CHƯƠNG 1

BIỂU DIỄN TRI THỨC VÀ HỆ GIẢI TOÁN DỰA TRÊN TRI THỨC

Tri thức đóng vai trò rất quan trọng đối với khả năng của một chuyên gia. Trong khoa học Trí tuệ Nhân tạo, để xây dựng các hệ chuyên gia và các hệ giải bài toán dựa trên tri thức người ta phải thiết kế một cơ sở tri thức cho hệ thống và một động cơ suy diễn để giải quyết vấn đề dựa trên tri thức. Chất lượng hoạt động chuyên gia phụ thuộc rất lớn vào kiến thức đã có cho nên nghiên cứu phát triển các phương pháp biểu diễn tri thức và cơ chế suy luận giải bài toán dựa trên tri thức có ý nghĩa rất lớn về lý thuyết cũng như ứng dụng của khoa học máy tính nói chung và của khoa học Trí tuệ Nhân tạo nói riêng, đặc biệt là đối với các hệ giải toán dựa trên tri thức. Các hệ chương trình như thế phải:

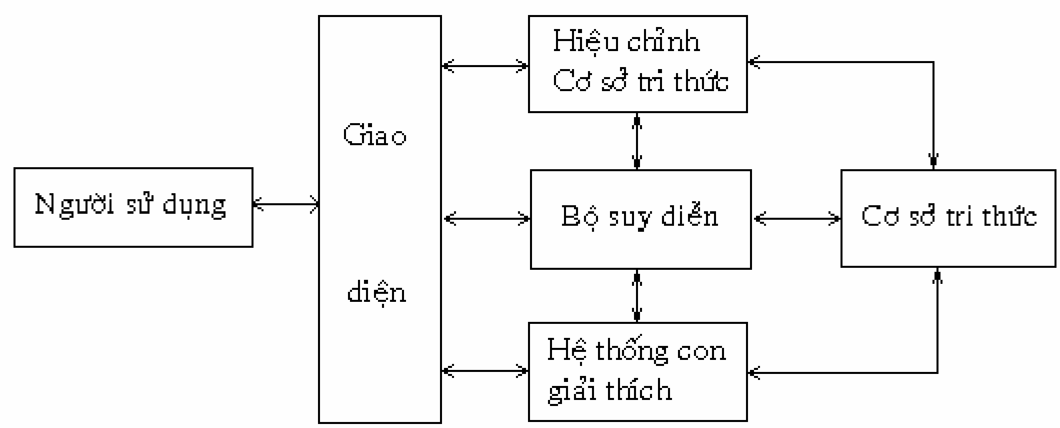
1. Cho phép kiểm tra quá trình suy luận bao gồm việc thể hiện cụ thể các bước giải bài toán và trả lời hay giải thích cho quá trình giải.
2. Cho phép việc hiệu chỉnh và cập nhật cơ sở tri thức như thêm và loại bớt kiến thức trong cơ sở tri thức.
3. Sử dụng các heuristic (thường không đầy đủ) trong việc suy luận giải bài toán nhằm đạt được các lời giải tốt.

Ngoài ra, các hệ giải bài toán dùng trong giáo dục còn đòi hỏi một lời giải tốt phù hợp với cách suy nghĩ và cách viết bình thường của con người, càng tự nhiên càng tốt.

1.1 Các vấn đề cơ bản trong thiết kế một hệ giải bài toán dựa trên tri thức

1.1.1 Cấu trúc của một 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. Cấu trúc cơ bản của hệ thống bao gồm các thành phần được chỉ ra trên hình 1.1 bên dưới.



Hình 1.1 Cấu trúc của một hệ giải toán thông minh

Có thể nói rằng trái tim của hệ thống là phần cơ sở tri thức, trong đó chứa các kiến thức cần thiết cho việc giải các bài toán trong một miền tri thức nhất định. Đối với các hệ chuyên gia dựa trên luật thì cơ sở tri thức sẽ bao gồm một hệ luật dẫn với mỗi luật có dạng *if ... then ...* .

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. Nó phải nhận dạng được bài toán và thực hiện các điều khiển thích hợp trong quá trình suy diễn. Các điều khiển nầy phải tách biệt đối với cơ sở tri thức. Bảo đảm sự tách biệt giữa cơ sở tri thức và bộ suy diễn là một tiêu chuẩn quan trọng bởi vì:

1. Sự tách biệt của kiến thức giải bài toán và bộ suy diễn sẽ làm cho việc biểu diễn tri thức được thực hiện một cách tự nhiên hơn, gần gũi hơn với quan niệm của con người.
2. Các nhà thiết kế hệ thống giải bài toán thông minh sẽ tập trung vào việc nắm bắt và tổ chức cơ sở tri thức hơn là phải đi vào những chi tiết cho việc cài đặt trên máy tính.
3. Sự tách biệt này sẽ tăng cường tính mô-đun hóa của phần cơ sở tri thức, bộ suy diễn và bộ phận cập nhật, hiệu chỉnh kiến thức. Sự bổ sung hay loại bỏ bớt một phần kiến thức sẽ không gây ra những hiệu ứng lề cho các thành phần khác trong hệ thống.
4. Cho phép cùng một chiến lược điều khiển và giao tiếp có thể được sử dụng cho nhiều hệ thống khác nhau.
5. Sự tách biệt của kiến thức giải bài toán và bộ suy diễn còn giúp ta có thể thử nghiệm nhiều chiến lượt điều khiển khác nhau trên cùng một cơ sở tri thức.

Hệ thống con giải thích cho phép chương trình diễn giải và trình bày sự suy luận. Nhiều hệ thống còn có mô-dun để hiệu chỉnh và cập nhật kiến thức cho chương trình.

1.1.2 Vấn đề Biểu diễn Tri thức

Như đã biết vấn đề biểu diễn tri thức đóng vai trò rất quan trọng trong thiết kế và xây dựng một hệ giải bài toán thông minh. Đây cũng là một hướng nghiên cứu quan trọng cho các nhà nghiên cứu về Trí tuệ Nhân tạo. Có thể nói Phương pháp biểu diễn tri thức thích hợp sẽ tạo nên một hệ thống có trái tim khỏe mạnh. Mặc dù đây là một vấn đề trung tâm nhưng ít khi người ta trả lời trực tiếp câu hỏi: Biểu diễn tri thức là gì? Một số bài báo thì nêu lên một cách khái quát khái niệm biểu diễn tri thức, còn một số khác thì bàn đến một số tính chất cần phải có của một biểu diễn. Theo sự quan niệm được nêu trong [??] thì

> "biểu diễn tri thức là thể hiện các mô tả về thế giới bên ngoài dưới dạng sao cho các máy thông minh có thể đưa tới những kết luận về môi trường quanh nó, trên cơ sở một cách hình thức các mô tả này."

Trong [12] và [13] tác giả định nghĩa về biểu diễn tri thức như sau:

> "Thể hiện tri thức là phương pháp dùng để mã hóa tri thức trong cơ sở tri thức của hệ thống"

Các tác giả khác như George F. Luger ([26]) và Gerhard Lakemeyer ([41]) cũng trình bày một số quan điểm về biểu diễn tri thức trong đó các phương pháp biểu diễn tri thức được xem xét và phân loại. Trong quá trình xây dựng cơ sở tri thức, người lập trình phải chọn lựa các đối tượng và các quan hệ thích hợp trong miền tri thức và thực hiện một phép ánh xạ chúng trên một ngôn ngữ hình thức. Từ đó chương trình sẽ có đủ kiến thức cho việc giải quyết các bài toán trong phạm vi kiến thức được biểu diễn. Các phương pháp biểu diễn khác nhau, như trình bày trong [26], 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. Các thủ tục chứng minh sẽ áp dụng kiến thức vào các bài toán cụ thể. Phép tính logic vị từ cấp 1 được sử dụng phổ biến nhất và có cả một ngôn ngữ lập trình hỗ trợ cho phương pháp này. Đó là ngôn ngữ lập trình PROLOG và các kỹ thuật lập trình với ngôn ngữ nầy có thể tìm thấy trong [33] và [52].
2. Các phương pháp biểu diễn tri thức thủ tục. Loại phương pháp nầy 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 nầy 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 biểu diễn tri thức nầy rõ ràng đều có những ưu nhược điểm nhất định nhất là mỗi phương pháp chỉ biểu diễn được một khía cạnh của kiến thức trong khi tri thức cần được biểu diễn trong các hệ ứng dụng rất đa dạng và thường 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... Chính vì vậy mà nhiều chuyên gia cũng đã khẳng định rằng khi xây dựng một cơ sở tri thức cho chương trình người thiết kế và lập trình phải vận dụng một cách linh hoạt sáng tạo các phương pháp biểu diễn tri thức cơ bản đã nêu trên.

1.1.3 Vấn đề Suy diễn Tự động

Cùng với vấn đề biểu diễn tri thức, suy diễn tự động để giải quyết các bài toán dựa trên tri thức cũng là một vấn đề quan trọng. Các phương pháp suy diễn tự động nhằm vận dụng kiến thức đã biết trong quá trính 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 . Trong phương pháp biểu diễn logic hình thức ta đã sử dụng các luật suy diễn như luật "Modus Ponens", luật "Modus Tollens" và luật "tam đoạn luận".
2. Phương pháp suy diễn tiến (forward chaining). Trong [13] tác giả có nêu lên định nghĩa cho phép suy luận tiến như sau:

   "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 (Backward chaining). 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 và suy diễn lùi. Mỗi phương pháp suy diễn tiến và lùi đều có ưu nhược điểm của nó. Việc kết hợp 2 phương pháp nầy một cách thích hợp sẽ cho ta một phương pháp suy diễn hiệu quả trong các ứng dụng cụ thể.

Trong cách tiếp cận theo phương pháp hình thức nhiều chuyên gia đã xây dựng các hệ thống logic khác nhau cho việc chứng minh định lý và suy diễn tự động. Trong [42] các tác giả đã trình bày nhiều công trình nghiên cứu theo hướng tiếp cận này. Ngoài ra trong [46] cũng nêu lên một số kết quả nghiên cứu về suy diễn tự động trong các ứng dụng cụ thể.

1.2 Phân tích, đánh giá một số công trình nghiên cứu đã có

Trong phần này sẽ bàn luận về một số công trình lý thuyết cũng như ứng dụng đã có liên quan đến mục tiêu của đề tài từ đó đề tài sẽ nêu lên các mục tiêu cụ thể được tập trung nghiên cứu, giải quyết.

1.2.1 Các phương pháp biểu diễn tri thức

Trong phần trước chúng ta biết rằng các phương pháp biểu diễn tri thức đóng vai trò rất lớn trong thiết kết các hệ chuyên gia và các hệ giải toán thông minh bởi vì nó là cơ sở và là phương tiện cho việc xây dựng trái tim của hệ thống: cơ sở tri thức. Một cơ sở tri thức với đầy đủ những kiến thức cần thiết và được tổ chức một cách thích hợp, có cấu trúc tường minh sẽ giúp cài đặt thuận lợi bộ suy diễn với hiệu quả cao. Các phương pháp biểu diễn tri thức chung đã biết như các phương pháp biểu diễn theo logic hình thức, các phương pháp biểu diễn thủ tục, các phương pháp biểu diễn dạng mạng và các phương pháp biểu diễn cấu trúc được trình bày trong các tài liệu [2], [8], [12], [13], [22], [26], [38] và [41] đều có những ưu điểm nhất định trong việc biểu diễn từng dạng tri thức. Tuy nhiên các phương pháp này đều có một nhược điểm chung là chỉ biểu diễn được một khía cạnh của tri thức đa dạng và chưa hướng tới một mô hình tri thức bao hàm nhiều dạng thông tin và nhiều dạng sự kiện khác nhau. Chính vì vậy mà nhiều chuyên gia đã khẳng định rằng cần phải có sự kết hợp của các phương pháp biểu diễn khác nhau trong thiết kế và xây dựng một cơ sở tri thức cho một hệ giải toán thông minh trong một miền tri thức. Tuy nhiên, các nhà nghiên cứu chưa chỉ rõ làm thế nào để thực hiện sự kết hợp nầy, nhất là chưa xây dựng các mô hình tri thức thật sự cho một cơ sở tri thức thuộc một lớp hay một số lớp miền tri thức nhất định.

1.2.2 Một số lý thuyết về chứng minh và suy diễn tự động

Trong cách tiếp cận theo phương pháp hình thức nhiều chuyên gia đã xây dựng các hệ thống logic khác nhau cho việc chứng minh định lý và suy diễn tự động. Trong [42] các tác giả đã trình bày nhiều công trình nghiên cứu theo hướng tiếp cận này. Tuy nhiên các kết quả lý thuyết khá trừu tượng như thế rất khó áp dụng trong các hệ chuyên gia và các hệ giải toán dựa trên tri thức trong thực tế vì các hệ nầy đòi hỏi phải có một cơ sở tri thức dựa trên các mô hình biểu diễn tri thức có tính trực quan , tính mô đun hóa cao và chứa đựng nhiều thành phần tri thức đa dạng. Các công trình nghiên cứu được trình bày trong [46] mang tính áp dụng thực tế cao hơn trong đó các chuyên gia đã nêu lên nhiều phương pháp suy diễn cụ thể khác nhau dựa trên các biểu diễn tri thức khá cụ thể như phương pháp tính toán với các tập đại số (computing with algebraic sets), phương pháp khử trong chứng minh các bài toán hình học (elimination method). Nhìn chung các phương pháp nầy chưa cho ta các mô hình biểu diễn tri thức thích hợp cho việc thiết kết các cơ sở tri thức cho các hệ thống giải quyết vấn đề dựa trên tri thức.

