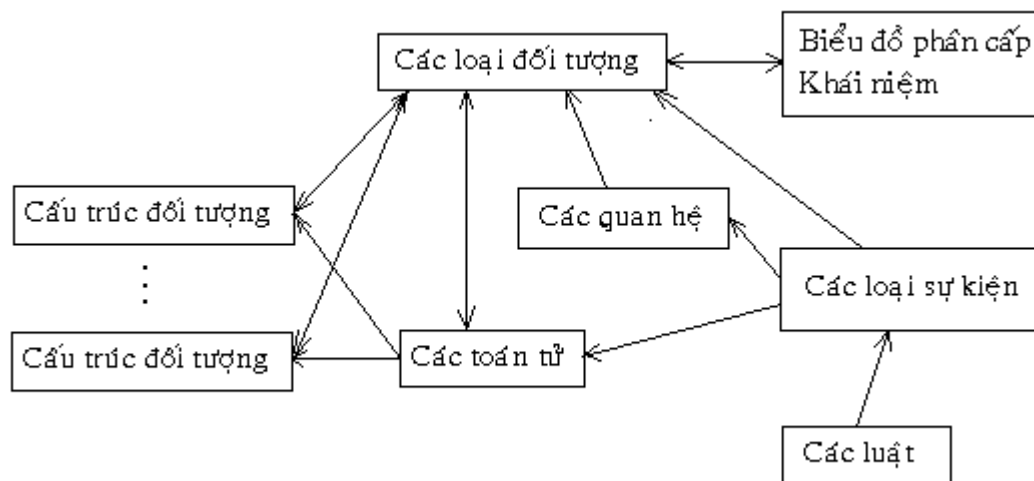
7. Các thành phần của COKB:

Một cơ sở tri thức về các đối tượng tính toán có thể được tổ chức bởi một hệ thống file text có cấu trúc gồm:

- Tập tin “Concepts.txt” lưu trữ các định danh (hay tên gọi) cho các khái niệm về các loại đối tượng; mỗi định danh có một file tương ứng lưu thông tin cấu trúc của loại đối tượng.
- Tập tin “RELATIONS.txt” lưu trữ thông tin về các loại quan hệ khác nhau trên các loại đối tượng.
- Tập tin “Hierarchy.txt” lưu lại các biểu đồ Hasse thể hiện quan hệ phân cấp đặc biệt hóa trên các khái niệm.
- Các tập tin với tên tập tin có dạng “<tên khái niệm đối tượng>.txt” để lưu trữ cấu trúc của loại đối tượng <tên khái niệm đối tượng>.
- Tập tin “Operators.txt” lưu trữ các thông tin về các toán tử trên các đối tượng.
- Tập tin “FACTS.txt” lưu trữ thông tin về các loại sự kiện khác nhau.
- Tập tin “RULES.txt” lưu trữ hệ luật của cơ sở tri thức.

- File “objects.txt” lưu các đối tượng cụ thể mặt định.

8. Biểu đồ liên hệ giữa các thành phần của COKB



Giải toán trên đối tượng tính toán

9. Các vấn đề cơ bản cho hành vi đối tượng

- Vấn đề 1:

Xét tính giải được của bài toán GT KL, trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của đối tượng.

- Vấn đề 2:

Tìm một lời giải cho bài toán GT KL, trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của đối tượng.

- Vấn đề 3:

Thực hiện tính toán các thuộc tính trong tập hợp KL từ các sự kiện trong GT trong trường hợp bài toán GT KL giải được, trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của đối tượng.

- Vấn đề 4:

Xét tính xác định của đối tượng dựa trên một tập sự kiện cho trước trên các thuộc tính của đối tượng.

10. Giải quyết vấn đề

- Định nghĩa về “sự hợp nhất” của các sự kiện.

Ví dụ về các sự kiện hợp nhất với nhau:

DOAN[A,B] và DOAN[A,B].

TAM_GIAC[A,B,C]. a và DOAN[B,C].

Ob.a = $(m+1)^2$ và Ob.a = $m^2 + 2*m + 1$.

