

KIẾN TRÚC HỆ GIẢI BÀI TẬP CHO NGƯỜI HỌC VÀ CÁC KỸ THUẬT THIẾT KẾ

Đỗ Văn Nhơn

ABSTRACT

Using of information technology for E-learning now becomes popular. Systems for solving exercises are very useful applications to learners. This kind of software is also called a problem solver based on a knowledge base. In this paper we present an organization of that software together with design techniques. A design pattern, the system for solving problems in 2-dimensionnal analytic geometry, will also be presented here.

TÓM TẮT

Ngày nay việc ứng dụng công nghệ thông tin trong dạy và học đã trở nên phổ biến. Hệ giải bài tập hay hỗ trợ giải bài tập cho người học là một trong những ứng dụng rất thiết thực trong E-learning nói chung. Loại hệ thống này thuộc lớp ứng dụng được gọi là các hệ giải bài toán dựa trên tri thức. Trong bài báo này sẽ trình bày về tổ chức của một hệ giải bài toán dựa trên tri thức cùng với những kỹ thuật thiết kế, và một mẫu thiết kế ứng dụng cụ thể là hệ hỗ trợ giải bài toán hình học giải tích 2 chiều.

I. GIỚI THIỆU

Hệ giải bài tập hay hỗ trợ giải bài tập cho người học là một trong những ứng dụng rất thiết thực trong E-learning nói chung. Loại hệ thống này thuộc lớp ứng dụng được gọi là các hệ giải bài toán dựa trên tri thức. Một hệ giải bài toán dựa trên tri thức phải là một hệ giải toán thông minh có thể giải được các dạng bài toán tổng quát trong một miền tri thức nào đó, trong đó nó có một cơ sở tri thức và một bộ phận thực hiện suy luận giải bài toán trong phạm vi tri thức của hệ thống. Hệ thống như thế về cơ bản có dạng như một hệ chuyên gia (xem [3]). Tuy nhiên đối với hệ giải bài tập dùng cho việc học tập còn đòi hỏi lời giải của hệ thống phải tương tự như lời giải của con người được trình bày một cách tự nhiên, dễ đọc và dễ hiểu.

Hiện nay đã có nhiều kết quả nghiên cứu liên quan đến loại ứng dụng này bao gồm các nghiên cứu chung về các hệ chuyên gia, các hệ cơ sở tri thức và các hệ trợ giúp quyết định. Về mặt kỹ thuật các nhà nghiên cứu đặc biệt chú trọng vào các phương pháp biểu diễn tri thức, các chiến lược suy diễn và giải toán tự động. Về vấn đề biểu diễn

tri thức và các phương pháp ta có thể tham khảo [1] và [2]. Các phương pháp biểu diễn tri thức có thể được phân chia thành 4 dạng phương pháp:

1. Các phương pháp biểu diễn dựa trên logic hình thức. Các phương pháp này sử dụng các biểu thức logic hình thức để diễn đạt các sự kiện và các luật trong cơ sở tri thức.
2. Các phương pháp biểu diễn tri thức thủ tục. Loại phương pháp này biểu diễn tri thức như là một tập hợp các chỉ thị dùng cho việc giải quyết các bài toán. Trong nhiều hệ chuyên gia ứng dụng các chỉ thị như thế thường được thể hiện bởi một tập các luật dẫn có dạng if ... then
3. Các phương pháp biểu diễn dạng mạng. Biểu diễn mạng nắm bắt kiến thức như là một đồ thị trong đó các đỉnh biểu diễn cho các khái niệm hay các đối tượng và các cạnh biểu diễn các quan hệ hay những sự kết hợp nào đó giữa các đối tượng và các khái niệm. Phổ biến nhất trong các phương pháp loại này là các mạng ngữ nghĩa và các đồ thị khái niệm.

4. Các phương pháp biểu diễn cấu trúc. Các ngôn ngữ biểu diễn cấu trúc cho phép sử dụng các cấu trúc dữ liệu phức tạp và các cấu trúc dữ liệu trừu tượng trong biểu diễn. Ví dụ như các khung (frames) và các đối tượng (objects).