1.2.3 Một số phương pháp chứng minh định lý hình học

Liên quan đến đề tài chứng minh tự động các định lý hình học có rất nhiều công trình khác nhau của các chuyên gia nổi tiếng trong lĩnh vực này. Trong [49] và [50] các tác giả đã giới thiệu một phương pháp chứng minh định lý hình học và cho lời giải đọc được theo nghĩa là: chứng minh đủ ngắn để người ta có thể lập lại bằng cách viết tay và mỗi bước chứng minh có một ý nghĩa hình học rõ ràng. Phương pháp này được gọi là phương pháp diện tích (area method) trong đó phần lớn các quan hệ hình học đều được qui về quan hệ trên diện tích của các hình và việc lập luận chứng minh sẽ tiến hành dựa trên các quan hệ về diện tích. Bổ sung cho phương pháp diện tích các tác giả đã đề xuất một khái niệm hình thức liên quan đến góc "*full angle*" và thực hiện một số chứng minh tự động dựa trên khái niệm nầy. Điểm hay của phương pháp là tìm ra được một vài đặc trưng cho một số quan hệ hình học để có thể chứng minh tự động với lời giải đọc được. Tuy nhiên ta có thể thấy ngay các hạn chế của phương pháp là nó cho lời giải thiếu tự nhiên và không thích hợp cho việc xây dựng một chương trình giải toán hay hỗ trợ giải toán trong giáo dục. Về mặt thiết kế hệ thống thì phương pháp này chưa đáp ứng như là một công cụ biểu diễn tri thức mang tính hệ thống giúp xây dựng các thành phần chính của một chương trình máy tính. Nói một cách khác các phương pháp diện tích và phương pháp "*full angle*" ở trên chưa cho ta một mô hình biểu diễn tri thức tốt để có thể xây dựng một cơ sở tri thức và một ngôn ngữ khai báo bài toán một cách tự nhiên.

1.2.4 Phương pháp Wu

Phương pháp Wu là một phương pháp biểu diễn và chứng minh định lý hình học theo cách tiếp cận đại số. Phương pháp Wu cùng với các phát triển khác nhau của nó được trình bày trong các tài liệu [20], [21], [43] và [56]. Kỹ thuật cơ bản của các phương pháp này là biểu diễn các sự kiện dựa trên các đa thức và các tính toán trên tập nghiệm của các đa thức. Phương pháp nầy cho ta một biểu diễn khá đẹp về mặt lý thuyết toán học. Tuy nhiên nó cũng có nhiều hạn chế như các phương pháp "diện tích" và "*full angle*" trong nhu cầu xây dựng một hệ giải bài toán dựa trên tri thức. Có thể kể ra một số hạn chế của phương pháp Wu như (1) thực hiện thiết kế một thuật toán thực hành tính toán trên máy tính một cách hiệu quả rất khó, (2) Không thể làm cơ sở hay công cụ cho việc thiết kế một cách hệ thống một chương trình giải toán dựa trên tri thức, đặc biệt là chương trình cho lời giải tường minh phù hợp với việc học và dạy toán hình học và (3) Có những quan hệ trên các yếu tố hình học không thể diễn đạt bởi các đa thức.

1.2.5 Các phương pháp chứng minh hình học bằng máy tính

Tổng kết các công trình nghiên cứu về chứng minh tự động các bài toán hình học, S.C. Chou và các đồng tác giả của tài liệu [19] đã liệt kê các phương pháp khác nhau có thể sử dụng để chứng minh các bài toán hình học bằng máy tính. Các phương pháp này bao gồm:

- Phương pháp diện tích chủ yếu tập trung vào sự thẳng hàng và sự song song.
- Phương pháp hiệu Pythagoras (Pythagoras Difference) tập trung vào khảo sát sự vuông góc của đường thẳng và đường tròn. Hiệu Pythagoras còn có thể được sử dụng để khảo sát sự song song và có liên quan đến phương pháp diện tích.
- Phương pháp "*full angle*" như được nêu lên trong mục trên.
- Phương pháp "thể tích" và phương pháp hiệu Pythagoras cho không gian.
- Phương pháp vector sử dụng đại số tuyến tính và lý thuyết về không gian vector trong chứng minh định lý hình học.

Các phương pháp nầy đều tập trung vào việc tìm ra các đặc trưng về mặt toán học nhất là các lý thuyết đại số trừu tượng cho từng khía cạnh liên hệ thể hiện một loại sự kiện hình học nhất định để dựa vào đó thực hiện các chứng minh tự động. Điều nầy làm cho các phương pháp trở nên nặng nề về tính toán toán học trừu tượng và khó cài đặt trên máy tính. Nhưng hạn chế lớn nhất của các phương pháp nầy trong việc xây dựng một hệ giải toán thông minh là chúng không cho ta những mô hình biểu diễn tri thức tốt giúp xây dựng một cơ sở tri thức, bộ suy diễn và các thành phần khác của hệ thống.

1.2.6 Một số kết quả nghiên cứu xây dựng hệ giải toán hình học

Trong mục nầy sẽ đề cập đến một số nghiên cứu xây dựng hệ giải toán hình học được trình bày trong các tài liệu: [4], [5], và [7]. Trong [4] tác giả đã nêu lên một số nhận định về "bộ phân tích bài toán hỗ trợ cho việc giải các bài toán phổ thông" đặc biệt là việc đề cập đến cách phát sinh đối tượng và một số qui tắc phát sinh đối tượng trong hình học phẳng. Đây là một ý kiến hay và có nguồn gốc tự nhiên xuất phát từ tư duy giải toán thông thường trong hình học phẳng. Tuy nhiên tác giả chưa có một mô hình biểu diễn tri thức cho khái niệm "đối tượng" có thể sử dụng trong biểu diễn kiến thức hình học và cũng chưa xây dựng một mô hình biểu diễn tri thức chung cho việc xây dựng một cơ sở tri thức với một hệ thống khái niệm cơ bản cùng với các thông tin đa dạng kèm theo, các loại sự kiện và luật khác nhau trên các khái niệm. Từ đó cũng chưa xây dựng một mô hình tổng quát cho các dạng bài toán hình học khác nhau. Đây là một vấn đề tồn tại cũng giống như các phương pháp chứng minh định lý hình học của Wu, phương pháp "diện tích", phương pháp "thể tích" và các phương pháp khác được nói đến trong mục 1.2.5 ở trên.

Trong tài liệu [5] tác giả cũng cố gắng xây dựng một hệ chuyên gia giải các bài toán hình học phẳng ở phổ thông và liệt kê một số luật suy diễn cụ thể thể hiện một số định lý và một số qui tắc có thể dùng trong suy luận của con người để giải toán hình học. Một khía cạnh đáng quan tâm ở đây là ý tưởng về việc xây dựng một ngôn ngữ giao tiếp để khai báo đề bài toán. Ý tưởng nầy thật ra cũng đã được một số nhà nghiên cứu đề cập tới dưới dạng những ý tưởng khái quát trong lĩnh vực nghiên cứu về biểu diễn tri thức và suy diễn tự động. Tác giả có công đóng góp trong việc thể hiện cụ thể ý tưởng nầy đối với hệ giải bài toán hình học phẳng. Tuy nhiên, do chưa có một mô hình tri thức chung cho kiến thức hình học phẳng và chưa có mô hình tổng quát cho các dạng bài toán khác nhau nên thể hiện này còn mang tính cục bộ, thiếu tính hệ thống trong thiết kế chương trình giải toán. Hơn nữa, đa số luật đều dựa trên các quan hệ hình học trực quan và một số luật có thể gây ra sai sót trong xử lý suy luận. Ngoài ra tác giả cũng không đề cập đến sự phát sinh đối tượng và các luật liên quan đến tính toán.

Tác giả của tài liệu [7] cũng nghiên cứu việc biểu diễn kiến thức và xây dựng chương trình giải bài toán hình học phẳng với mục tiêu giải các bài toán hình học phổ thông và cho lời giải tường minh giống như con người. Cách biểu diễn được trình bày trong [7] cho mục tiêu này khá hợp lý cho các suy diễn định tính trên biểu diễn những quan hệ hình học. Tuy nhiên tác giả chưa nêu lên một biểu diễn đầy đủ hơn để có thể xây dựng một cơ sở tri thức cho chương trình và do đó cũng chưa xây dựng mô hình cho các lớp bài toán hình học khác nhau. Điều nầy là một hạn chế chung dẫn đến một số hạn chế cụ thể khác như:

- Chỉ xét một số luật suy diễn cụ thể trên các quan hệ hình học và ứng với mỗi luật phải viết riêng một thủ tục thi hành luật.
- Chưa có một cơ sở tri thức có thể hiệu chỉnh được và bộ suy diễn sẽ hoạt động dựa trên cơ sở tri thức.
- Chưa xem xét đến vấn đề tính toán.
- Chưa có một ngôn ngữ qui ước cho việc đặc tả các dạng bài toán khác nhau bao gồm cả những yêu cầu chứng minh cũng như những yêu cầu tính toán.

1.2.7 Một số sản phẩm phần mềm giải toán

Trong mục nầy sẽ đề cập đến một số phầm mềm cụ thể có liên quan đến tri thức và giải toán.

1.2.7.1Các chương trình tính toán hình học trong bộ phần mềm Engineering 2000

Các chương trình này gồm chương trình Geometrix version 1.0 của ByteSize CD-ROM, Inc (1998) và chương trình Geometry. Các chương trình này chỉ thực hiện các tính toán dữ liệu số cụ thể dựa vào một vài khung (frames) dữ liệu đơn giản. Ví dụ: Khung tính toán thể tích hình hộp chữ nhật khi biết các độ dài của các cạnh theo 3 chiều, các khung tính toán diện tích của một số hình phẳng đơn giản và tính toán thể tích của một số hình khối đơn giản. Có thể nói rằng các chương trình này chỉ làm một số bài tính toán số liệu hình học đơn giản mà không có sự suy luận giải một lớp bài toán tổng quát dựa trên tri thức.

1.2.7.2Chương trình StudyWorks

Chương trình StudyWorks của MathSoft, Inc. (1996) hỗ trợ soạn thảo và một số tính toán cho các tài liệu giáo khoa và khoa học bao gồm toán học, vật lý, thống kê, v.v... trong đó các hỗ trợ tính toán cũng được thiết kế theo các khung dữ liệu. Như vậy, cũng tương tự như chương trình Geometrix và Geometry ở trên, chương trình này chưa có một cơ sở tri thức và một bộ giải bài toán dựa trên tri thức có thể giải các lớp bài toán tổng quát khác nhau.

1.2.7.3Chương trình Math Express!

Chương trình Math Express version 1.0 của Aces Research, Inc. (1998). Sản phẩm này bao gồm các chương trình tính toán đại số, hình học và lượng giác. Chương trình cung cấp một số phần kiến thức giáo khoa cơ bản và liên quan đến mỗi phần kiến thức có một số câu hỏi trắc nghiệm để kiểm tra kiến thức. Như thế chương trình cũng chưa phải là một hệ giải toán dựa trên tri thức.

1.2.7.4 Phần mềm toán học MAPLE

Phần mềm MAPLE là một phần mềm đại số tính toán (Computer Algebra) khá mạnh trong đó hỗ trợ không chỉ các tính toán số mà cả các tính toán ký hiệu (symbolic computation). Nó có một nhân tính toán rất mạnh và một hệ thống thư viện tính toán gồm nhiều gói chương trình (package) cho các phần tính toán toán học khác nhau như đại số tuyến tính, giải tích, hình học, số học, v.v... Ngoài ra Maple còn cho phép khả năng lập trình với các cấu trúc dữ liệu trừu tượng để xây dựng những gói chương trình mới bổ sung vào hệ thống thư viện của nó. Tuy khả năng hỗ trợ tính toán và lập trình của Maple đạt đến trình độ rất cao nhưng chủ yếu là các thủ tục tính toán cho từng vấn đề đơn lẻ như giải một phương trình, tính đạo hàm của một hàm số, thực hiện các phép tính trên đa thức ... Maple chưa thật sự cài đặt một cơ sở tri thức cho các phần toán học khác nhau với các bộ suy luận giải toán dựa trên tri thức. Chẳng hạn như gói chương trình về hình học chỉ cung cấp một số thủ tục tính toán đơn giản như: tính diện tích hình vuông khi biết tọa độ 4 điểm đỉnh, tính độ dài của đoạn thẳng khi biết tọa độ 2 điểm đầu của đoạn thẳng, ...

Ngoài phần mềm toán học MAPLE, chúng ta cũng còn thấy nhiều phần mềm tính toán toán học khác như MATHEMATICA, MATHCAD, REDUCE, v.v... Các phần mềm này cũng như MAPLE chỉ cung cấp những đơn vị chương trình giúp thực hiện các thao tác tính toán riêng lẻ mà chưa có một bộ giải toán dựa trên một cơ sở tri thức thật sự. Người sử dụng các phần mềm nầy khi cần giải quyết một vấn đề nào đó như giải một bài toán tính toán về tam giác hay tứ giác, giải một bài toán khảo sát hàm số thì họ phải thiết kế một qui trình giải bài toán dưới dạng thuật giải và tự cài đặt thuật giải đó dựa vào ngôn ngữ lập trình được cho trong phần mềm.

1.3 Những Vấn đề mà Luận án tập trung Nghiên cứu, Giải quyết

Từ những trình bày tổng quan ở trên về cấu trúc của một hệ giải toán dựa trên tri thức mà trong đó 2 thành phần trung tâm là cơ sở tri thức và bộ suy diễn dựa trên tri thức. Đề tài mong muốn nghiên cứu, phát triển một số mô hình biểu diễn tri thức và các thuật giải để giải tự động các dạng bài toán khác nhau dựa trên tri thức. Cách tiếp cận được sử dụng là kết hợp các phương pháp biểu diễn tri thức đã có với những phát triển nhất định để tạo ra một số mô hình biểu diễn tri thức mới với phạm vi tri thức bao gồm nhiều dạng kiến thức đa dạng hơn, và như thế các mô hình biểu diễn tri thức này có thể được sử dụng như là cơ sở và là công cụ cho việc thiết kế cơ sở tri thức, bộ phận suy luận giải toán tự động cũng như thiết kế phần giao diện của chương trình. Cụ thể là luận văn tập trung xây dựng một số mô hình biểu diễn tri thức gồm:

1. Mô hình mạng suy diễn và tính toán.
2. Mô hình một đối tượng tính toán được gọi là một C-Object.
3. Mô hình tri thức về các C-Object, và mô hình mạng các C-Object.