Ob1 = Ob2 và Ob2 = Ob1.

$a^2 = b^2 + c^2$ và $b^2 = a^2 - c^2$.

“a song song b” và “b song song a”.

- Định nghĩa về một bước giải là một bước suy ra sự kiện mới từ một số sự kiện đã biết thuộc một trong các dạng suy luận như: suy diễn mặc nhiên, áp dụng luật suy diễn, áp dụng quan hệ tính toán, giải hệ phương trình, ...

Ví dụ về các bước giải:

$$\{a^2 = m^2 + 2m + 1, GocA = \frac{1}{2}\pi, c = 4\} \quad \{a, c, GocA\}$$

$$GocA = \frac{1}{2}\pi, GocB = \frac{1}{3}\pi$$

$$GocC = \frac{1}{6}\pi$$

$$GocA + GocB + GocC = \pi$$

$$GocB, GocA = \frac{1}{2} \pi$$

$$GocC = \frac{1}{2} \pi - GocB$$

$$GocA + GocB + GocC = \pi$$

$$b^2 = a^2 - c^2$$

$$\text{If } a^2 = b^2 + c^2 \text{ then } GocA = \frac{1}{2} \pi \quad GocA = \frac{1}{2} \pi$$

- Định nghĩa về một *lời giải* của một bài toán và tính giải được dựa trên quan hệ “bao hàm hợp nhất”.
- Thực hiện quá trình tìm lời giải theo cách suy diễn tiến với sự áp dụng của một số qui tắc heuristic, kết hợp với kỹ thuật loại bỏ các bước dư thừa để rút gọn lời giải.

Ví dụ 1: Xét bài toán GT KL trên đối tượng “TAM_GIAC”, với

$$GT = a, b=1, c, GocA = m*(b+c), GocA = 2*GocB ,$$

$$KL = GocB, GocC .$$

Kết quả: bài toán giải được.

Ví dụ 2: Xét bài toán GT KL trên đối tượng “TAM_GIAC”, với

$$GT = a, b=5, GocA = m*(b+c), GocA = 2*GocB, a^2=b^2+c^2 , KL = GocB, GocC .$$

Lời giải:

$$1. \text{ Suy ra } \{ GocB = \frac{1}{2} GocA \} \text{ từ } \{ GocA = 2 GocB \}$$

$$2. \text{ Suy ra } \{ GocA = \frac{1}{2} \pi \} \text{ từ } \{ a^2 = b^2 + c^2 \}$$

3. Suy ra $\{GocB = \frac{1}{4}\pi\}$ từ $\{GocB = \frac{1}{2}GocA, GocA = \frac{1}{2}\pi\}$

4. Suy ra $\{GocB\}$ từ $\{GocB = \frac{1}{4}\pi\}$

5. Suy ra $\{GocC = \frac{1}{4}\pi\}$ từ $\{GocA = \frac{1}{2}\pi, GocB = \frac{1}{4}\pi\}$ và $GocA + GocB + GocC = \pi$

6. Suy ra $\{GocC\}$ từ $\{GocC = \frac{1}{4}\pi\}$.

PHẦN 2: ÁP DỤNG MÔ HÌNH COKB GIẢI TOÁN HÌNH HỌC PHẪNG

I. Giới thiệu

Trên cơ sở hệ thống bài tập về hình học phẳng và Kỹ thuật thiết kế hệ giải toán tự động, ta dùng mô hình COKB để xây dựng hệ cơ sở tri thức và bộ suy diễn cho ứng dụng: “Chương trình giải toán tự động các bài toán hình học phẳng trong chương trình toán cấp THCS.”

Thiết kế hệ cơ sở tri thức cho miền tri thức hình học phẳng

Trên cơ sở hệ thống bài tập về hình học phẳng và Kỹ thuật thiết kế hệ giải toán tự động, ta dùng mô hình COKB

11. Tập C – tập các khái niệm đối tượng tính toán

Tập C gồm các khái niệm: “Điểm”, “Tia”, “Đoạn”, “Góc”, “Đường thẳng”, “Tam giác”, “Hình thang”, “Đường tròn”...