Các phương pháp trên rõ ràng đều chỉ biểu diễn được một khía cạnh của kiến thức thực tế. Tri thức cần được biểu diễn trong các hệ giải bài tập thường rất đa dạng và bao gồm các khái niệm từ đơn giản đến có cấu trúc phức tạp, các hệ thức tính toán với những qui luật nhất định, các liên hệ đa dạng bao gồm cả định tính lẫn định lượng, các luật dẫn và các heuristics, v.v... Trong [4] và [6] các tác giả đã đề xuất một phương pháp biểu diễn tri thức với mô hình tri thức có thể dùng cho việc xây dựng các cơ sở tri thức ứng dụng trong thực tế, và [8] trình bày một mẫu thiết kế ứng dụng cụ thể minh họa cho việc ứng dụng các mô hình và các kỹ thuật đề xuất trong các báo cáo này. Một cách tiếp cận nữa khá hiện đại cho việc biểu diễn tri thức và thiết kế các cơ sở tri thức là sử dụng các “ontology” ([7]).

Các phương pháp suy diễn tự động nhằm vận dụng kiến thức đã biết trong quá trình lập luận giải quyết vấn đề trong đó quan trọng nhất là các chiến lược điều khiển giúp phát sinh những sự kiện mới từ các sự kiện đã có. Các kỹ thuật suy diễn tự động đã được các nhà nghiên cứu khảo sát khá đầy đủ ở mức độ tương đối khái quát bao gồm:

1. Phương pháp hợp giải trong biểu diễn tri thức dưới dạng logic vị từ.

2. Phương pháp suy diễn tiến: “Chiến lược suy luận được bắt đầu bằng tập sự kiện đã biết, rút ra các sự kiện mới nhờ dùng các luật mà phần giả thiết khớp với sự kiện đã biết, và tiếp tục quá trình này cho đến khi thấy trạng thái đích, hoặc cho đến khi không còn luật nào khớp được các sự kiện đã biết hay được sự kiện suy luận”.

3. Phương pháp suy diễn lùi: phương pháp này được tiến hành bằng cách truy

ngược từ mục tiêu cần đạt được trở về phần giả thiết của bài toán bằng cách áp dụng các luật trong cơ sở tri thức. Quá trình suy diễn lùi này sẽ phát sinh một sơ đồ cây mục tiêu kèm theo một cơ chế quay lui và lời giải sẽ được tìm thấy khi tất cả các mục tiêu ở các nút lá của cây mục tiêu đều thuộc về những sự kiện đã biết.

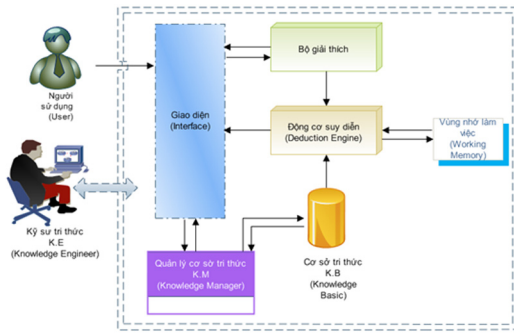
4. Kết hợp suy diễn tiến-lùi và sử dụng các heuristics. Mỗi phương pháp suy diễn tiến và lùi đều có ưu nhược điểm của nó. Việc kết hợp 2 phương pháp này một cách thích hợp cùng với việc sử dụng các qui tắc heuristics sẽ cho ta một phương pháp suy diễn hiệu quả trong các ứng dụng.

Các kết quả trên còn mang tính khái quát cao, một số mô hình và kỹ thuật lại mang tính bộ phận chưa đủ tốt cho việc xây dựng một hệ giải bài tập cho người học với những yêu cầu khá tự nhiên nhưng không dễ thực hiện đối với người thiết kế và cài đặt hệ thống ứng dụng. Trong báo cáo này chúng tôi đúc kết một số nghiên cứu về cấu trúc hệ thống và các kỹ thuật thiết kế đã được sử dụng trong xây dựng các ứng dụng cụ thể: hệ tra cứu và hỗ trợ giải toán hình học phẳng, hệ hỗ trợ giải toán hình học giải tích, hệ truy vấn kiến thức về đại số tuyến tính. Mẫu thiết kế hệ giải bài toán hình học giải tích 2 chiều cũng sẽ được trình bày như một minh họa cụ thể.