Trên các mô hình biểu diễn tri thức này, một số thuật giải được xây dựng để có thể cài đặt các thủ tục giải bài toán dựa trên các kiến thức trong cơ sở tri thức. Các mô hình biểu diễn tri thức trên sẽ được sử dụng trong thiết kế và cài đặt một số chương trình giải tự động một số lớp bài toán về các tam giác, các tứ giác, các bài toán hình học phẳng, các bài toán hình học giải tích và một số bài toán trên các phản ứng hóa học.

CHƯƠNG 2

MẠNG SUY DIỄN-TÍNH TOÁN

2.1 DẪN NHẬP

Một trong những vấn đề hiện nay đang được quan tâm của "Trí Tuệ Nhân Tạo" là nghiên cứu các phương pháp biểu diễn và xử lý tri thức. Trên cơ sở đó có thể tạo ra những chương trình "thông minh" ở một mức độ nào đó. Có nhiều phương pháp biểu diễn tri thức đã được đề cập đến và đã được áp dụng. Trong chương nầy chúng ta xét đến một trường hợp của biểu diễn và xử lý tri thức theo mạng ngữ nghĩa: "Mạng suy diễn và tính toán". Các kết quả nghiên cứu về mô hình này đã được trình bày trong các bài báo [57], [63], [65] và [67].

Trong nhiều lĩnh vực ứng dụng ta thường gặp những vấn đề đặt ra dưới dạng như sau: Chúng ta phải thực hiện những tính toán hay suy diễn ra những yếu tố cần thiết nào đó từ một số yếu tố đã được biết trước. Để giải quyết vấn đề người ta phải vận dụng một số hiểu biết (tri thức) nào đó về những liên hệ giữa các yếu tố đang được xem xét. Những liên hệ cho phép ta có thể suy ra được một số yếu tố từ giả thiết đã biết một số yếu tố khác. Dưới đây là một số ví dụ minh họa.

<u>Ví dụ 2.1</u>: Giả sử ta đang quan tâm đến một số yếu tố trong một tam giác, chẳng hạn : 3 cạnh a, b, c; 3 góc tương ứng với 3 cạnh : α, β, γ; 3 đường cao tương ứng : ha, hb, hc; diện tích S của tam giác; nửa chu vi p của tam giác; bán kính đường tròn nội tiếp r của tam giác. Giữa 12 yếu tố trên có các công thức thể hiện những mối quan hệ giúp chúng ta có thể giải quyết được một số vấn đề tính toán đặt ra. Trong tam giác có thể kể ra một số quan hệ dưới dạng công thức sau đây :

- Liên hệ giữa 3 góc : $\alpha + \beta + \gamma = \pi$ (radian).
- Định lý cosin : $a^2 = b^2 + c^2 - 2.b.c.\cos\alpha$; $b^2 = a^2 + c^2 - 2.a.c.\cos\beta$;

  $c^2 = a^2 + b^2 - 2.a.b.\cos\gamma$
- Định lý Sin : $\frac{a}{\sin\alpha} = \frac{b}{\sin\beta}$; $\frac{c}{\sin\gamma} = \frac{b}{\sin\beta}$; $\frac{a}{\sin\alpha} = \frac{c}{\sin\gamma}$
- Liên hệ giữa nửa chu vi và 3 cạnh : $2.p = a + b + c$
- Công thức tính diện tích theo 3 cạnh (công thức Heron):

  $S = \sqrt{p(p-a)(p-b)(p-c)}$
- Một số công thức tính diện tích :

  $S = a.h_a/2$; $S = b.h_b/2$; $S = c.h_c/2$; $S = p.r$

Đối với bài toán giải tam giác với 12 yếu tố được xem xét, nếu xét các đề tính diện tích tam giác với 3 yếu tố được cho trước thì số bài toán cụ thể lên đến: $C_{12}^3 = 220$; đó là chưa kể đến những yêu cầu tính toán các yếu tố khác nữa, và như thế số bài toán tính ra sẽ rất lớn. Nếu như ứng với mỗi bài toán ta cài đặt một thuật giải để giải quyết thì không thể chấp nhận được. Tuy nhiên có thể thấy rằng mặc dù có quá nhiều bài toán cụ thể khác nhau nhưng để giải quyết chúng ta chỉ cần áp dụng một số công thức nào đó trong số các công thức thể hiện những quan hệ giữa các yếu tố đang được xem xét. Nếu chúng ta có cách để biểu diễn và xử lý các yếu tố và các quan hệ giữa các yếu tố thì có thể tìm một số thuật giải cài đặt được để giải các bài toán một cách hiệu quả.

Ví dụ 2.2: Một vật thể có khối lượng m chuyển động thẳng với gia tốc không thay đổi là a trong một khoảng thời gian tính từ thời điểm t_1 đến thời điểm t_2. Vận tốc ban đầu của vật thể là v_1, vận tốc ở thời điểm cuối là v_2, và vận tốc trung bình là v. Khoảng cách giữa điểm đầu và điểm cuối là Δs. Lực tác động của chuyển động là f. Độ biến thiên vận tốc giữa 2 thời điểm là Δv, và độ biến thiên thời gian là Δt. Ngoài ra còn có một số yếu tố khác nữa của chuyển động vật thể có thể được quan tâm. Tương tự như trong ví dụ 2.1, để giải những bài toán về chuyển động nầy chúng ta phải sử dụng một số công thức liên hệ giữa các yếu tố của chuyển động, chẳng hạn như :

$f = m * a;$

$\Delta v = a*\Delta t;$

$\Delta s = v*\Delta t;$

$2*v = v_1 + v_2;$

$\Delta v = v_2 - v_1;$

$\Delta t = t_2 - t_1;$

Ví dụ 2.3: Trong hóa học chúng ta thường phải sử dụng các phản ứng hóa học để điều chế các chất nầy từ các chất khác. Loại vấn đề nầy cũng cho ta một dạng tương tự như trong 2 ví dụ trên : Cho trước một số chất hóa học, hãy tìm cách điều chế ra một hay một số chất nào đó.

Nói tóm lại, có nhiều vấn đề trong các lĩnh vực ứng dụng khác nhau đặt ra dưới dạng một "mạng" các yếu tố, trong đó các yếu tố có những mối liên hệ (hay quan hệ) cho phép ta có thể suy ra được một số yếu tố nầy từ một số yếu tố khác. Mô hình mạng suy diễn và tính toán là một sự khái quát cho một dạng tri thức dùng cho việc biểu diễn tri thức và thiết kế các chương trình giải toán tự động, chẳng hạn như chương trình giải các lớp bài toán tương tự như trong các ví dụ trên.

2.2 MẠNG SUY DIỄN VÀ CÁC VẤN ĐỀ CƠ BẢN

Trong mục nầy chúng ta xét một mạng suy diễn gồm một tập hợp các biến cùng với một tập các quan hệ suy diễn giữa các biến. Trong ứng dụng cụ thể mỗi biến và giá trị của nó thường gắn liền với một khái niệm cụ thể về sự vật, mỗi quan hệ thể hiện một sự tri thức về sự vật.

2.2.1 Quan hệ và luật suy diễn

Cho M = $\{x_1,x_2,...,x_m\}$ là một tập hợp các biến có thể lấy giá trị trong các miền xác định tương ứng $D_1,D_2,...,D_m$. Đối với mỗi *quan hệ* $R \subseteq D_1xD_2x...xD_m$ trên các tập hợp $D_1,D_2,...,D_m$ ta nói rằng quan hệ nầy liên kết các biến $x_1,x_2,...,x_m$, và ký hiệu là $R(x_1,x_2,...,x_m)$ hay vắn tắt là R(x) (ký hiệu x dùng để chỉ bộ biến $< x_1,x_2,...,x_m >$). Ta sẽ xét các quan hệ R(x) xác định một (hay một số) ánh xạ :

$$f_{R,u,v} : D_u \rightarrow D_v,$$

trong đó u,v là các bộ biến và $u \subseteq x$, $v \subseteq x$; D_u và D_v là tích của các miền xác định tương ứng của các biến trong u và trong v. Quan hệ như thế được gọi là *quan hệ suy diễn*. Có thể thấy rằng quan hệ suy diễn R(x) có thể được biểu diễn bởi một (hay một số) ánh xạ $f_{R,u,v}$ với $u \cup v = x$, và ta viết :

$$f_{R,u,v} : u \rightarrow v,$$

hay vắn tắt là: $f : u \rightarrow v$.

Cách ký hiệu trên bao hàm ý nghĩa như là một *luật suy diễn*: ta có thể xác định hay suy ra được các biến thuộc v khi biết được các biến thuộc u. Trong phần sau ta xét các quan hệ xác định các luật suy diễn có dạng:

$$f : u \rightarrow v,$$

trong đó $u \cap v = \varnothing$ (tập rỗng). Ngoài ra, trong trường hợp cần nói rõ ta viết u(f) thay cho u, v(f) thay cho v. Ta gọi một quan hệ là *quan hệ đối xứng* có hạng (rank) bằng một số nguyên dương k khi quan hệ đó giúp ta có thể tính được k biến bất kỳ từ m-k biến kia (ở đây x là bộ gồm m biến $< x_1,x_2,...,x_m >$). Đối với các quan hệ không phải là đối xứng có hạng k, không làm mất tính tổng quát, ta có thể giả sử quan hệ xác định duy nhất một luật f với tập biến vào là u(f) và tập biến ra là v(f); ta gọi loại quan hệ nầy là quan hệ không đối xứng xác định một luật dẫn, hay gọi vắn tắt là *quan hệ không đối xứng*.

Nhận xét: Một quan hệ không đối xứng hạng k có thể được viết thành k quan hệ không đối xứng có hạng 1. Nếu biểu diễn một quan hệ đối xứng có hạng k thành các quan hệ đối xứng có hạng là 1 thì số quan hệ có hạng 1 bằng :

$$m\,C_m^{m-k} = m\,C_m^{k}$$

Dưới đây là một vài ví dụ về các quan hệ suy diễn.

Ví dụ 2.4: quan hệ f giữa 3 góc A, B, C trong tam giác ABC cho bởi hệ thức:

$$A+B+C = 180 \qquad \text{(đơn vị: độ)}$$

Quan hệ f giữa 3 góc trong một tam giác trên đây là một quan hệ đối xứng có hạng 1. Quan hệ nầy bao hàm 3 luật suy diễn:

$A, B \Rightarrow C$

$A, C \Rightarrow B$

$C, B \Rightarrow A$

Ví dụ 2.5: quan hệ f giữa nửa chu vi p với các độ dài của 3 cạnh a, b, c:

$2*p = a + b + c$

Ví dụ 2.6: quan hệ f giữa n biến $x_1, x_2, ..., x_n$ được cho dưới dạng một hệ phương trình tuyến tính có nghiệm. Trong trường hợp nầy f là một quan hệ có hạng k bằng hạng của ma trận hệ số của hệ phương trình.

2.2.2 Mạng suy diễn

Từ sự trình bày trên ta có thể nêu lên định nghĩa hình thức cho mạng suy diễn như sau:

- **Định nghĩa 2.1**: Ta gọi một *mạng suy diễn*, viết tắt là **MSD**, là một cấu trúc (M,F) gồm 2 tập hợp:

  (1) $M = \{x_1,x_2,...,x_n\}$, là tập hợp các thuộc tính hay các yếu tố lấy giá trị trong các miền xác định nào đó.

  (2) $F = \{f_1,f_2,...,f_m\}$, là tập hợp các luật suy diễn có dạng:

$$f : u(f) \rightarrow v(f)$$

trong đó u(f) và v(f) là các tập hợp con khác rỗng của M sao cho

$u(f) \cap v(f) = \varnothing$.

Đối với mỗi $f \in F$, ta ký hiệu M(f) là tập các biến có liên hệ trong quan hệ f, nghĩa là $M(f) = u(f) \cup v(f)$.

Nhận xét rằng một mạng (M,R) với R là một tập các quan hệ suy diễn cũng có thể được xem như một mạng suy diễn bằng cách thay tập R bởi tập F gồm tất cả các luật suy diễn xác định bởi các quan hệ suy diễn.

Ví dụ 2.7 :

Trong ví dụ 2.4 ở trên, ta có $M(f) = \{A,B,C\}$.

Trong ví dụ 2.5 ở trên, ta có $M(f) = \{a,b,c,p\}$.

Trong ví dụ 2.6 ở trên, ta có $M(f) = \{x_1, x_2, ..., x_n\}$.

Ví dụ 2.8: Mạng suy diễn cho một hình chữ nhật. Việc tính toán trên một hình chữ nhật liên quan đến một số yếu tố của hình chữ nhật như sau :

b_1, b_2 : hai cạnh của hình chữ nhật;

d : đường chéo của hình chữ nhật;

S : diện tích của hình chữ nhật;

p : chu vi của hình chữ nhật;

trong đó mỗi biến đều có giá trị thuộc tập các số thực dương. Giữa các biến ta đã biết có các quan hệ tính toán sau đây:

f_1 : $S = b_1 * b_2$;

f_2 : $p = 2 * (b_1 + b_2)$;

f_3 : $d^2 = b_1^2 + b_2^2$;

Về mặt suy luận, các quan hệ nầy đều có thể xem là các quan hệ suy diễn đối xứng có hạng là 1. Như vậy tập biến và tập quan hệ của mạng nầy là :

$M = \{b_1, b_2, d, s, p\}$,

$R = \{f_1, f_2, f_3\}$.

Mạng (M,R) nầy tương ứng với mạng suy diễn (M, F) với F là tập các luật suy diễn sau đây:

$b_1, b_2 \Rightarrow S$

$S, b_2 \Rightarrow b_1$

$S, b_1 \Rightarrow b_2$

$b_1, b_2 \Rightarrow p$

$p, b_2 \Rightarrow b_1$

$p, b_1 \Rightarrow b_2$

$b_1, b_2 \Rightarrow d$

$d, b_2 \Rightarrow b_1$

$d, b_1 \Rightarrow b_2$

2.2.3 Các vấn đề cơ bản trên mạng suy diễn

Cho một mạng suy diễn (M,F) với M là tập các thuộc tính (hay các biến) và F là tập các quan hệ suy diễn hay các luật suy diễn. Giả sử có một tập biến $A \subseteq M$ đã được xác định (tức là tập gồm các biến đã biết trước), và B là một tập biến bất kỳ trong M.