- “Điểm” là đối tượng cơ bản.
- “Tia”, “Đoạn” là các đối tượng cấp 1.
- “Tam giác”, “Đường tròn” là các đối tượng cấp 2.

12. Tập H tập quan hệ phân cấp giữa các đối tượng

Từ tập hợp các khái niệm về đối tượng tính toán ở trên ta có quan hệ phân cấp giữa các tượng, ví dụ như:.

- “Góc nhọn”, “Góc tù” là những dạng của khái niệm của “Góc”.
- “Tam giác cân”, “Tam giác vuông”, “Tam giác đều” là những dạng của khái niệm “Tam Giác”.

13. Tập R – tập quan hệ giữa các đối tượng tính toán

Ta có các loại quan hệ sau:

- Quan hệ nền: là quan hệ giữa các số thực.
- Quan hệ cấp cơ bản: là quan hệ giữa các đối tượng nền và quan hệ giữa các đối tượng cấp 1.
- Quan hệ cấp 1: là quan hệ giữa các đối tượng cơ bản, đối tượng cấp 1 và đối

14. Tập Ops – tập hợp các toán tử

Trong miền tri thức hình học phẳng ở cấp THCS, toán tử là quan hệ giữa các số thực nên ta có thể xem như $Ops = \{ \}$.

15. Tập Funcs – tập hợp các hàm

Tập Func gồm các hàm sau:

- Trung điểm của đoạn thẳng.
- Hình chiếu của một điểm trên đường thẳng.
- Hàm đối xứng của một điểm qua một đường thẳng.

16. Rules – tập hợp các luật

Các tính chất, mệnh đề, định lý trong tri thức toán hình học phẳng ở cấp THCS có thể được biểu diễn bằng các luật trên các đối tượng tính toán. Chẳng hạn:

$$\{a: \text{DOAN}, b: \text{DOAN}, c: \text{DOAN}, a // b, c \perp a\} \Rightarrow \{c \perp b\}.$$

$$\{A: \text{DIEM}, B: \text{DIEM}, C: \text{DIEM}, BC = AC\} \Rightarrow \{ABC \text{ là tam cân tại } C\}.$$

$$\{A: \text{DIEM}, B: \text{DIEM}, C: \text{DIEM}, AB \perp BC\} \Rightarrow \{ \text{góc } ABC = 90^\circ \}.$$

17. Tập Sample – tập hợp các bài toán mẫu

- Bài toán mẫu về việc xác định loại của đối tượng: Xác định tam giác vuông, Hình chữ nhật, đường tròn.
- Bài toán mẫu về:
 - o Giải tam giác vuông.
 - o Giải tam giác cân.
 - o Quan hệ giữa đường kính và dây cung trong đường tròn.

Thiết kế bộ suy diễn tự động của chương trình

Mô hình bài toán trên miền tri thức hình học phẳng được định nghĩa như sau:

(O, F, Goal)

Bài toán P: “Cho tam giác ABC với các giả thiết sau: đoạn AH = 6, BC = 211, Góc A = 50°. Tìm đoạn AB?”

- Mô hình bài toán:

$O = \{ \text{TAM_GIAC}[A,B,C], [H, \text{DIEM}] \}$

$F = \{ H = \text{HINHCHIEU}(A, \text{DOAN}[B,C]),$

$\text{DOAN}[A,H].\text{dai} = 6, \text{DOAN}[B,C].\text{dai} = 211$

$\text{GOC}[C,A,B] = 50; \}$

$\text{Goal} = \{ \text{DOAN}[A,B].\text{dai} \}.$

- Lời giải của chương trình:

- o Bước 1:

$\{ H = \text{HINHCHIEU}(A, \text{DOAN}[B,C]), \text{DOAN}[B,C].\text{dai},$

$\text{GOC}[C,A,B] \}$

→ {GOC[B,C,A]}

Bởi “Luật suy diễn”: [“Tính chất của tam giác”]

o Bước 2:

{ GOC[C,A,B], GOC [B,C,A] }

→ {GOC[A,B,C]}

Bởi “Luật suy diễn”: [“Tính chất của tam giác”]

o Bước 3:

{ DOAN[A,C], GOC[C,A,B] }

→ {[“DUONGCAO”, DOAN[C,H], TAMGIAC[A,B,C]]}

Bởi “Luật suy diễn”: [“Tính chất của tam giác”]

o Bước 4:

{[“DUONGCAO”, DOAN[C,H], TAMGIAC[A,B,C]],
DOAN[A,B,C] }

→ {DOAN[B,C]}

Bởi “Luật suy diễn”: [“Tính chất của tam giác”]

PHẦN 3: LẬP TRÌNH ỨNG DỤNG COKB GIẢI TOÁN HÌNH HỌC PHẪNG

I. Giới thiệu:

Để viết chương trình demo giải toán hình học phẳng theo mô hình COKB, em quyết định sử dụng ngôn ngữ lập trình Maple vì những ưu điểm sau:

- Maple là phần mềm chuyên dụng cho công việc tính toán bao gồm các tính toán thuần túy bằng ký hiệu toán học, các tính toán số và các tính toán bằng đồ thị.
- Maple dễ sử dụng, đòi hỏi cấu hình không lớn, đáp ứng nhu cầu tính toán của nhiều đối tượng.

II. Tạo package đọc File:

2. Danh sách các files theo cấu trúc COKB

Danh sách các files theo cấu trúc COKB như đã đề cập ở mục III.1

- TIA.txt
- DIEM.txt
- DOAN.txt
- DUONG_THANG.txt
- GOC.txt
- HINH_BINH_HANH.txt
- HINH_VUONG.txt
- TAM_GIAC.txt

- TU_GIAC.txt
- OBJECTS.txt
- Hierarchy.txt
- RELATIONS.txt
- RULES.txt

3. Tạo package đọc files:

Đây là code mẫu để đọc file Rules.

```
Readrules := proc()
```

```
local read_1Rule;
```

```
read_1Rule := proc()
```

```
local loai, tens, kieus, ten1, n1, kieu1, gt_kl, k;
```

```
loai := ""; tens := []; kieus := []; gt_kl := [{}, {}];
```

```
line := readline(fd);
```

```
# bỏ qua dòng begin_rule
```

```
while line <> 0 and SearchText("begin_rule", line) = 0 and
```

```
SearchText("end_rules", line) = 0 do
```

```
line := readline(fd);
```

```
end do;
```

```
# thoát khỏi hàm nếu đến dòng cuối cùng end_rules
```

```
if SearchText("end_rules", line) > 0 then
```

```
RETURN (NULL);
```

```
fi;
```

```
line := readline(fd);
```

```
# đọc đến dòng end_rule thì dừng lại
```

```
while line <> 0 and SearchText("end_rule", line) = 0 do

    if SearchText("kind_rule", line) > 0 then

        loi := rhs(parse(line));

    else

        k := SearchText(":", line);

        # doc phan hypothesis_part va gan vao bien hypothesis_part = gt_kl[1]

        if SearchText("hypothesis_part", line) > 0 then

            gt_kl[1] := substring(line, (k+1)..length(line));

            line := readline(fd);

            while line <> 0 and SearchText("end_hypothesis_part", line) = 0 do

                gt_kl[1] := cat (gt_kl[1], line);

                line := readline(fd);

            end do;

            gt_kl[1] := parse(gt_kl[1]);

        # doc phan goal_part va gan vao bien goal_part = gt_kl[2]

        elif SearchText("goal_part", line) > 0 then

            gt_kl[2] := substring(line, (k+1)..length(line));

            line := readline(fd);

            while line <> 0 and SearchText("end_goal_part", line) = 0 do

                gt_kl[2] := cat (gt_kl[2], line);

                line := readline(fd);

            end do;

            gt_kl[2] := parse(gt_kl[2]);

        # doc phan A,B,C : DIEM; tens = A,B,C; kieus = DIEM

        elif k > 0 then

            ten1 := [parse( substring(line, 1..(k-1)) )]; n1 := nops(ten1);
```

```
        kieu1 := convert ( parse( substring(line, (k+1)..length(line)) ), string);

        tens := [op(tens), op(ten1)]; kieus := [op(kieus), kieu1 $ n1];

        fi;

    fi;

    line := readline(fd);

end do; # while

if gt_kl[1] <> {} or gt_kl[2] <> {} then

    Rule := [op(Rule), [loai, tens, kieus, gt_kl] ];

fi;

end: # Read_1Rule

while line <> 0 and SearchText("end_rules", line) = 0 do

    read_1Rule();

end do; # while

end: # Readrules
```