II. CẤU TRÚC HỆ GIẢI BÀI TẬP

Một hệ giải bài tập có cấu trúc tương tự như một hệ chuyên gia, với một số tiêu chuẩn bổ sung thêm cho phù hợp với nhu cầu hỗ trợ học và luyện tập giải bài tập cho người học. Ta có thể thiết kế hệ thống loại bao gồm các thành phần sau đây:

- Cơ sở tri thức
- Động cơ suy diễn
- Bộ giải thích
- Vùng nhớ làm việc
- Bộ quản lý cơ sở tri thức
- Phần giao diện



Hình 1: Hệ giải bài toán dựa trên tri thức

Hình 1 thể hiện kiến trúc của hệ thống nhằm phục vụ cho 2 loại đối tượng: người học (hay người sử dụng) và người quản trị cơ sở tri thức (hay kỹ sư tri thức).

Cơ sở tri thức chứa các kiến thức cần thiết cho việc giải các bài toán trong một miền tri thức nhất định mà hệ thống đang hỗ trợ. Ví dụ: Cơ sở tri thức của hệ giải bài toán hình học giải tích 2 chiều bao gồm các kiến thức của hình học giải tích 2 chiều và một vài kiến thức liên quan đến hình học phẳng.

Bộ suy diễn (còn gọi là mô-tơ suy diễn) sẽ áp dụng kiến thức được lưu trữ trong cơ sở tri thức để giải quyết hay tìm lời giải cho các bài toán đặt ra được đưa tới (từ người học) hệ thống. Nó phải nhận dạng được bài toán và thực hiện các suy diễn thích hợp để tìm được các kiến thức cần thiết và áp dụng để giải bài toán.

Vùng nhớ làm việc lưu trữ những sự kiện và các kiến thức được sử dụng trong quá trình tìm lời giải cho bài toán.

Bộ giải thích hỗ trợ diễn giải cho các bước giải, các khái niệm và các luật được sử dụng trong lời giải.

Bộ quản lý tri thức nhằm giúp người quản trị cập nhật vào cơ sở tri thức. Nó cũng hỗ trợ tìm kiếm kiến thức và kiểm tra phần nào tính nhất quán của kiến thức được đưa vào trong cơ sở tri thức.

Phần giao diện của hệ thống đòi hỏi phải có ngôn ngữ quy ước giúp sự giao tiếp giữa người học với hệ thống, giữa người kỹ sư tri thức với hệ thống.

III. PHƯƠNG PHÁP THIẾT KẾ

Trong phần này sẽ trình bày quy trình xây dựng một hệ giải bài tập cho người học, và các kỹ thuật liên quan đến mỗi giai đoạn trong quá trình xây dựng ứng dụng.

Để xây dựng hệ ứng dụng hỗ trợ giải bài tập ta có thể thực hiện một quy trình gồm các giai đoạn dưới đây.

Giai đoạn 1: Thu thập tri thức thực tế và phân loại tri thức dựa trên các mô hình biểu diễn tri thức thích hợp đối với tri thức của ứng dụng, đồng thời xác lập các loại yêu cầu xử lý tri thức được đặt ra dựa trên việc sưu tập và phân loại các dạng bài tập mà hệ thống phải thực hiện tìm lời giải.

Giai đoạn 2: Xây dựng tổ chức cơ sở tri thức cho hệ thống dựa trên việc mô hình hóa cho toàn bộ tri thức hay mô hình hóa cho cơ sở tri thức. Về mặt kỹ thuật, để có được mô hình tri thức cho hệ thống ta phải phải tiến hành các bước sau đây:

Bước 1: Dựa trên sự phân loại tri thức thích hợp ở giai đoạn 1, thực hiện việc phân rã toàn bộ tri thức thành các phần tri thức bộ phận.

Bước 2: Mô hình hóa cho các bộ phận tri thức. Mô hình hóa tri thức thường dựa trên việc vận dụng các phương pháp biểu diễn tri thức cơ bản và một số phương pháp tiếp cận mới như: các phương pháp dùng các hệ logic hình thức, mạng ngữ nghĩa, hệ luật dẫn, các frames và classes, các ngôn ngữ đặc tả, các ontology.

Bước 3: Tích hợp các mô hình bộ phận để có mô hình cho tri thức tổng thể. Về mặt kỹ thuật, việc tích hợp này thường dựa trên các phép ghép nối, tổ hợp theo các cấu trúc hay các mô hình cơ bản cùng với các đặc tả dùng các ngôn ngữ quy ước cho máy tính.

Giai đoạn 3: Mô hình hóa vấn đề/bài toán và xây dựng các thuật giải.