- Vấn đề 1: Có thể xác định được (hay suy ra) tập B từ tập A nhờ các quan hệ trong F hay không? Nói cách khác, ta có thể tính được giá trị của các biến thuộc B với giả thiết đã biết giá trị của các biến thuộc A hay không?

- Vấn đề 2: Nếu có thể suy ra được B từ A thì quá trình suy diễn như thế nào? Trong trường hợp có nhiều cách suy diễn khác nhau thì cách suy diễn nào là tốt nhất?

- Vấn đề 3: Trong trường hợp không thể xác định được B, thì cần cho thêm điều kiện gì để có thể xác định được B.

Bài toán xác định B từ A trên mạng suy diễn (M,F) được viết dưới dạng:

$$A \rightarrow B$$

trong đó A được gọi là giả thiết, B được gọi là mục tiêu (hay tập biến cần xác định) của bài toán. Trường hợp tập B chỉ gồm có một phần tử b, ta viết vắn tắt bài toán trên là: $A \rightarrow b$.

Việc tìm lời giải cho bài toán là tìm ra một dãy quan hệ suy diễn để có thể áp dụng suy ra được B từ A. Điều nầy cũng có nghĩa là tìm ra được một quá trình tính toán hay suy diễn để giải bài toán. Trong việc tìm lời giải cho bài toán chúng ta cần xét một dãy các quan hệ suy diễn hay các luật suy diễn nào đó xem có thể suy ra được các biến từ một tập biến cho trước nhờ dãy quan hệ suy diễn nầy hay không. Từ đó chúng ta có định nghĩa sau đây.

- **<u>Định nghĩa 2.2</u>**: Giả sử (M,F) là một mạng suy diễn và A là một tập con của M. Một luật suy diễn $u \rightarrow v$ được được gọi là áp dụng được trên A khi $u \subset A$. Một quan hệ suy diễn được gọi là áp dụng được trên A khi nó xác định một luật suy diễn áp dụng được trên A. Cho D = $\{f_1, f_2, ..., f_k\}$ là một dãy các quan hệ suy diễn (hay luật suy diễn) của mạng suy diễn (M,F), ta nói dãy D là *áp dụng được* trên tập A khi và chỉ khi ta có thể lần lượt áp dụng được các quan hệ $f_1, f_2, ..., f_k$ xuất phát từ giả thiết A.

Trong định nghĩa trên, đặt : $A_0 = A$, $A_1 = A_0 \cup M(f_1)$, . . . , $A_k = A_{k-1} \cup M(f_k)$, và ký hiệu A_k là **D(A)**. Trong trường hợp D là một dãy luật suy diễn tùy ý ta vẫn ký hiệu D(A) là tập biến đạt được khi lần lượt áp dụng các quan hệ trong dãy D (nếu được). Có thể nói rằng D(A) là sự mở rộng của tập A nhờ áp dụng dãy quan hệ D.

<u>Thuật toán tính D(A)</u> :

<u>Nhập</u> : Mạng suy diễn (M,F), $A \subseteq M$,

dãy các quan hệ suy diễn D = $\{f_1, f_2, ..., f_m\}$.

<u>Xuất</u> : D(A).

<u>Thuật toán</u> :

1. A' ← A;
2. **for** i=1 to m **do**

    if f_i áp dụng được trên A' **then**

        A' ← A' ∪ M(f_i);
3. D(A) ← A'

- **<u>Định nghĩa 2.3</u>**: Ta nói rằng một dãy các quan hệ suy diễn D = $\{f_1, f_2, ..., f_k\} \subseteq F$ là một *lời giải* của bài toán A → B nếu như ta lần lượt áp dụng các quan hệ f_i (i=1,...,k) xuất phát từ giả thiết A thì sẽ suy ra được các biến thuộc B. Nói cách khác D là một lời giải của bài toán khi D(A) ⊃ B. Bài toán A → B được gọi là *giải được* khi nó có một lời giải.

    Lời giải $\{f_1, f_2, ..., f_k\}$ được gọi là *lời giải tốt* nếu không thể bỏ bớt một số bước tính toán trong quá trình giải, tức là không thể bỏ bớt một số quan hệ trong lời giải. Lời giải được gọi là *một lời giải ngắn nhất* khi nó có số bước suy diễn thấp nhất, tức là số quan hệ suy diễn áp dụng trong suy diễn là ít nhất.

2.3 TÌM LỜI GIẢI

Xét bài toán A → B trên mạng suy diễn (M,F). Trong mục nầy chúng ta sẽ trình bày cách giải quyết các vấn đề cơ bản sau đây:

- Khảo sát tính giải được của bài toán suy diễn.
- Tìm một lời giải tốt cho bài toán suy diễn và phân tích quá trình suy diễn.

2.3.1 Tính giải được của bài toán

Trong mục nầy chúng ta nêu lên một khái niệm có liên quan đến tính giải được của bài toán: bao đóng của một tập hợp biến trên một mạng suy diễn.

- **Định nghĩa 2.4**: Cho mạng suy diễn (M,F), và A là một tập con của M. Có thể thấy rằng có duy nhất một tập hợp B lớn nhất $\subseteq$ M sao cho bài toán A$\rightarrow$B là giải được, và tập hợp B nầy được gọi là *bao đóng* của A trên mạng (M, F). Một cách trực quan, có thể nói bao đóng của A là sự mở rộng tối đa của A trên mạng suy diễn (M,F). Ký hiệu bao đóng của A là $\overline{A}$, ta có thể kiểm tra dễ dàng các tính chất liên quan đến bao đóng trong mệnh đề dưới đây.

Mệnh đề 2.1. Cho A và B là hai tập con của M. Ta có:

(1) $\overline{A} \supseteq A$.

(2) $\overline{\overline{A}} = \overline{A}$.

(3) $A \subseteq B \Rightarrow \overline{A} \subseteq \overline{B}$

(4) $\overline{A \cap B} \subseteq \overline{A} \cap \overline{B}$

(5) $\overline{A \cup B} \supseteq \overline{A} \cup \overline{B}$

Đối với tính giải được của bài toán, ta có thể dễ dàng kiểm chứng mệnh đề sau:

Mệnh đề 2.2.

(1) Bài toán $A \rightarrow B$ là giải được khi và chỉ khi các bài toán $A \rightarrow b$ là giải được với mọi $b \in B$.

(2) Nếu $A \rightarrow B$ và $B \rightarrow C$ là các bài toán giải được thì bài toán $A \rightarrow C$ cũng giải được. Hơn nữa, nếu $\{f_1, f_2, ..., f_m\}$ và $\{g_1, g_2, ..., g_p\}$ lần lượt là lời giải của bài toán $A \rightarrow B$ và bài toán $B \rightarrow C$ thì $\{f_1, f_2, ..., f_m, g_1, g_2, ..., g_p\}$ là một lời giải của bài toán $A \rightarrow C$.

(3) Nếu bài toán $A \rightarrow B$ là giải được và B' là một tập con của B thì $A \rightarrow B'$ cũng là một bài toán giải được. Hơn nữa, nếu $\{f_1, f_2, ..., f_m\}$ là một lời giải của bài toán $A \rightarrow B$ thì đó cũng là một lời giải của bài toán $A \rightarrow B'$.

Từ khái niệm bao đóng đã nói ở trên ta cũng có các định lý sau:

Định lý 2.1 Trên một mạng suy diễn (M,F), bài toán $A \rightarrow B$ là giải được khi và chỉ khi $B \subseteq \overline{A}$.

Từ định lý nầy, ta có thể kiểm tra tính giải được của bài toán $A \rightarrow B$ bằng cách tìm bao đóng của tập A rồi xét xem B có bao hàm trong $\overline{A}$ hay không.

Mệnh đề 2.3 : Cho một dãy quan hệ $D = \{f_1, f_2, ..., f_k\} \subseteq F$, $A \subseteq M$. Đặt :

$A_0 = A$, $A_1 = A_0 \cup M(f_1)$, ..., $A_k = A_{k-1} \cup M(f_k)$. Ta có các điều sau đây là tương đương :

(1) Dãy D áp được trên A.

(2) Với mọi i=1, ..., k ta có:

$\text{Card}\,(M(f_i) \setminus A_{i-1}) \leq r(f_i)$ nếu f_i là quan hệ đối xứng,

$M(f_i) \setminus A_{i-1} \subseteq v(f_i)$ nếu f_i là quan hệ không đối xứng.

(ký hiệu Card (X) chỉ số phần tử của tập X).

Ghi chú: Dựa vào mệnh đề 2.3 ta có một thuật toán để kiểm tra tính áp dụng được của một dãy quan hệ D trên một tập biến A.

Định lý 2.2 Trên một mạng suy diễn (M,F), giả sử A, B là hai tập con của M. Ta có các điều sau đây là tương đương:

(1) $B \subseteq \overline{A}$.

(2) Có một dãy quan hệ $D = \{f_1, f_2, ..., f_k\} \subseteq F$ thỏa các điều kiện :

(a) D áp dụng được trên A.

(b) $D(A) \supseteq B$.

Chứng minh : Giả sử có (1), tức là $B \subseteq \overline{A}$. Khi đó bài toán $A \rightarrow B$ là giải được. Do đó có một dãy quan hệ $\{f_1, f_2, ..., f_k\} \subseteq F$ sao cho khi ta lần lượt áp dụng các quan hệ f_i (i=1,...,k) xuất phát từ giả thiết A thì sẽ tính được các biến thuộc B. Dễ dàng thấy rằng dãy $\{f_1, f_2, ..., f_k\}$ nầy thỏa các điều kiện (2).

Đảo lại, giả sử có (2). Với các điều kiện có được bởi (2) ta thấy $\{f_i\}$ là lời giải của vấn đề $A_{i-1} \rightarrow A_i$, với mọi i = 1,2, ..., k. Từ mệnh đề 3.2 suy ra bài toán $A_0 \rightarrow A_k$ là giải được. Do đó bài toán $A \rightarrow B$ cũng giải được, suy ra $B \subseteq \overline{A}$ theo định lý 2.1.

Nhận xét :

- dãy quan hệ $\{f_1, f_2, ..., f_k\}$ trong định lý trên là một lời giải của bài toán $A \rightarrow B$ trên mạng (M,F).
- Trong lời giải $\{f_1, f_2, ..., f_k\}$ ta có thể bỏ bớt những f_i nào mà $M(f_i) \subseteq D_{i-1}(A)$, với $D_{i-1} = \{f_1, f_2, ..., f_{i-1}\}$.

Dưới đây là thuật toán cho phép xác định bao đóng của tập hợp $A \subseteq M$. Trong thuật toán nầy chúng ta thử áp dụng các quan hệ $f \in F$ để tìm dần những biến thuộc M có thể suy ra được từ A; cuối cùng sẽ được bao đóng của A.

Thuật toán 2.1 tìm bao đóng của tập $A \subseteq M$:

Nhập : Mạng tính toán (M,F), $A \subseteq M$.

Xuất : $\overline{A}$

Thuật toán :

1. $B \leftarrow A$;
2. **Repeat**

   $B1 \leftarrow B$;

   for $f \in F$ **do**

   if (f đối xứng **and** Card $(M(f) \setminus B) \leq r(f)$) **or**

(f không đối xứng **and** $M(f) \setminus B \subseteq v(f)$) **then**

begin

$B \leftarrow B \cup M(f)$;

$F \leftarrow F \setminus \{f\}$; *// loại f khỏi lần xem xét sau*

end;

Until $B = B1$;

3. $\overline{A} \leftarrow B$;

Ghi chú : Trên đây ta đã nêu lên đặc trưng cho tính giải được của bài toán trên một mạng suy diễn và chỉ ra thuật toán để kiểm tra khi nào bài toán là giải được. Ngoài ra chúng ta sẽ còn nêu lên cách để kiểm định giả thiết của bài toán để trong trường hợp bài toán chưa đủ giả thiết có thể bổ sung thêm nếu được.

2.3.2 Lời giải của bài toán

Ở trên ta đã nêu lên cách xác định tính giải được của bài toán. Tiếp theo, ta sẽ trình bày cách tìm ra lời giải cho bài toán $A \rightarrow B$ trên mạng suy diễn (M, F). Trước hết từ nhận xét sau định lý 2.2 ta có mệnh đề sau đây:

Mệnh đề 2.4: Dãy quan hệ suy diễn D là một lời giải của bài toán $A \rightarrow B$ khi và chỉ khi D áp dụng được trên A và $D(A) \supseteq B$.

Do mệnh đề trên, để tìm một lời giải ta có thể làm như sau: Xuất phát từ giả thiết A, ta thử áp dụng các quan hệ để mở rộng dần tập các biến được xác định (được biết); và quá trình nầy tạo ra một sự lan truyền tính xác định trên tập các biến cho đến khi đạt đến tập biến B. Dưới đây là thuật toán tìm một lời giải cho bài toán $A \rightarrow B$ trên mạng (M,F).

Thuật toán 2.2 Tìm một lời giải cho bài toán $A \rightarrow B$:

Nhập : Mạng suy diễn (M, F),

tập giả thiết $A \subseteq M$,

tập biến cần tính $B \subseteq M$.