III. Code xử lý chính của chương trình:

1. Hàm đọc sự kiện Facts:

```
GetFacts:=proc(nameObj)

    local facts,f;

    facts := {};

    for f in ObjStruct(nameObj)[6] do

        facts := `union`(facts, {f[6]});

    end do;

    return facts
```


end proc;

2. Hàm đọc Rules:

GetRules:=proc(nameObj)

local rules,r;

rules := {};

for r in ObjStruct(nameObj)[7] do

rules := `union`(rules, {{r[4][1][1], r[4][2][1]}});

end do;

return rules;

end proc;

3. Hàm xử lý chính cho bài toán:

Tinh:=proc(GT,KL,nameObj)

*local knownVar,knownVal,r,flag,Vnew,findResultPath,flag1,exactResult,
checked,knownVar1,r1,rr1,rules,g,u,M,F,s,conditions,v;*

M := ObjStruct(nameObj)[2];

F := GetFacts(nameObj);

findResultPath := [];

knownVal := GT;

knownVar := {};

start of bo them vao F dua vao rules

for g in GT do

if evalb(SetVars(rhs(g),nameObj) = {}) then

knownVar := {lhs(g), op(knownVar)};

else

duyệt rules và gán vào cho F

rules := GetRules(nameObj);

```
for u in rules do
    if evalb(u[1] = g) then
        F := `union`(F, {convert(g, name), convert(u[2], name)});
    elif evalb(u[2]=g) then
        F := `union`(F, {convert(g, name), convert(u[1], name)});
    else
        F := `union`(F, {convert(g, name)});
    end if;
end do;

end if;

end do;

# end of bo them vao F dua vao rules

while not(KL subset knownVar) do
    flag := false;

    for r in F do
        if nops(SetVars(parse(r),nameObj) minus knownVar) = 1 then
            flag := true; Vnew := `minus`(SetVars(parse(r), nameObj), knownVar);
            break;
        end if;
    end do;

    if not flag then
        return printf(" Không tìm ra được kết quả\n");
    end if;

    knownVar := `union`(knownVar, Vnew);

    # sub: thay the KnownVal trong ctrinh r

    knownVal := `union`(knownVal, solve(subs(knownVal, parse(r)), Vnew));
```

```
findResultPath := [op(findResultPath), [parse(r), Vnew[1]]];

end do;

# viet thuat giai loai bo luat thua

# danh sach tham so tinh ra ket qua minus tham so gia thiet

knowVar1 := `minus`(knownVar, map(proc (x) options operator, arrow; lhs(x) end proc, GT));

# danh sach ket qua da rut gon

exactResult := [findResultPath[nops(findResultPath)]];

# truy vet lai ket qua vi du:  $p=2 \cdot a+2 \cdot b$  suy ra duoc qua 2 tham so a va b

checked := `minus`(SetVars(findResultPath[nops(findResultPath)])[1], nameObj), KL);

# lap lai cho den khi ket qua truy vet la {}

while not evalb(nops(checked) = 0) do

  for r1 in checked[1] do

    for rr1 in findResultPath do

      if (evalb(rr1[2] = r1)) and not ({r1} subset map(x->lhs(x),GT)) then

        checked := `minus`(`union`(checked, SetVars(rr1[1], nameObj)), {rr1[2]});

        checked := `minus`(checked, {r1});

        exactResult := [op(exactResult), rr1];

      end if;

    end do;

  end do;

  if (not (checked subset knowVar1)) and evalb(nops(checked)=nops(GT)) then

    checked := `minus`(checked, map(proc (x) options operator, arrow; lhs(x) end proc, GT));

  end if;

  if (not (checked subset knowVar1)) then

    checked := `minus`(checked, map(proc (x) options operator, arrow; lhs(x) end proc, GT));

  end if;
```

```
end do;

# Xuất kết quả

XuấtKQ(exactResult, GT, nameObj);

# In giá trị kết quả

for s in knownVal do

    if evalb(lhs(s)=KL[1]) then

        printf("_ Kết quả %s\n", convert(s, string));

    end if;

end do;

end proc;

XuấtKQ:=proc(Results,GT,nameObj)

    local i,j,xuat,temp,temp1,deduces,l,temp2,temp3,temp4,temp5,m;

    j := 1;

    deduces := [];

    for i from nops(Results) by -1 to 1 do

        deduces := [op(deduces), [Results[i], j]];

        j := j+1;

    end do;

    j := 1;

    for i from nops(Results) by -1 to 1 do

        temp := convert(Results[i][1], string);

        temp1 := convert(j, string);

        printf("_ Bước thu %s : %s\n", temp1, temp);

        # Xét từng biến bên vế phải

        for l in SetVars(Results[i][1],nameObj) do
```

```
# Chỉ xét trường hợp không phải là kết quả
if not evalb(l = Results[i][2]) then

# Kiểm tra có thuộc tập gia thiết
if {l} subset map(x->lhs(x), GT) then

temp2 := convert(l, string);

printf(" %s : gia thiết\n", temp2);

else # Kiểm tra có thuộc tập suy diễn

for m in deduces do

if evalb(m[1][2] = l) then

temp3 := convert(m[1][2], string);

temp4 := convert(m[2], string);

printf(" %s : suy ra từ bước thu %s\n", temp3, temp4);

end if;

end

end if;

end do;

# In kết quả

temp5 := convert(Results[i][2], string);

printf(" Tính được %s\n", temp5);

j := j+1;

end do;

return deduces;

end proc;
```