Dựa trên hệ thống các bài tập được thu thập, phân loại, và mô hình cơ bản cho vấn đề/bài toán trong lý thuyết về “phương pháp giải quyết vấn đề/bài toán” ta có thể xây dựng các mô hình tổng quát cho các lớp bài toán để từ đó có thể thiết kế các thuật giải suy diễn cùng giao diện của hệ thống. Các bước để xây dựng các mô hình gồm:

Bước 1: Phân loại bài toán theo các dạng khác nhau từ đơn giản đến phức tạp như: dạng “khung” (frame), dạng bài toán xác nhận sự kiện hay chứng minh, dạng bài toán tìm kiếm đối tượng hay sự kiện, v.v...

Bước 2: Phân loại sự kiện, và biểu diễn sự kiện một cách thích hợp cho việc xử lý trên máy tính. Trong thực tế các sự kiện được phân loại và thường có cấu trúc. Chính điều này làm cho vấn đề biểu diễn tri thức và suy diễn trên máy tính trở nên khó khăn và phức tạp.

Bước 3: Mô hình hóa cho các dạng bài toán theo sự phân loại và biểu diễn ở bước 1 và bước 2. Từ các mô hình cho từng dạng bài toán ta có thể xây dựng mô hình tổng quát cho vấn đề/bài toán đặt ra cho hệ thống giải quyết.

Vấn đề cơ bản cho việc thiết kế các thuật giải suy diễn giải bài toán dựa trên tri thức là kỹ thuật hợp nhất các sự kiện. Về mặt kỹ thuật, dựa trên các loại sự kiện với cấu trúc nhất định ta phải xác lập tiêu chuẩn (hay ý nghĩa) cho sự hợp nhất sự kiện. Trên cơ sở đó thiết kế thuật giải thực hiện xem xét tính hợp nhất của 2 sự kiện.

Vấn đề tiếp theo là chiến lược và kỹ thuật suy diễn trên máy tính. Về mặt kỹ thuật, vấn đề khó khăn nhất là mô hình hóa cho kinh nghiệm và sự nhạy bén mang tính cảm tính và trực giác của con người, nhằm tìm ra các quy tắc heuristics thích hợp cho quá trình suy diễn và mô

phỏng được quá trình tư duy giải bài toán của con người. Ví dụ: kinh nghiệm chọn lựa các luật để áp dụng, kinh nghiệm và khả năng phát sinh đối tượng mới một cách thông minh.

Giai đoạn 4: Thiết kế giao diện và cài đặt chương trình.

Tương tự như các loại ứng dụng (phần mềm máy tính) khác, ta phải thiết kế giao diện và viết chương trình dựa trên các ngôn ngữ lập trình hay các công cụ thích hợp. Phần giao diện thường kết hợp các thành phần cơ bản của giao diện theo chuẩn “windows” với các ngôn ngữ quy ước để đặc tả cho vấn đề/bài toán.

Giai đoạn 5: Kiểm tra, thử nghiệm và triển khai ứng dụng.

IV. MỘT MẪU THIẾT KẾ: HỆ GIẢI BÀI TOÁN HÌNH HỌC GIẢI TÍCH 2 CHIỀU

Trong phần này trình bày tóm lược thiết kế hệ giải bài tập hình học giải tích 2 chiều trong chương trình giáo dục phổ thông trung học.

Đầu tiên các kiến thức về hình học giải tích 2 chiều được thu thập và phân loại bao gồm:

(1) Các khái niệm về các đối tượng hình học cơ bản như “điểm”, “vector”, “đường thẳng”, “đoạn thẳng”, “đường tròn”, v.v... Mỗi khái niệm có cấu trúc tương ứng gồm các thuộc tính và các quy tắc liên hệ.

(2) Các quan hệ trên các đối tượng như:

- Quan hệ “song song” giữa 2 vector.
- Quan hệ “vuông góc” giữa 2 vector.
- Quan hệ “thuộc về” giữa điểm và đường thẳng.
- Quan hệ “thẳng hàng” giữa 3 điểm.

(3) Các phép toán trên các đối tượng hình học cơ bản như:

- Phép toán cộng/trừ 2 vector.
- Phép nhân vô hướng 2 vector.

– Phép nhân số với vector.

(4) Các hàm tính toán trên các đối tượng hình học như:

– Hàm khoảng cách giữa 2 điểm.

– Hàm khoảng cách giữa điểm và đường thẳng.

– Hàm tìm giao điểm của 2 đường thẳng.