<u>Xuất</u> : lời giải cho bài toán $A \to B$

<u>Thuật toán</u> :

1. Solution ← empty; // Solution *là dãy các quan hệ sẽ áp dụng*
2. **if** $B \subseteq A$ **then**

    begin

    Solution_found ← true; // *biến* Solution_found = true

    // khi bài toán là giải được

    goto bước 4;

    end

   else

    Solution_found ← false;

3. **Repeat**

    Aold ← A;

    Chọn ra một $f \in F$ chưa xem xét;

    while not Solution_found **and** (chọn được f) **do**

     begin

     if (f đối xứng **and** $0 < \text{Card}(M(f) \setminus A) \leq r(f)$) **or**

      (f không đối xứng **and** $\varnothing \neq M(f) \setminus A \subseteq v(f)$) **then**

      begin

      $A \leftarrow A \cup M(f)$;

      $\text{Solution} \leftarrow \text{Solution} \cup \{f\}$;

      end;

     if $B \subseteq A$ **then**

      Solution_found ← true;

Chọn ra một $f \in F$ chưa xem xét;

end; { while }

Until Solution_found **or** (A = Aold);

4. **if not** Solution_found **then**

    Bài toán không có lời giải;

    else

    Solution là một lời giải;

Ghi chú :

1. Về sau, khi cần trình bày quá trình giải (hay bài giải) ta có thể xuất phát từ lời giải tìm được dưới dạng một dãy các quan hệ để xây dựng bài giải.
2. Lời giải (nếu có) tìm được trong thuật toán trên chưa chắc là một lời giải tốt. Ta có thể bổ sung thêm cho thuật toán ở trên một thuật toán để tìm một lời giải tốt từ một lời giải đã biết nhưng chưa chắc là tốt. Thuật toán sẽ dựa trên định lý được trình bày tiếp theo đây.

Định lý 2.3 Cho $D=\{f_1, f_2, ..., f_m\}$ là một lời giải của bài toán $A \rightarrow B$. Ứng với mỗi i=1,...,m đặt $D_i = \{f_1, f_2, ..., f_i\}$, $D_0 = \varnothing$. Ta xây dựng một họ các dãy con $S_m, S_{m-1}, ..., S_2, S_1$ của dãy D như sau :

$S_m = \varnothing$ nếu D_{m-1} là một lời giải,

$S_m = \{f_m\}$ nếu D_{m-1} không là một lời giải,

$S_i = S_{i+1}$ nếu $D_{i-1} \cup S_{i+1}$ là một lời giải,

$S_i = \{f_i\} \cup S_{i+1}$ nếu $D_{i-1} \cup S_{i+1}$ không là một lời giải,

với mọi i = m-1, m-2, ..., 2, 1.

Khi đó ta có :

(1) $S_m \subseteq S_{m-1} \subseteq ... \subseteq S_2 \subseteq S_1$.

(2) $D_{i-1} \cup S_i$ là một lời giải của bài toán $A \rightarrow B$ với mọi i=m, ..., 2, 1.

(3) Nếu S'_i là một dãy con thật sự của S_i thì $D_{i-1} \cup S'_i$ không phải là một lời giải của bài toán $A \to B$ với mọi i.

(4) S_1 là một lời giải tốt của bài toán $A \to B$.

Từ định lý trên ta có một thuật toán tìm lời giải tốt từ một lời giải đã biết sau đây:

Thuật toán 2.3 Tìm một lời giải tốt từ một lời giải đã biết.

Nhập : Mạng suy diễn (M,F),

lời giải $\{f_1, f_2, ..., f_m\}$ của bài toán $A \to B$.

Xuất : lời giải tốt cho bài toán $A \to B$

Thuật toán :

1. $D \leftarrow \{f_1, f_2, ..., f_m\}$;
2. **for** i=m **downto** 1 **do**

   if $D \setminus \{f_i\}$ là một lời giải **then**

   $D \leftarrow D \setminus \{f_i\}$;
3. D là một lời giải tốt.

Trong thuật toán 2.3 có sử dụng việc kiểm tra một dãy quan hệ có phải là lời giải hay không. Việc kiểm tra nầy có thể được thực hiện nhờ thuật toán sau đây:

Thuật toán kiểm tra lời giải cho bài toán :

Nhập : Mạng suy diễn (M,F),

bài toán $A \to B$,

dãy các quan hệ $\{f_1, f_2, ..., f_m\}$.

Xuất : thông tin cho biết $\{f_1, f_2, ..., f_m\}$ có phải là lời giải của bài toán $A \to B$ hay không.

Thuật toán :

1. **for** i=1 **to** m **do**

if (f_i đối xứng **and** Card $(M(f_i) \setminus A) \leq r(f_i)$) **or**

(f_i không đối xứng **and** $M(f_i) \setminus A \subseteq v(f_i)$) **then**

$A \leftarrow A \cup M(f_i)$;

2. **if** $A \supseteq B$ **then**

$\{f_1, f_2, ..., f_m\}$ là lời giải

else

$\{f_1, f_2, ..., f_m\}$ không là lời giải;

Ở trên ta đã có một thuật toán tổng quát để tìm lời giải tốt cho bài toán khi đã biết trước một lời giải. Thật ra, ta có thể áp dụng một thuật toán khác để tìm một lời giải tốt từ một lời giải biết trước với mức độ tính toán ít hơn. Theo thuật toán nầy, ta lần lượt xem xét các quan hệ trong tập lời giải đã biết và chọn ra các quan hệ để đưa vào một lời giải mới sao cho trong lời giải mới nầy không thể bớt ra bất kỳ một quan hệ nào.

Ví dụ 2.9: Bây giờ ta xét một ví dụ cụ thể để minh họa cho các thuật toán trên.

Cho tam giác ABC có cạnh a và 2 góc kề là β, γ được cho trước.

Hãy xác định (hay suy ra) S của tam giác.

Để tìm ra lời giải cho bài toán trước hết ta xét mạng suy diễn của tam giác. Mạng suy diễn nầy gồm :

1. Tập biến $M = \{a, b, c, \alpha, \beta, \gamma, h_a, h_b, h_c, S, p, R, r, ...\}$,

trong đó a,b,c là 3 cạnh; α, β, γ là 3 góc tương ứng với 3 cạnh; h_a, h_b, h_c là 3 đường cao; S là diện tích tam giác; p là nửa chu vi; R là bán kính đường tròn ngoại tiếp tam giác; r là bán kính đường tròn nội tiếp tam giác, v.v...

2. Các quan hệ suy diễn thể hiện bởi các công thức sau đây:

f_1 : $\alpha + \beta + \gamma = 180$

f_2 : $\dfrac{a}{\sin\alpha} = \dfrac{b}{\sin\beta}$

f_3 : $\dfrac{c}{\sin\gamma} = \dfrac{b}{\sin\beta}$

f_4 : $\dfrac{a}{\sin\alpha} = \dfrac{c}{\sin\gamma}$

f_5 : $p = (a+b+c)/2$

f_6 : $S = a.h_a / 2$

f_7 : $S = b.h_b / 2$

f_8 : $S = c.h_c / 2$

f_9 : $S = a.b.\sin\gamma / 2$

f_{10} : $S = b.c.\sin\alpha / 2$

f_{11} : $S = c.a.\sin\beta / 2$

f_{12} : $S = \sqrt{p(p-a)(p-b)(p-c)}$

v.v ...

Mục tiêu của bài toán là suy ra S (diện tích của tam giác).

Theo đề bài ta có giả thiết là : $A = \{a, \beta, \gamma\}$, và tập biến cần xác định là $B = \{S\}$.

Áp dụng thuật toán tìm lời giải (thuật toán 2.2) sẽ được một lời giải cho bài toán là dãy quan hệ suy diễn sau:

$\{f_1, f_2, f_3, f_5, f_9\}$.

Xuất phát từ tập biến A, lần lượt áp dụng các quan hệ trong lời giải ta có tập các biến được xác định mở rộng dần đến khi S được xác định :

$$\{a, \beta, \gamma\} \xrightarrow{f_1} \{a, \beta, \gamma, \alpha\} \xrightarrow{f_2} \{a, \beta, \gamma, \alpha, b\} \xrightarrow{f_3} \{a, \beta, \gamma, \alpha, b, c\}$$
$$\xrightarrow{f_5} \{a, \beta, \gamma, \alpha, b, c, p\} \xrightarrow{f_9} \{a, \beta, \gamma, \alpha, b, c, p, S\}.$$

Có thể nhận thấy rằng lời giải nầy không phải là lời giải tốt vì có bước suy diễn thừa, chẳng hạn là f_5. Thuật toán 2.3 sẽ lọc ra từ lời giải trên một lời giải tốt là $\{f_1, f_2, f_9\}$:

$$\{a, \beta, \gamma\} \xrightarrow{f_1} \{a, \beta, \gamma, \alpha\} \xrightarrow{f_2} \{a, \beta, \gamma, \alpha, b\} \xrightarrow{f_9} \{a, \beta, \gamma, \alpha, b, S\}.$$

Theo lời giải nầy, ta có quá trình suy diễn như sau :

bước 1: Xác định α (áp dụng f_1).

bước 2: Xác định b (áp dụng f_2).

bước 3: Xác định S (áp dụng f_9).

2.3.3 Định lý về sự phân tích quá trình giải

Xét bài toán $A \rightarrow B$ trên mạng suy diễn (M,F). Trong các mục trên đã trình bày một số phương pháp để xác định tính giải được của bài toán, tìm ra một lời giải tốt cho bài toán. Trong mục nầy ta nêu lên một cách xây dựng quá trình giải từ một lời giải đã biết. Đối với một lời giải, rất có khả năng một quan hệ nào đó dẫn tới việc tính toán một số biến thừa, tức là các biến tính ra mà không có sử dụng cho các bước tính phía sau. Do đó, chúng ta cần xem xét quá trình áp dụng các quan hệ trong lời giải và chỉ tính toán các biến thật sự cần thiết cho quá trình giải theo lời giải. Định lý sau đây cho ta một sự phân tích tập các biến được xác định theo lời giải và trên cơ sở đó có thể xây dựng quá trình suy diễn để giải quyết bài toán.

Định lý 2.4 Cho $\{f_1, f_2, ..., f_m\}$ là một lời giải tốt cho bài toán $A \rightarrow B$ trên một mạng suy diễn (M, F). Đặt :

$$A_0 = A,\ A_i = \{f_1, f_2, ..., f_i\}(A), \text{ với mọi } i=1,...,m.$$

Khi đó có một dãy $\{B_0, B_1, ..., B_{m-1}, B_m\}$, thỏa các điều kiện sau đây:

(1) $B_m = B$.

(2) $B_i \subseteq A_i$, với mọi i=0,1,...,m.

(3) Với mọi i=1,...,m, $\{f_i\}$ là lời giải của bài toán $B_{i-1} \rightarrow B_i$ nhưng không phải là lời giải của bài toán $G \rightarrow B_i$, trong đó G là một tập con thật sự tùy ý của B_{i-1}.

<u>Chứng minh</u> : Ta xây dựng dãy $\{B_0, B_1, ..., B_{m-1}, B_m\}$ bằng cách đặt: $B_m = B$, và ứng với mỗi $i < m$, đặt:

$$B_i = (B_{i+1} \cap A_i) \cup A_i',$$

với A_i' là tập có ít phần tử nhất trong $A_i \setminus B_{i+1}$ sao cho f_{i+1} áp dụng được trên tập hợp $(B_{i+1} \cap A_i) \cup A_i'$. Thật ra, A_i' có được xác định như sau:

$A_i' = u(f_{i+1}) \setminus B_{i+1}$ nếu f_{i+1} không đối xứng,

và nếu f_{i+1} đối xứng thì

A_i' = một tập con gồm $\max(0,t_i)$ phần tử của tập hợp $(M(f_{i+1})\setminus B_{i+1}) \cap A_i$

trong đó $t_i = card(M(f_{i+1})) - r(f_{i+1}) - card(M(f_{i+1}) \cap B_{i+1} \cap A_i)$.

Với cách xây dựng nầy ta có thể kiểm tra được rằng dãy $\{B_0, B_1, ..., B_{m-1}, B_m\}$ thỏa mãn các điều kiện ghi trong định lý.

<u>**Ghi chú**</u> :

(**1**) Từ định lý trên ta có quá trình suy diễn và xác định các biến để giải bài toán $A \rightarrow B$ như sau:

bước 1: tính các biến trong tập $B_1 \setminus B_0$ (áp dụng f_1).

bước 2: tính các biến trong tập $B_2 \setminus B_1$ (áp dụng f_2).

v.v...

bước m: tính các biến trong tập $B_m \setminus B_{m-1}$ (áp dụng f_m).

(**2**) Từ chứng minh của định lý trên, ta có thể ghi ra một thuật toán để xây dựng dãy các tập biến $\{B_1', ..., B_{m-1}', B_m'\}$ rời nhau cần lần lượt được xác định (hay được suy ra) trong quá trình giải bài toán ($B_i' = B_i \setminus B_{i-1}$) gồm các bước chính như sau:

- xác định các tập $A_0, A_1, ..., A_m$.
- xác định các tập $B_m, B_{m-1}, ..., B_1, B_0$.
- xác định các tập $B_1', B_2', ..., B_m'$.

Ví dụ 2.10: Giả sử ta có dãy $\{f_1, f_2, f_3\}$ là một lời giải tốt cho bài toán $A \rightarrow B$, trong đó :

$A = \{a_1, b_1, b_2\}$,

$B = \{b_3\}$,

f_1, f_2 là các quan hệ đối xứng có hạng 2, f_3 là quan hệ đối xứng có hạng 1, và

$M(f_1) = \{a_1, b_1, c_1, d_1\}$,

$M(f_2) = \{a_1, c_1, b_2, d_2, e_2\}$,

$M(f_3) = \{b_1, e_2, b_3\}$.

Dựa theo sự phân tích quá trình giải trong định lý trên ta có :

$A_0 = A$,

$A_1 = \{a_1, b_1, b_2, c_1, d_1\}$,

$A_2 = \{a_1, b_1, b_2, c_1, d_1, d_2, e_2\}$,

$A_3 = \{a_1, b_1, b_2, c_1, d_1, d_2, e_2, b_3\}$,

$B_3 = B$,

$B_2 = \{b_1, e_2\}$,

$B_1 = \{b_1, a_1, c_1, b_2\}$,

$B_0 = \{a_1, b_1, b_2\}$,

và các tập biến cần lần lượt tính toán trong bài giải cho bài toán là :

$B_1' = \{c_1\}$,

$B_2' = \{e_2\}$,

$B_3' = \{b_3\}$.

Từ đó ta có quá trình suy diễn theo lời giải trên như sau:

Xác định c_1 (áp dụng f_1),

Xác định e_2 (áp dụng f_2),

Xác định b_3 (áp dụng f_3).