IV. Kết quả chương trình:

- Bài toán 1: Cho tam giác ABC, với các giả thiết GT sau: bán kính đường tròn nội tiếp $r = 3$, chu vi $p = 4$, đoạn $AC = 21$. Tính đường cao HB.

- o Nhập vào chương trình như sau:

Tinh({b = 21, p = 4, r = 3}, {hb}, "TAM_GIAC")

- o Kết quả xuất ra:

```
_ Bước thu 1 : S = p*r
p : giả thiết
r : giả thiết
Tính được S
_ Bước thu 2 : S = 1/2*b*hb
S : suy ra từ bước thu 1
b : giả thiết
Tính được hb
_ Kết quả hb = 8/7
```

- Bài toán 2: Cho tam giác ABC với các giả thiết sau: đoạn $AH = 6$, $BC = 211$, Góc $A = 50^\circ$. Tính đoạn AB.

- o Nhập vào chương trình như sau:

Tinh({b = 211, ha = 6, GocA = 50}, {a}, "TAM_GIAC")

- o Kết quả xuất ra:

```
_ Bước thu 1 : ha = b*sin(GocC)
b : giả thiết
ha : giả thiết
Tính được GocC
_ Bước thu 2 : GocA+GocB+GocC = Pi
GocA : giả thiết
GocC : suy ra từ bước thu 1
Tính được GocB
_ Bước thu 3 : hc = b*sin(GocA)
b : giả thiết
GocA : giả thiết
Tính được hc
_ Bước thu 4 : hc = a*sin(GocB)
hc : suy ra từ bước thu 3
```

GocB : suy ra từ bước 2

Tính được a

_ Kết quả a = $211 \cdot \sin(50) / \sin(\pi - 50 - \arcsin(6/211))$

- Bài toán 3: Cho tam giác ABC với các giả thiết sau: góc A = 50° , AC = 69. Tính đoạn BC.