– Hàm tìm hình chiếu của một điểm lên một đường thẳng.

– Hàm tìm điểm đối xứng của một điểm qua một đường thẳng.

(5) Các mệnh đề và các định lý như:

– Hai vector vuông góc nhau khi tích vô hướng của chúng bằng 0.

– Hai đường thẳng có các pháp vector vuông góc với nhau thì chúng sẽ vuông góc nhau.

– Nếu điểm N là đối xứng của điểm M qua mặt phẳng (P) thì khoảng cách từ M đến (P) bằng khoảng cách từ N đến (P).

Từ sự thu thập kiến thức hình học giải tích 2 chiều với sự phân loại như trên, và dựa trên kỹ thuật thiết kế được trình bày trong mục 3 ta đi đến mô hình hóa cho tri thức bởi bộ sáu:

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

Trong đó các thành phần được mô tả như sau:

(1) C là một tập hợp các khái niệm về các C-Object.

(2) H là một tập hợp các quan hệ phân cấp giữa các loại đối tượng.

(3) R là tập hợp các khái niệm về các loại quan hệ trên các C-Object.

(4) Ops là một tập hợp các toán tử.

(5) Funcs là một tập hợp các hàm.

(6) Rules là tập hợp các luật được phân lớp.

Mỗi đối tượng thuộc C là một lớp các đối tượng tính toán có cấu trúc nhất định và được phân cấp theo sự thiết lập của cấu trúc các đối tượng.

Tiếp theo là việc xây dựng mô hình tổng quát cho các bài toán hình học giải tích 2 chiều. Dựa trên việc khảo sát và phân tích các bài tập cụ thể, ta đi đến phân loại và cấu trúc hóa các sự kiện gồm 12 loại sự kiện như sau:

– Sự kiện loại 1: Sự kiện thông tin về loại của đối tượng.

– Sự kiện loại 2: Sự kiện về tính xác định của một đối tượng hay của một thuộc tính của đối tượng.

– Sự kiện loại 3: Sự kiện về tính xác định của một đối tượng hay của một thuộc tính của đối tượng thông qua biểu thức hằng.

– Sự kiện loại 4: Sự kiện về sự bằng nhau của một đối tượng hay một thuộc tính của đối tượng với một đối tượng hay một thuộc tính khác.

– Sự kiện loại 5: Sự kiện về sự phụ thuộc giữa các đối tượng và các thuộc tính của các đối tượng thông qua một công thức tính toán hay một đẳng thức theo các đối tượng hoặc các thuộc tính.

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

– Sự kiện loại 7: Sự kiện về tính xác định của một hàm.

– Sự kiện loại 8: Sự kiện về tính xác định của một hàm thông qua một biểu thức hằng.

– Sự kiện loại 9: Sự kiện về sự bằng nhau giữa một đối tượng với một hàm.

– Sự kiện loại 10: Sự kiện về sự bằng nhau của một hàm với một hàm khác.

– Sự kiện loại 11: Sự kiện về sự phụ thuộc của một hàm theo các hàm hay các đối tượng khác thông qua một công thức tính toán.

– Sự kiện loại 12: Sự kiện về sự phụ thuộc giữa các hàm hay các đối tượng thông qua một đẳng thức theo các hàm hay các đối tượng.

Mô hình hóa cho bài toán tổng quát bởi:

(O, M, F, Facts, FFacts, Goal)

Trong đó các thành phần được mô tả như sau:

O là tập các đối tượng tính toán trong bài toán,

M là tập các thuộc tính của các đối tượng được đề cập tới trong bài toán,

F là tập các hàm được đề cập trong bài toán,

Facts là tập các sự kiện thuộc 6 loại 1-6, và

FFacts là tập các sự kiện thuộc các loại khác liên quan đến tri thức hàm.

Goal là mục tiêu hay yêu cầu của bài toán có thể được ghi dưới dạng danh sách.

Kế đến là việc xây dựng các thuật giải mô phỏng tư duy tìm lời giải của con người cho bài toán. Về mặt lý thuyết, các thuật giải dựa trên việc tìm kiếm các bước giải. Bước giải là một bước suy ra sự kiện mới từ một số sự kiện đã biết, thuộc một trong các dạng suy luận sau:

(1) Deduce_From3s: suy ra các sự kiện loại 2 từ các sự kiện loại 3.