2.4 MẠNG SUY DIỄN CÓ TRỌNG SỐ VÀ LỜI GIẢI TỐI ƯU

Vấn đề ta quan tâm trong phần nầy là việc tìm một lời giải tốt nhất (hay lời giải tối ưu). Quan niệm về lời giải tốt nhất phụ thuộc vào việc ta quan tâm đến mục tiêu tối ưu nào đối với lời giải. Trong định nghĩa 2.3, ta quan tâm đến những lời giải với số bước suy diễn ít nhất mà ta gọi là lời giải ngắn nhất. Do tính thứ tự tốt của tập hợp số tự nhiên ta có thể thấy rằng: nếu bài toán $A \rightarrow B$ là giải được thì sẽ tồn tại một lời giải ngắn nhất cho bài toán. Tuy nhiên, trong nhiều trường hợp áp dụng cụ thể như hệ suy diễn tính toán hay hệ suy diễn trên các phản ứng hóa học, mỗi luật suy diễn có một tham số tương ứng đại diện cho độ phức tạp của luật suy diễn hay chi phí. Những tham số nầy đóng vai trò quan trọng trong tính hiệu quả của lời giải. Ví dụ, nếu so sánh 2 công thức tính toán

(1) $a + b + c = 2*p$, và

(2) $a^2 = b^2 + c^2 - 2*b*c*\cos(A)$

để chọn công thức cho việc thực hiện tính toán a thì công thức (1) tốt hơn vì trong công thức (1) ta chỉ sử dụng các phép tính +, - và *; trong khi ở công thức (2) ta phải tính giá trị hàm lượng giác và tính căn bậc hai. Trong phần nầy ta sẽ xem xét các mạng suy diễn có trọng số, trong đó ứng với mỗi luật suy diễn có một trọng số dương tương ứng, và lời giải tối ưu sẽ được đề cập đến theo nghĩa là tổng trọng số thấp nhất. Lời giải ngắn nhất chính là lời giải tối ưu trong trường hợp trọng số của các luật dẫn đều là 1. Thuật giải A^* là cơ sở để tìm ra một lời giải tối ưu trong trường hợp bài toán là giải được.

2.4.1 Định nghĩa và ký hiệu

- **Định nghĩa 2.5**: Ta gọi một *mạng suy diễn có trọng số* là một mô hình (A, D, w) bao gồm:

    (1) một tập hợp các thuộc tính A,

(2) một tập hợp các luật suy diễn D, và

(3) một hàm trọng số dương $w : D \rightarrow \mathbf{R}^+$

Mỗi luật dẫn r thuộc D có dạng $r : u \Rightarrow v$, với u và v là các tập hợp con khác rỗng và rời nhau của A. Ta gọi u là phần giả thiết của luật r và ký hiệu là *hypothesis*(r). Tập v được gọi là phần kết luận của luật r và ký hiệu là *goal*(r). Tập hợp *attr*(r) = *hypothesis*(r) ∪ *goal*(r) được gọi là tập hợp các thuộc tính trong luật r. Mạng suy diễn có trọng số sẽ được viết tắt là **MSDT**.

Ghi chú:

- Tập hợp D các luật suy diễn có thể viết dưới dạng một tập hợp các quan hệ suy diễn mà mỗi quan hệ suy diễn xác định các luật suy diễn có trọng số bằng nhau.
- Trong một MSDT (*A*, *D*, w) ta có (*A*, *D*) là một MSD.

Ví dụ 2.11: Gọi A là tập hợp gồm 3 cạnh của một tam giác (a, b, và c), 3 góc trong (A, B, và C), nửa chu vi p, diện tích S, 3 đường cao (ha, hb, và hc), và các yếu tố khác của tam giác:

A = {A, B, C, a, b, c, p, S, ha, hb, hc, ...}.

Giữa các thuộc tính trên của một tam giác ta có các luật suy diễn được xác định bởi các quan hệ suy diễn thể hiện bởi các công thức liệt kê trong tập hợp sau đây:

D = {

f1: A + B + C = 180; f2: a/sin(A) = b/sin(B);

f3: b/sin(B) = c/sin(C); f4: a/sin(A) = c/sin(C);

f5: 2*p = a + b + c; f6: S = a*ha/2;

f7: S = b * hb / 2; f8: S = c*hc/2;

f9: S = sqrt(p*(p-a)*(p-b)*(p-c)); f10: ha = b*sin(C);

f11: hb = a*sin(C); f12: hc = b*sin(A)

}.

Giả sử các phép toán +, -, * và / được đặt cho trọng số là 1, phép tính căn bậc 2 có trọng số là một hằng số dương c1, và các tính toán hàm lượng giác có trọng số là một hằng số dương c2; trong đó c1 >> 1 và c2 >> 1 (c1 và c2 lớn hơn 1 nhiều). Như thế các quan hệ suy diễn có trọng số tương ứng như sau:

w(f1) = 2; w(f2) = w(f3) = w(f4) = 2*c2 + 2;

w(f5) = 3; w(f6) = w(f7) = w(f8) = 2;

w(f9) = c1 + 6; w(f10) = w(f11) = w(f12) = c2 + 1;

v.v . . .

Khi đó ta có (***A***, ***D***, w) là một mạng suy diễn có trọng số.

- **<u>Định nghĩa 2.6</u>**: Giả sử (*A*, *D*, w) là một MSDT. Cho $S = \{f_1,...,f_k\}$ là một dãy các luật suy diễn và A là một tập hợp các thuộc tính. Đặt

$S_1(A) = A \cup goal(f_1)$ nếu $hypothesis(f_1) \subseteq A$,

$= A$ nếu ngược lại.

$S_i(A) = S_{i-1}(A) \cup goal(f_i)$ nếu $hypothesis\ (f_i) \subseteq S_{i-1}(A)$,

$= S_{i-1}(A)$ nếu ngược lại.

(với mọi i = 2,...,k).

$S\ (A) = S_k(A)$,

w(*S*) = tổng của các w(f), với f chạy trên *S*

$= w(f_1) + w(f_2) + ... + w(f_k)$.

Ta gọi w(*S*) là trọng số của *S*.

Cho một bài toán A → B. Dãy các luật suy diễn *S* được gọi là một lời giải tối ưu của bài toán khi nó thỏa mãn các điều kiện sau đây:

(1) *S* là một lời giải của bài toán A → B.

(2) Trọng số của S nhỏ hơn hoặc bằng trọng số của bất kỳ một lời giải nào khác của bài toán. Nói một cách khác là:

$$w(S) = \min \{w(S') \mid S' \text{ là một lời giải của bài toán } A \rightarrow B \}$$

Nhận xét: Trong trường hợp hàm trọng số là hàm hằng thì một lời giải tối ưu chính là một lời giải với số bước suy diễn thấp nhất, tức là lời giải ngắn nhất. Cũng có thể thấy rằng lời giải tối ưu không nhất thiết là duy nhất. Nói một cách khác, bài toán có thể có nhiều lời giải tối ưu khác nhau. Tuy nhiên, chúng ta thường chỉ cần tìm ra một lời giải tối ưu mà thôi.

2.4.2 Lời giải và độ phức tạp của quá trình tìm lời giải

Xét bài toán $H \rightarrow G$ trên một MSDT (A, D, w), với H và G là các tập con của tập thuộc tính A. Đây cũng là một bài toán trên mạng suy diễn (A, D). Từ những kết quả được trình bày trong các mục trước ta có thể tóm tắt quá trình tìm một lời giải theo phương pháp suy diễn tiến qua thuật giải dưới đây:

Thuật toán 2.4

Bước 1: Tìm một lời giải.

Khi G chưa bao hàm trong H ta thực hiện quá trình lặp cho các bước dưới đây:

Bước 1.1: Tìm luật r $\in D$ có thể áp dụng được để suy ra các thuộc tính mới, tức là *hypothesis*(r) bao hàm trong H nhưng *goal*(r) không bào hàm trong H.

Bước 1.2: Nếu việc tìm kiếm ở bước 1.1 thất bại (không tìm được luật r như mong muốn) thì kết thúc với kết quả là: bài toán không có lời giải.

Bước 1.3: Ngược lại thì bổ sung thêm *goal*(r) vào H và ghi nhận r vào danh sách các luật đã được áp dụng.

Bước 2: Rút gọn lời giải.

Giả sử S ={r_1, ..., r_p} là một lời giải (nếu có) tìm được trong bước 1. Ta thực hiện các bước sau đây:

Bước 2.1: $G' \leftarrow G \setminus H$;

Bước 2.2: for k:=p downto 1 do

if ($goal(r_k) \notin G'$) then

Loại r_k ra khỏi danh sách S

else

$G' \leftarrow (G' \setminus goal(r_k)) \cup (hypothesis(r_k) \setminus H)$

<u>Mệnh đề 2.5</u> Thuật toán 2.4 cho lời giải là đúng và có độ phức tạp là O(|A|.|D|.min(|A|,|D|)).

Trong các cài đặt cụ thể ta có thể sử dụng thêm các qui tắc heuristics để tăng thêm hiệu quả cho việc tìm lời giải.

2.4.3 Tìm lời giải tối ưu

Trong mục nầy ta sẽ xem xét vấn đề tìm lời giải tối ưu cho bài toán $H \rightarrow G$ trên một MSDT (A, D, w). Cách tiếp cận để giải quyết vấn đề nầy là dựa trên thuật giải A^*. Để có thể áp dụng thuật giải nầy chúng ta cần có một biểu diễn thích hợp cho không gian trạng thái của bài toán cũng như cho yêu cầu của bài toán.

- <u>Không gian trạng thái của bài toán</u>: Xét bài toán $H \rightarrow G$ trên một MSDT (A, D, w). Với mỗi luật r mà ta có thể áp dụng trên H để suy ra những thuộc tính mới (nghĩa là *hypothesis*(r) bao hàm trong H nhưng *goal*(r) thì không bao hàm trong H) sẽ dẫn tới một tập thuộc tính mới $H' = H \cup goal(r)$. Ta nói rằng r là một *cạnh* nối từ *đỉnh H* đến *đỉnh H'*. Như thế, tập hợp gồm tất cả các tập con (hay đỉnh) H' của A sao cho có một dãy S gồm các luật (hay cạnh) thỏa mãn điều kiện $H' = S(H)$, cùng với các cạnh trong các dãy S sẽ cho ta một không gian trạng thái của bài toán có dạng đồ thị. Hơn nữa đồ thị nầy có trọng số được định nghĩa bởi: trọng số của cạnh r (tức là một luật suy diễn) là w(r). Đồ thị có trọng số nầy sẽ được ký hiệu là Graph($H \rightarrow G$). Từ cách xây dựng đồ thị của bài toán ta có mệnh đề sau đây:

Mệnh đề 2.6:

(1) Một dãy S gồm các luật là một lời giải của bài toán $H \rightarrow G$ khi và chỉ khi S là một lộ trình trên đồ thị Graph($H \rightarrow G$) nối từ H đến $S(H)$ và $S(H) \supset G$.

(2) Độ dài của một lộ trình S trên đồ thị Graph($H \rightarrow G$) là w(S), trọng số của danh sách luật S trên MSDT (A, D, w).

Từ mệnh đề nầy, việc tìm lời giải tối ưu cho bài toán $H \rightarrow G$ tương đương với việc tìm một đường đi ngắn nhất trên đồ thị Graph($H \rightarrow G$) từ H đến một đỉnh mục tiêu H' thỏa điều kiện H' chứa G. Đối với mỗi đỉnh N trên đồ thị, đặt

$$h(N) = \min \{w(r) \mid hypothesis(r) \subset N\}$$

tức là h(N) là trọng số nhỏ nhất của các luật áp dụng được trên N. Giá trị h(N) nầy có thể xem là một ước lượng cho lộ trình từ N đến một đỉnh mục tiêu. Từ đó chúng ta có thể viết thuật giải tìm lời giải tối ưu cho bài toán như sau:

Thuật toán 2.5

Bước 1: Khởi tạo trạng thái xuất phát.

Open ← {H}; // danh sách đỉnh mở ban đầu chỉ có đỉnh xuất phát

Close ← {}; // danh sách đỉnh đóng

g(H) ← 0; // độ dài lộ trình đến H là 0

f(H) ← h(H); // độ dài lộ trình ước tính từ H đến mục tiêu là h(H)

found ← false; // biến kiểm tra quá trình tìm lời giải

Bước 2: Thực hiện quá trình lặp để tìm lời giải tối ưu.

While (Open ≠ {}) do

Begin

Bước 2.1: Chọn một đỉnh N trong Open với ước tính đường đi f nhỏ nhất.

Bước 2.2: Chuyển N từ danh sách Open sang danh sách Close.

Bước 2.3: if (N là một mục tiêu) then

Begin

Found ← true; Break; // Kết thúc quá trình lặp

End

Bước 2.4: else // N không là một mục tiêu

Begin

Duyệt qua các đỉnh kế S của N (tức là có cung r nối N và S) mà S $\notin$ Close, ứng với mỗi S ta xét các trường hợp sau:

1. S $\notin$ Open: Tính

   $g(S) = g(N) + w(r)$; $f(S) = g(S) + h(S)$;

   Bổ sung S vao Open;

2. S $\in$ Open:

   If $g(N) + w(r) < g(S)$ then

   Begin

   $g(S) \leftarrow g(N) + w(r)$; $f(S) \leftarrow g(S) + h(S)$;

   Cập nhật thông tin về đỉnh kế trước của S trên lộ trình;

   end

End

End // Kết thúc vòng lặp while

Bước 3: Kiểm tra kết quả việc tìm kiếm.

If Found then Kết quả là tìm được lời giải tối ưu và thiết lập lời giải

Else Kết quả là bài toán không có lời giải.

<u>Mệnh đề 2.7</u> Thuật toán 2.5 cho lời giải là đúng và có độ phức tạp là $O(|A|^2.|D|^2)$.

<u>Ví dụ 2.12</u>: Giả sử ta có mạng suy diễn có trọng số (*A*, *D*, w) như sau:

A = {A, B, C, a, b, c, p, S, ha, hb, hc}.