- o Nhập vào chương trình như sau:

Tính($\{b = 69, \text{GocA} = 50\}, \{a\}, \text{"TAM_GIAC"}$)

- o Kết quả xuất ra:

Không tìm ra được kết quả

V. Hướng dẫn sử dụng chương trình:

- Download phần mềm Maple 13.
- Tạo thư mục CObject_Knowledge tại ổ đĩa D, và chép các files TIA.txt, DIEM.txt, DOAN.txt, DUONG_THANG.txt, GOC.txt, HINH_BINH_HANH.txt, HINH_VUONG.txt, TAM_GIAC.txt, TU_GIAC.txt, OBJECTS.txt, Hierarchy.txt, RELATIONS.txt, RULES.txt
- Chép file package “TriangleLib.m” vào thư mục lib của thư mục cài đặt Maple (thường là C:\Program Files\Maple 13\lib).
- Sử dụng maple để mở file “TieuLuan_Triangle.mw”.
- Click vào biểu tượng  để thực thi chương trình.
- Đưa con trỏ đến cuối dòng của chương trình và thực hiện tính toán như bước II.4.
 - o Bài toán: Cho tứ giác ABCD, với các giả thiết sau: Góc A = 60° , cạnh AB = 110, cạnh BC = 10, cạnh CD = 120, chu vi p = 190, Góc C = 30° , Góc D = 140° . Tính diện tích S.

o Nhập vào chương trình như sau:

tugiac := Tinh({GA = 60, GC = 30, GD = 140, a = 110, b = 10, c = 120,
p = 190}, {S}, "TU_GIAC")

o Kết quả xuất ra:

_ Buoc thu 1 : $a+b+c+d = p$
a : gia thiet
b : gia thiet
c : gia thiet
p : gia thiet
Tinh duoc d
_ Buoc thu 2 : $GA+GB+GC+GD = 2*Pi$
GA : gia thiet
GC : gia thiet
GD : gia thiet
Tinh duoc GB
_ Buoc thu 3 : $2*S = a*b*\sin(GB)+c*d*\sin(GD)$
GB : suy ra tu buoc thu 2
GD : gia thiet
a : gia thiet
b : gia thiet
c : gia thiet
d : suy ra tu buoc thu 1
Tinh duoc S
_ Ket qua S = $550*\sin(2*Pi-230)-3000*\sin(140)$


```
tgiac4 := Tinh({b = 69, GocA = 50}, {a}, "TAM_GIAC");
  Không tìm ra được kết quả

tugiac := Tinh({GA = 60, a = 110, b = 10, c = 120, p = 190, GC = 30, GD = 140}, {S}, "TU_GIAC");
_ Bước thu 1 : a+b+c+d = p
  a : giá thiết
  b : giá thiết
  c : giá thiết
  p : giá thiết
  Tính được d
_ Bước thu 2 : GA+GB+GC+GD = 2*Pi
  GA : giá thiết
  GC : giá thiết
  GD : giá thiết
  Tính được GB
_ Bước thu 3 : 2*S = a*b*sin(GB)+c*d*sin(GD)
  GB : suy ra từ bước thu 2
  GD : giá thiết
  a : giá thiết
  b : giá thiết
  c : giá thiết
  d : suy ra từ bước thu 1
  Tính được S
_ Kết quả S = 550*sin(2*Pi-230)-3000*sin(140)
```

KẾT LUẬN



M

ô hình COKB là mô hình thích hợp cho việc thiết kế một cơ sở tri thức với các khái niệm có thể được biểu diễn bởi các đối tượng tính toán, cấu trúc tường minh giúp dễ dàng thiết kế các mô đun truy cập cơ sở tri thức. Tiện lợi cho việc thiết kế các mô đun giải toán tự động. Thích hợp cho việc định ra một ngôn ngữ khai báo bài toán và đặc tả bài toán một cách tự nhiên. Các mô hình và thuật giải được đề xuất có thể làm công cụ cho việc xây dựng các hệ giải bài toán dựa trên tri thức, các hệ cơ sở tri thức, và các phần mềm dạy học với sự hỗ trợ giải toán thông minh.