(2) Deduce_From43s: suy ra các sự kiện mới loại 3 từ các sự kiện loại 3 và 4 bằng cách thay thế các biến trong sự kiện loại 3 vào sự kiện loại 4.

(3) Deduce_From53s: suy ra các sự kiện mới loại 3, 4, 5 từ các sự kiện loại 3 và 5 bằng cách thay thế các biến trong sự kiện loại 3 vào sự kiện loại 5.

(4) Deduce_From45s: suy ra các sự kiện mới loại 3 từ các sự kiện loại 4 và 5 bằng cách giải hệ phương trình.

(5) Deduce_From8s: suy ra các sự kiện loại 7 từ các sự kiện loại 8.

(6) Deduce_From983s: suy ra các sự kiện loại 3, 8 từ các sự kiện loại 3, 8, 9 bằng cách thế các biến trong sự kiện loại 8 (hay sự kiện loại 3) vào sự kiện loại 9.

(7) Deduce_Objects: thực hiện suy diễn và tính toán bên trong cấu trúc của từng đối tượng. Các đối tượng tham gia vào bước giải có khả năng thực hiện các hành vi nhất định để phát sinh sự kiện mới, thực hiện suy diễn tính toán trên các thuộc tính của đối tượng, bản thân đối tượng hay các đối tượng liên quan được thiết lập trên nền của đối tượng.

(8) Deduce_From9s: suy ra các sự kiện loại 2, 3, 6, 7, 8 từ các sự kiện loại 9 bằng cách thực hiện tính toán hàm.

(9) Deduce_Rules: dò tìm luật có thể áp dụng được.

(10) Deduce_Funcs: dò tìm hàm có thể áp dụng được.

(11) Deduce_EqsGoal: giải hệ phương trình đơn giản gồm n phương trình với n ẩn.

Thuật giải tổng quát được xây dựng dựa trên chiến lược suy diễn tiến kết hợp với suy diễn lùi và việc sử dụng các heuristic. Việc áp dụng các quy tắc heuristic là rất quan trọng trong tìm kiếm suy diễn và tính toán. Các heuristic giúp ta có thể tìm được lời giải tốt (nhưng không chắc chắn là lời giải tốt nhất), thường dễ dàng và nhanh chóng đưa ra kết quả hơn, cho lời giải tự nhiên gần với cách suy nghĩ và hành động của con người. Trong một số thuật toán, ta thường dùng bổ sung các thủ thuật heuristic ở những khâu lựa chọn nhất định.

Bài toán sẽ được đưa vào hệ thống nhờ vào một ngôn ngữ quy ước khá đơn giản và tự nhiên để đặc tả cho bài toán.

Ví dụ: Xét bài toán: Viết phương trình đường thẳng đi qua giao điểm hai đường thẳng $2x + 7y - 8 = 0$; $3x + 2y + 5 = 0$ và hợp với đường thẳng (d): $2x + 3y - 7 = 0$ một góc 45° .

Bài toán có thể được mô hình hóa theo mô hình trên như sau:

$O := \{ d: \text{đường thẳng},$

$d1: \text{đường thẳng},$

$d2: \text{đường thẳng},$

$d3: \text{đường thẳng}, A: \text{điểm}\}$

$M := \{d.ptdt, d1.ptdt, d2.ptdt, d3.ptdt\}$

$F := \{Goc(\text{đthẳng}, \text{đthẳng}),$

$GiaoDiem(\text{đthẳng}, \text{đthẳng})\}$

$Facts := \{ d.ptdt : 2x + 3y - 7 = 0,$

$d1.ptdt: 2x + 7y - 8 = 0,$

$d2.ptdt: 3x + 2y + 5 = 0,$

$A \text{ “thuộc” } d3\}$

$FFacts := \{ Goc(d,d3) = 45,$

$A = GiaoDiem(d1, d2)\}$

$Goal := [d3.ptdt]$

Chương trình được cài đặt dựa trên phần mềm đại số máy tính MAPLE (xem [9]) cho việc suy luận và tính toán, kết hợp với ngôn ngữ JAVA cho việc thiết kế giao diện. Chương trình cho phép nhập vào bài toán theo cách khá tự nhiên và dễ dàng, rồi thực hiện các thuật giải tìm kiếm lời giải cho bài toán dựa trên cơ sở tri thức tổ chức theo mô hình ECOKB, và cuối cùng chương trình cho lời giải tương minh bao gồm những bước giải cụ thể được trình bày tương tự như cách giải của con người. Dưới đây là một ví dụ minh họa.