D = { f1: A + B + C = 180; f2: a/sin(A) = b/sin(B);
f3: b/sin(B) = c/sin(C); f4: a/sin(A) = c/sin(C);
f5: 2*p = a + b + c; f6: S = a*ha/2;
f7: S = b * hb / 2; f8: S = c*hc/2;
f9: S = sqrt(p*(p-a)*(p-b)*(p-c)); f10: ha = b*sin(C);
f11: hb = a*sin(C); f12: hc = b*sin(A).
}.

w(f1) = 2; w(f2) = w(f3) = w(f4) = 2*c2 + 2;

w(f5) = 3; w(f6) = w(f7) = w(f8) = 2;

w(f9) = c1 + 6; w(f10) = w(f11) = w(f12) = c2 + 1.

Trong đó c1 và c2 là các hằng số dương với c1 >> 1 và c2 >> 1.

Xét bài toán $H \rightarrow G$ với *H* = {a, b, B} và *G* = {S}. Trên mạng suy diễn (*A*, *D*), nếu áp dụng thuật toán 2.4 ta có thể tìm được một lời giải *S* = {f2, f1, f3, f5, f9}. Trên mạng suy diễn có trọng số (*A*, *D*, w) lời giải *S* có trọng số là w(*S*) = 4*c2 + c1 + 15. Áp dụng thuật toán 2.5 để tìm lời giải trên mạng (*A*, *D*, w) ta có thể tìm được một lời giải tối ưu *S'* = {f2, f1, f10, f6} với trọng số là w(*S'*) = 3*c2 + 7.

2.5 TẬP HỢP SINH VÀ VIỆC KIỂM ĐỊNH, BỔ SUNG GIẢ THIẾT

Trong mục nầy sẽ xem xét về sự thừa hay thiếu đối với giả thiết của bài toán $H \rightarrow G$ trên một mạng suy diễn (*A*, *D*) và trong trường hợp cần thiết thì tìm cách điều chỉnh giả thiết *H*. Trước hết ta cần xét xem bài toán có giải được hay không. Trường hợp bài toán giải được thì giả thiết là đủ. Tuy nhiên có thể xảy ra tình trạng thừa giả thiết. Để biết được bài toán có thật sự thừa giả thiết hay không, ta có thể dựa vào thuật toán tìm một sự thu gọn giả thiết sau đây:

Thuật toán: Tìm một sự thu gọn giả thiết của bài toán.

Nhập : Mạng suy diễn (A, D),

Bài toán $H \rightarrow G$ giải được,

Xuất : tập giả thiết mới $H' \subseteq H$ tối tiểu theo thứ tự $\subseteq$.

Thuật toán :

Repeat

$H' \leftarrow H$;

for $x \in H$ **do**

if $H - \{x\} \rightarrow G$ giải được **then** $H \leftarrow H - \{x\}$;

Until $H = H'$;

Trong thuật toán trên nếu tập giả thiết mới H' thật sự bao hàm trong H thì bài toán bị thừa giả thiết và ta có thể bớt ra từ giả thiết H tập hợp các biến không thuộc H', coi như là giả thiết cho thừa.

Trường hợp bài toán $H \rightarrow G$ là không giải được thì ta nói giả thiết H thiếu. Khi đó có thể điều chỉnh bài toán bằng nhiều cách khác nhau để cho bài toán là giải được. Chẳng hạn ta có thể sử dụng một số phương án sau đây:

phương án 1 : Tìm một $A' \subseteq M \setminus (\overline{A} \cup B)$ tối tiểu sao cho bao đóng của tập hợp $A' \cup A$ chứa B.

phương án 2 : Khi phương án 1 không thể thực hiện được thì ta không thể chỉ điều chỉnh giả thiết để cho bài toán là giải được. Trong tình huống nầy, ta phải bỏ bớt kết luận hoặc chuyển bớt một phần kết luận sang giả thiết để xem xét lại bài toán theo phương án 1.

Khái niệm tập hợp sinh và phương pháp tìm một tập hợp sinh trên một mạng suy diễn là cơ sở cho việc phát triển các thuật toán giải quyết vấn đề kiểm định giả thiết của bài toán suy diễn. Hơn nữa việc khảo sát về tập hợp sinh cũng là một vấn đề được đặt ra một cách tự nhiên trên mạng suy diễn nhằm tìm ra một tập hợp tối thiểu các thuộc tính và các luật suy diễn sinh ra tất cả các thuộc tính khác.

2.5.1 Khái niệm tập hợp sinh

- **<u>Định nghĩa 2.7</u>**: Cho (*A*, *D*) là một mạng suy diễn. Một tập thuộc tính $S \subset A$ được gọi là một *tập hợp sinh* của mạng suy diễn khi ta có bao đóng của S trên mạng là *A*, nghĩa là $\overline{S} = A$.

<u>Ví dụ 2.13</u>: Xét mạng suy diễn (*A*, *D*) với *A* = {a, b, c, A, B, C, R, p, S} gồm các biến số thực dương và *D* là tập hợp các luật suy diễn tương ứng của các quan hệ suy diễn thể hiện bởi các công thức sau đây:

f1: $A + B + C = \pi$ f2: a/sin(A) = 2*R

f3: b/sin(B) = 2*R f4: c/sin(C) = 2*R

f5: a + b + c = 2*p f6: $a^2 = b^2 + c^2 - 2*b*c*\cos(A)$

f7: $b^2 = a^2 + c^2 - 2*a*c*\cos(B)$ f8: $c^2 = b^2 + a^2 - 2*b*a*\cos(C)$

f9: S = sqrt(p*(p-a)*(p-b)*(p-c))

Mạng suy diễn nầy có một tập hợp sinh là *T* = {a, A, B} và từ *T* ta suy ra được *D* nhờ các luật suy diễn sau:

$A, a \Rightarrow R$; $A, B \Rightarrow C$; $R, C \Rightarrow c$; $R, B \Rightarrow b$; $a, b, c \Rightarrow p$; $a, b, c, p \Rightarrow S$

Từ định nghĩa về tập hợp sinh ở trên ta có các tính chất sau:

(1) Nếu S là một tập hợp sinh trong một mạng suy diễn và $S \subset T$ thì T cũng là một tập hợp sinh.

(2) Nếu S là một tập hợp sinh trong mạng suy diễn (A, D) và D' là một tập hợp mở rộng của D, nghĩa là D ⊂ D', thì ta cũng có S là một tập hợp sinh của mạng suy diễn (A, D').

Hiển nhiên là mỗi mạng suy diễn đều có tập hợp sinh. Vấn đề mà chúng ta sẽ khảo sát là tìm một tập hợp sinh.

2.5.2 Tìm tập hợp sinh

Trong mục nầy sẽ trình bày cách tìm một tập hợp sinh không tầm thường với số phần tử càng ít càng tốt của mạng suy diễn. Trước hết, ta có thể tìm một tập hợp sinh bằng phương pháp thử dần được thể hiện bởi thuật toán sau đây:

Thuật toán 2.6: Tìm một tập hợp sinh S trong mạng suy diễn (*A*, *D*).

Bước 1: S ← {} // Đặt cho S ban đầu là rỗng

Bước 2: Tính bao đóng *Closure*(S) của tập hợp S.

Bước 3: Kiểm tra so sánh *Closure*(S) và *A*.

If *Closure*(S) = *A* then Kết thúc

Else

Begin

Chọn một phần tử x ∈ A-S; S ← S ∪ {x};

Quay lại Bước 2;

End

Thuật toán nầy sẽ cho ta một tập hợp sinh với độ phức tạp $O(|A|^2.|D|.\min(|A|,|D|))$, do độ phức tạp của phép toán tập hợp trên các tập hợp thuộc tính của *A* là $O(|A|)$ và độ phức tạp của thuật toán tìm bao đóng của một tập hợp trên mạng suy diễn (*A*, *D*) là $O(|A|.|D|.\min(|A|,|D|))$.

Ta có thể xây dựng một thuật toán tốt hơn thuật toán trên bằng cách thiết lập một "biểu đồ (hay đồ thị) phân cấp" của một mạng suy diễn. Không làm mất tính tổng quát ta có thể giả sử các luật suy diễn có phần kết luận gồm 1 phần tử.

- **Định nghĩa 2.8**: Cho một mạng suy diễn (A, D) mà mỗi luật suy diễn có phần kết luận gồm 1 phần tử. Ta xây dựng một đồ thị định hướng Graph(A, D) như sau:

  (1) Tập hợp đỉnh gồm tất cả các thuộc tính và tất cả các luật suy diễn, tức là $A \cup D$.

  (2) Ứng với mỗi luật r : *hypothesis*(r) $\rightarrow$ *goal*(r) ta thiết lập một tập hợp *edges*(r) gồm tất cả các cung định hướng (x, r) và (r, y) thỏa x $\in$ *hypothesis*(r) và y $\in$ *goal*(r) một cách tương ứng. Tập hợp các cạnh của đồ thị Graph(A, D) là hợp của tất cả các tập hợp *edges*(r) với r chạy trong tập D.

  Trong trường hợp trên đồ thị Graph(A, D) ta có $\deg_{in}(x) \leq 1$ với mọi x $\in A$, thì ta có thể hiểu ngầm các đỉnh thuộc D và xét một đồ thị thu gọn $\text{Graph}_D(A)$ gồm:

  (1) Tập hợp đỉnh là tập hợp các thuộc tính A.

  (2) Tập hợp cạnh gồm tất cả các cung (x,y) thỏa mãn điều kiện: Có (duy nhất) một luật r sao cho *goal*(r) = y và x $\in$ *hypotheis*(r).

<u>Ví dụ 2.14</u>: Mạng suy diễn (A, D) với A = {a, b, c, A, B, C, R, p, S} và D là tập các luật suy diễn sau:

r1: A, a $\Rightarrow$ R; r2: A, B $\Rightarrow$ C; r3: R, C $\Rightarrow$ c;

r4: R, B $\Rightarrow$ b; r5: a, b, c $\Rightarrow$ p; r6: a, b, c, p $\Rightarrow$ S

sẽ có đồ thị Graph(A, D) có tập hợp đỉnh là

$A \cup D$ = {a, b, c, A, B, C, R, p, S, r1, r2, r3, r4, r5, r6}

và tập hợp các cung là

{(A,r1), (a,r1), (r1,R), (A,r2), (B,r2), (r2,C),
(R,r3), (C,r3), (r3,c), (R,r4), (B,r4), (r4,b),
(a,r5), (b,r5), (c,r5), (r5,p), (a,r6), (b,r6), (c,r6), (p,r6), (r6,S) }

Trên đồ thị Graph(A, D) mỗi x ∈ A có đúng một cung hướng tới, và đồ thị thu gọn $\text{Graph}_D(A)$ có tập đỉnh là

A = {a, b, c, A, B, C, R, p, S }

và tập hợp cạnh là

{(A,R), (a,R), (A,C), (B,C), (R,c), (C,c), (R,b), (B,b),
(a,p), (b,p), (c,p), (a,S), (b,S), (c,S), (p,S)}

- **Định nghĩa 2.9**: Ta gọi một đồ thị định hướng đơn giản không có chu trình là một *biểu đồ phân cấp* (hay một đồ thị phân cấp). Đỉnh x của đồ thị mà không có cung hướng tới sẽ được gọi là đỉnh mức 0, và ta viết *level*(x) = 0. Đối với các đỉnh x khác, ta định nghĩa một cách qui nạp mức của nó và mức của đỉnh nầy có thể được cho bởi

$$level(x) = \max \{level(a) \mid \text{có cung nối a với x}\} + 1$$

Như thế tập hợp đỉnh của biểu đồ phân cấp được sắp xếp thành các mức, với mức 0 của biểu đồ là tập hợp tất cả các đỉnh mức 0, mức 1 của biểu đồ là tập hợp tất cả các đỉnh mức 1, v.v…

Ví dụ 2.15: Đồ thị $\text{Graph}_D(A)$ trong ví dụ 2.14, được vẽ trong hình 2.1, là một biểu đồ phân cấp với 5 mức là:

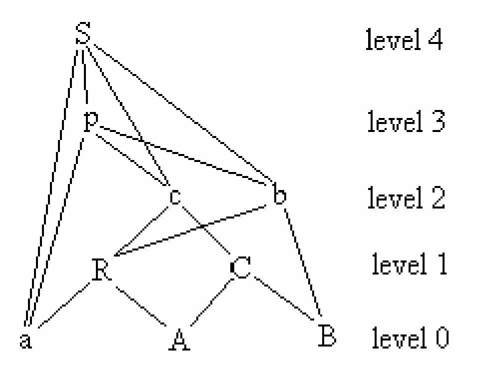
Level_0 = {a, A, B}, nghĩa là *level*(a) = *level*(A) = *level*(B) = 0.

Level_1 = {R, C}, nghĩa là *level*(R) = *level*(C) = 1.

Level_2 = {c, b}, nghĩa là *level*(c) = *level*(b) = 2.

Level_3 = {p}, nghĩa là *level*(p) = 3.

Level_4 = {S}, nghĩa là *level*(S) = 4.



Hình 2.1 Một biểu đồ phân cấp gồm 5 mức.

Từ các định nghĩa trên ta có thể dễ dàng chứng minh các mệnh đề sau đây:

Mệnh đề 2.8: Cho mạng suy diễn (*A*, *D*). Giả sử đồ thị Graph(*A*, *D*) có đồ thị thu gọn $\text{Graph}_D(A)$. Khi ấy, nếu $\text{Graph}_D(A)$ là một đồ thị phân cấp thì tập hợp S = Level_0 gồm tất cả các đỉnh mức 0 sẽ cho ta một tập hợp sinh của mạng suy diễn. Hơn nữa trong trường hợp nầy ta còn có:

(1) S là tập hợp sinh nhỏ nhất trên mạng suy diễn.

(2) Tập *D* là tập hợp luật tối thiểu để Level_0 sinh ra *A*. Nói một cách khác, nếu *D'* là một tập hợp con thật sự của *D* thì S không phải là tập hợp sinh trên mạng suy diễn (*A*, *D'*).