Là mô hình rất tốt cho việc biểu diễn các tri thức của con người, đặc biệt là các tri thức về Toán học, Vật lý, Hóa học.

Chương trình giải toán tự động về Toán Hình Học phẳng ở THCS được xây dựng bằng ứng dụng mô hình COKB cho việc biểu diễn tri thức trên miền tri thức này. Lời giải của hệ thống tự nhiên, chính xác và phù hợp cách suy nghĩ của con người.

Tuy nhiên việc ứng dụng các thuật toán vào thực tiễn là bài toán khó, kiến thức của học viên còn chưa sâu vì vậy chương trình cũng như nội dung tiểu luận vẫn còn những khiếm khuyết, rất mong được Thầy nhận xét và góp ý để học viên hiểu biết thêm nhiều kiến thức về công nghệ này.

Em xin chân thành cảm ơn Thầy PGS.TS Đỗ Văn Nhơn, giảng viên chuyên đề Biểu Diễn tri thức và Ứng dụng, đã truyền đạt những kiến thức quý báu về các thuật toán và tạo cơ hội được làm chuyên đề này để hiểu sâu hơn

Trân trọng cảm ơn Thầy.

TÀI LIỆU THAM KHẢO



Slide COKB(2011) – 5 components. PGS.TS Đỗ Văn Nhơn.

Phương pháp suy diễn trên mô hình COKB dựa trên tri thức bài toán mẫu và ứng dụng. PGS.TS Đỗ Văn Nhơn

Đỗ Văn Nhơn, Xây dựng hệ tính toán thông minh – Xây dựng và phát triển các mô hình biểu diễn tri thức cho các hệ giải toán tự động, Luận án tiến sĩ, Đại học quốc gia – HCM (2001-2002).

Hoàng Kiếm & Đỗ Văn Nhơn, Mở rộng và phát triển mô hình tri thức các đối tượng tính toán, Kỷ yếu Hội thảo Quốc Gia Một số vấn đề chọn lọc của CNTT, NXB Khoa học kỹ thuật (2005).

Đỗ Văn Nhơn, Kiến trúc hệ giải bài tập cho người học và kỹ thuật thiết kế, Tạp chí Khoa. Học Giáo dục kỹ thuật, Đại học sư phạm kỹ thuật TpHCM, Số 2(4) 2007.

Bộ Giáo dục và Đào tạo, Sách Giáo khoa Bài học và Bài tập các lớp THCS, NXB Giáo dục.

Hoàng Kiếm, Giải một Bài toán trên Máy tính như thế nào, tập 1, NXB Giáo Dục (2000).

G. Polya, Giải bài toán như thế nào, Nhà xuất bản Giáo dục (1997).

Frank van Harmelen & Vladimir & Bruce, 2008, “Handbook of Knowledge, Representation”, Elsevier.

Stuart Russell & Peter Norvig, Artificial Intelligence – A modern approach (second edition), Prentice Hall (2003).

John F. Sowa. Knowledge Representation: Logical, Philosophical and Computational Foundations, Brooks/Cole (2000).

Nhon Do, An ontology for knowledge representation and Applications, Proceeding of World Academy of science, engineer and technology, vol. 32, August 2008, ISSN: 2070-370

Nhon Van Do, Computational Networks for Knowledge Representation, World Academy of Science, Engineering and Technology, Volume 56, August 2009, ISSN 2070 – 3724 (ICCSISE 2009), Singapore, 2009.

Nhon Do & Hien Nguyen, Model for Knowledge Representation using Sample Problems and Designing a Program for automatically solving algebraic problems, World Academy of Science, Engineering and Technology, (ICEEEL 2010), Paris, 2010.

Nhon Do, Hien Nguyen, “A reasoning method on Computation Network and Its applications.”, 2011 International MultiConference of Engineers and Computer Scientists, IMECS 2011, ISBN: 978-988-18210-3-4, pp. 137-141, Hongkong , March 2011.

<http://www.maplesoft.com>.

<http://maplevn2008.wordpress.com>