Ví dụ: Cho $A(1, 1)$, $B(-1, 3)$ và đường thẳng (L): $x + y + 4 = 0$. Tìm trên (L) điểm C cách đều 2 điểm A, B.

$GT = \{A=[1,1], B=[-1,3], [“THUOC”, C, L], L.f=(x+y+4=0), KHOANGCACH(A, C) = KHOANGCACH(B, C)\}$

$KL = \{C\}$

Lời giải của hệ thống:

Bước 1: $\{A = [1, 1], B = [-1, 3], L.f = (x+y+4 = 0)\} \rightarrow \{A, B, L.f\}$

bởi “Deduce_From3”.

Bước 2: $\{L.f\} \rightarrow \{L\}$

bởi “Deduce_ObjRules”: [$“xac_dinh_doi_tuong”$, $[L.f.]$, [$“PTDT_TQ”$], $[L.f.]$, $\{L\}$], [$“”$], $\{L.f\}$, $\{L\}$].

Bước 3: $\{L, C\} \rightarrow \{C[1]+C[2]+4 = 0\}$

bởi “Ap dụng luật”: [$“thanh_lap_phuong_trinh”$, $[C, L]$, [$“DIEM”$, $“DUONGTHANG”$], $[{[“THUOC”, C, L]}, \{L\}]$, $tlpt[C, L]$].

Bước 4: $\{A, B, C\} \rightarrow$

$$\begin{aligned} & \sqrt{C[1]^2 - 2C[1] + C[2]^2 - 2C[2] + 2} \\ & = \sqrt{C[1]^2 + 2C[1] + C[2]^2 - 6C[2] + 10} \end{aligned}$$

bởi “Ap dụng sk10”:

$KHOANGCACH(A, C) =$

$KHOANGCACH(B, C).$

Bước 5: $\{C[1]+C[2]+4 = 0,$

$$\begin{aligned} & \sqrt{C[1]^2 - 2C[1] + C[2]^2 - 2C[2] + 2} \\ & = \sqrt{C[1]^2 + 2C[1] + C[2]^2 - 6C[2] + 10} \end{aligned} \}$$

$\rightarrow \{C = [-3, -1]\}$ bởi “Deduce_EqsGoal”

Bước 6: $\{C = [-3, -1]\} \rightarrow \{C\}$

bởi “Deduce_From3”

V. KẾT LUẬN

Trong bài báo này trình bày một sự đúc kết về cấu trúc và các kỹ thuật thiết kế hệ giải bài tập dựa trên cơ sở tri thức của môn học nhằm hỗ trợ cho người học luyện tập giải toán trong quá trình học tập. Kết quả được trình bày là sự tổng hợp những nghiên cứu về phương pháp, kỹ thuật xây dựng các hệ ứng dụng giải toán dựa trên tri thức trong thực tế của việc dạy và học như hệ giải bài toán hình học giải tích 2 chiều.

TÀI LIỆU THAM KHẢO

- [1] Stuart Russell & Peter Norvig, Artificial Intelligence – A modern approach (second edition), Prentice Hall (2003).
- [2] [2] John F. Sowa. Knowledge Representation: Logical, Philosophical and Computational Foundations, Brooks/Cole (2000).
- [3] Joseph C. Giarratano & Gary D. Riley, Expert Systems: Principles and Programming, fourth edition, International Thomson Publishing (2004).
- [4] 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).
- [5] Đỗ Văn Nhơn & Phạm Thi Vương, Hệ giải toán dựa trên tri thức về giới hạn, đạo hàm và tích phân, Kỷ yếu Hội thảo Quốc gia lần 2 FAIR'05, NXB KHKT (2006).
- [6] 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).
- [7] L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Suder, T. Lumpp, A. Abecker, G. Breiter, J. Dinger, The Role of Ontologies in Autonomic Computing Systems, IBM Systems Journal, Vol 43, No 3 (2004).
- [8] Do Van Nhon. A system that supports studying knowledge and solving of analytic geometry problems. 16th World Computer Congress 2000, Proceedings of Conference on Education Uses of Information and Communication Technologies, Beijing, China.
- [9] M.B. Monagan, K.O. Geddes, G. Labahn, S.M. Vorkoetter, J. Maccarron, P. Demarco. Maple 9 – Introductory Programming Guide. Waterloo Maple Inc. 2003.