Định lý 2.5: Cho mạng suy diễn (*A*, *D*). ta có:

(1) S $\subset$ *A* là một tập hợp sinh trên mạng suy diễn khi và chỉ khi có một tập luật *D'* $\subset$ *D* sao cho Graph(*A*, *D'*) là một đồ thị phân cấp và S chứa tập hợp các đỉnh mức 0 của đồ thị nầy.

(2) Tồn tại một tập luật *D'* $\subset$ *D* sao cho Graph(*A*, *D'*) là một đồ thị phân cấp.

Từ định lý trên ta có thể tìm một tập hợp sinh trên mạng suy diễn (*A*, *D*) bằng cách xây dựng một thuật toán tìm một tập luật *D'* sao cho Graph(*A*, *D'*) là một đồ thị phân cấp, và lấy các đỉnh ở mức 0 của đồ thị nầy. Cách làm nầy sẽ được áp dụng trong việc tìm một sự bổ sung giả thiết cho bài toán suy diễn một MSD trong trường hợp bài toán thiếu giả thiết. Chúng ta cũng có một thuật toán như sau:

<u>Thuật toán 2.7</u>: Tìm một tập hợp sinh S trong mạng suy diễn (A, D) bằng cách xây dựng một mạng con (A', D') với $A' = A$ và có Graph(A', D') là một biểu đồ phân cấp.

Bước 1: $A' \leftarrow \{\}$; $D' \leftarrow \{\}$; S $\leftarrow$ {};

Bước 2: For r $\in D$ do

If not ($attr$(r) $\subset A'$) then

Thực hiện việc cập nhật A', D' và S theo 2 trường hợp như sau:

- Trường hợp 1: goal(r) $\notin A'$

  S $\leftarrow$ S $\cup$ ($hypothesis$(r) - A');

- Trường hợp 2: goal(r) $\in A'$

  Loại r' (nếu có) trong D' mà $goal$(r') = $goal$(r);

  Loại một số luật suy ra các x $\in$ $hypothesis$(r) để bảo đảm tính phân cấp của biểu đồ và cập nhật S.

$A' \leftarrow A' \cup attr$(r); $D' \leftarrow D' \cup$ {r};

Bước 3: S $\leftarrow$ S $\cup$ (A - A'); $A' \leftarrow A$

Ghi chú:

- Nhằm hướng tới việc xác định một tập hợp sinh với số phần tử nhỏ và với độ phức tạp thấp ta có thể sử dụng thêm các qui tắc heuristic bổ sung thêm cho quá trình xây dựng một mạng suy diễn con mà đồ thị của nó có dạng một biểu đồ phân cấp.
- Từ phương pháp tìm một tập hợp sinh trong một mạng suy diễn đã được trình bày ở trên ta có thể phát triển một thuật toán tương tự để tìm một tập hợp sinh chứa một tập hợp thuộc tính cho trước.

2.5.3 Bổ sung giả thiết cho bài toán suy diễn

Trong mục nầy ta sẽ xem xét việc bổ sung giả thiết cho bài toán $H \rightarrow G$ trên một mạng suy diễn (A, D) trong trường hợp bài toán không giải được. Ý tưởng chính ở đây là tiến hành một quá trình xây dựng một biểu đồ phân cấp với tập hợp đỉnh chứa G và ưu tiên cho việc đặt các phần tử của H ở mức 0. Thuật toán cơ bản dưới đây cho ta một cách để tìm một tập thuộc tính H' sao cho $H \cap H' = \varnothing$ và bài toán $(H \cup H') \rightarrow G$ là giải được trên mạng suy diễn.

<u>Thuật toán 2.8</u>: Cho mạng suy diễn (A, D) và bài toán $H \rightarrow G$ không giải được (không có lời giải). Tìm H' sao cho $H \cap H' = \varnothing$ và bài toán $(H \cup H') \rightarrow G$ là giải được.

Bước 1: $A' \leftarrow H$; $D' \leftarrow \{\}$; $G \leftarrow G \setminus A'$;

Bước 2: while ($G \neq \{\}$ and $D \neq \{\}$) do

2.1: Lấy ra một r từ D và cập nhật D.

2.2: if *hypothesis*(r) $\cap\, G \neq \{\}$ or *attr*(r) $\subseteq A'$ then Bỏ qua r

2.3: else Thêm r vào D' và bổ sung *attr*(r) vào A'và trong trường hợp goal(r) $\in A'$ thì loại ra r' từ D' (nếu có) thỏa *goal*(r') = *goal*(r) và loại một số luật suy ra các x $\in$ *hypothesis*(r) để bảo đảm tính phân cấp của biểu đồ.

2.4: $G \leftarrow G - A'$

Bước 3: if $G \neq \{\}$ then Kết thúc với kết luận:Vấn đề bổ sung giả thiết là không giải quyết được

Bước 4: else (Trong trường hợp nầy Graph(A',D') có biểu đồ phân cấp tương ứng là $\text{Graph}_{D'}A'$) Gọi L_0 là mức 0 của biểu đồ $\text{Graph}_{D'}A'$. Đặt: $H' \leftarrow L_0 - H$.

<u>Mệnh đề 2.9</u>: Thuật toán 2.8 để tìm sự bổ sung giả thiết cho bài suy diễn là đúng và có độ phức tạp là O(|*A*|.|*D*|).

Ghi chú: Ta có thể sử dụng thêm các qui tắc heuristic trong bước 2 của thuật toán trên nhằm giảm thiểu số thuộc tính ở mức 0 và/hay để ưu tiên chọn lựa các thuộc tính được ưu tiên cho việc dùng bổ sung giả thiết của bài toán.

2.6 MẠNG SUY DIỄN-TÍNH TOÁN

Trong phần nầy chúng ta xét đến một mô hình mạng suy diễn liên qua đến tính toán, được gọi là mạng suy diễn – tính toán, mà trong nhiều ứng dụng nó cho ta một sự biểu diễn đầy đủ và thích hợp hơn của tri thức cần thiết liên quan đến các vấn đề hay các bài toán cần giải quyết. Cách biểu diễn như vậy sẽ thể hiện một cách tự nhiên cấu trúc của tri thức, và giúp ta thiết kế các thuật giải toán tự động một cách dễ dàng với những lời giải tường minh phù hợp với cách nghĩ và làm của con người trong các ứng dụng giải toán, đặc biệt là các hệ thống hỗ trợ cho việc học tập và giảng dạy. Những phần mềm như thế cần có cách biểu diễn tri thức tự nhiên và thích hợp cho việc thiết kế các môđun giải toán cho lời giải phù hợp với học sinh cũng như các môđun giải thích, môđun kiểm tra và luyện tập giải toán.

2.6.1 Mô hình

- **<u>Định nghĩa 2.10</u>**: Một mạng suy diễn-tính toán gồm bốn thành phần:

  (1) Tập hợp *A* gồm các thuộc tính.

  (2) Tập hợp *D* gồm các luật suy diễn (hay các quan hệ suy diễn) trên các thuộc tính.

  (3) Tập hợp *F* gồm các công thức tính toán hay các thủ tục tính toán tương ứng với các luật suy diễn. Sự tương ứng nầy thể hiện bởi một ánh xạ

  $$f : D \to F$$

(4) Tập hợp *R* gồm một số qui tắc hay điều kiện ràng buộc trên các thuộc tính.

Mạng suy diễn tính toán gồm 4 tập hợp *A*, *D*, *F* và *R* như thế sẽ được ký hiệu bởi bộ bốn (*A*, *D*, *F*, *R*). Theo định nghĩa, ta có (*A*, *D*) là một mạng suy diễn và lời giải cho bài toán $H \rightarrow G$ trên mạng suy diễn nầy sẽ xác định các công thức hay các thủ tục tính toán các phần tử thuộc *G* từ các phần tử thuộc *H*.

Ví dụ 2.16: Kiến thức về một tam giác có thể được biểu diễn bởi một mạng suy diễn tính toán (*A*, *D*, *F*, *R*) như sau:

A = {A, B, C, a, b, c, R, S, p, ...} là tập hợp các yếu tố của một tam giác gồm 3 góc, 3 cạnh, bán kính vòng tròn ngoại tiếp, diện tích, nửa chu vi, v.v…

D = {r1: A, B $\Rightarrow$ C; r2: A, C $\Rightarrow$ B; r3: B, C $\Rightarrow$ A;

r4: A, a $\Rightarrow$ R; r5: A, R $\Rightarrow$ a; r6: R, a $\Rightarrow$ A; r7: A, b, c $\Rightarrow$ a; ...}

F = {f1: C = π-A-B; f2: B = π-A-C; f3: A = π-B-C;

f4: R = a/(2.sin(A)); f5: a = 2.R.sin(A); f6: A = arcsin(a/(2.R));

f7: a= b^2+c^2-2.b.c.cos(A); ...}

R = {a+b > c; a+c > b; b+c > a; a > b $\Leftrightarrow$ A > B; a = b $\Leftrightarrow$ A = B; ...}

2.6.2 Giải bài toán trên mạng suy diễn-tính toán

Trên một mạng suy diễn-tính toán ta có thể giải quyết các bài toán suy diễn tính toán chẳng hạn bài toán giải tam giác hay bài toán giải tứ giác dựa trên việc giải bài toán suy diễn trên mạng suy diễn. Hơn nữa, các công thức hay thủ tục tính toán có thể được gán cho các trọng số thể hiện độ phức tạp tính toán của chúng. Từ đó ta có thể tìm ra được những lời giải cho bài toán suy diễn tính toán với chi phí tính toán thấp nhất dựa trên việc tìm lời giải tối ưu trên mạng suy diễn có trọng số. Ngoài ra, chúng ta có thể tìm ra được các công thức tường minh qua các bước giải bài toán và rút gọn các công thức dưới dạng ký hiệu.

Như thế trên mạng suy diễn-tính toán một tam giác ta có thể chỉ ra một cách tự động các công thức tường minh để tính một số yếu tố nầy từ một số yếu tố khác (nếu bài toán có lời giải). Kết hợp điều nầy với việc dò tìm những sự liên hệ suy diễn giữa các yếu tố nào đó mà ta quan tâm sẽ cho ta một phương pháp để tự động tìm ra thêm những luật suy diễn và những công thức tính toán liên quan đến các yếu tố. Điều nầy có ý nghĩa như một kỹ thuật khám phá tri thức.

Một sự mở rộng của mạng suy diễn tính toán là cho phép xét thêm các quan hệ khác với các quan hệ suy diễn – tính toán ở đây, chẳng hạn các quan hệ hình học giữa các đối tượng hình học điểm, đoạn, tia, góc. Sự mở rộng nầy sẽ được tích hợp trong một cấu trúc trừu tượng theo phương pháp lập trình hướng đối tượng mà ta gọi là một đối tượng tính toán. Khái niệm về đối tượng tính toán sẽ được xây dựng trong chương tiếp theo và nó được xem xét trong một mô hình tri thức được gọi là mô hình tri thức các đối tượng tính toán.

CHƯƠNG 3

MÔ HÌNH TRI THỨC CÁC ĐỐI TƯỢNG TÍNH TOÁN

3.1 KHÁI NIỆM VỀ ĐỐI TƯỢNG TÍNH TOÁN VÀ MÔ HÌNH

3.1.1 Đối tượng tính toán (C-object)

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. Những quan hệ nầy giúp ta thực hiện sự suy diễn, tính toán và giải một số bài toán suy diễn-tính toán trên các thuộc tính của đối tượng. Ví dụ: trong giải toán hình học, một tam giác với các thuộc tính như 3 cạnh, 3 góc trong, diện tích, nửa chu vi, bán kính vòng tròn ngoại tiếp, v.v … cùng với các công thức liên hệ giữa các thuộc tính đó sẽ cho ta một cấu trúc của một đối tượng như thế. Theo cách tiếp cận hướng đối tượng (xem [25], [32] và [35]) trong biểu diễn tri thức và giải toán, chúng ta tích hợp vào cấu trúc đối tượng trên một số hành vi giải toán nhất định để tạo ra một đối tượng. Dựa trên các đối tượng nầy, nhiều bài toán khác nhau có thể được biểu diễn dưới dạng mạng các đối tượng. Cách biểu diễn nầy 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. So với các phương pháp được trình bày trong các tài liệu [19], [42], [46] và [56] cũng như trong [49] và [50], cách mô hình nầy tỏ ra có nhiều ưu điểm, đặc biệt là khả năng biểu diễn hầu như toàn bộ tri thức và các dạng bài toán tổng quát thuận tiện cho việc phát triển các thuật toán giải tự động và cung cấp những lời giải tự nhiên và phù hợp với cách nghĩ và viết của con người. Ngoài ra, nó còn giúp ích cho việc thiết kế và cài đặt phần cơ sở tri thức cũng như ngôn ngữ qui ước để đặc tả bài toán. Các kết quả nghiên cứu liên quan đến khái niệm về đối tượng tính toán và mô hình tri thức các đối tượng tính toán cùng với một số áp dụng của các mô hình được trình bày trong các bài báo [68], [58], [59], [64] và [65].

- **Định nghĩa 3.1**: Ta gọi một đối tượng tính toán (C-object) là một đối tượng O có cấu trúc bao gồm

  (1) Một danh sách các thuộc tính Attr(O) = $\{x_1, x_2, \ldots, x_n\}$ 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.

  (2) 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 $\subset$ Attr(O), tức là đối tượng O có khả năng cho ta biết tập thuộc tính lớn nhất có thể được suy ra từ A trong đối tượng O.
  - Xác định tính giải được của bài toán suy diễn tính toán có dạng A $\rightarrow$ B với A $\subset$ Attr(O) và B $\subset$ Attr(O). Nói một cách khác, đối tượng có khả năng trả lời câu hỏi rằng có thể suy ra được các thuộc tính trong B từ các thuộc tính trong A không.
  - Thực hiện các tính toán
  - Thực hiện việc gợi ý bổ sung giả thiết cho bài toán
  - Xem xét tính xác định của đối tượng, hay của một sự kiện

Ví dụ 3.1: Một cấu trúc tam giác với cấu trúc gồm các yếu tố như : 3 cạnh a, b, c; 3 góc tương ứng với 3 cạnh : A, B, C; 3 đường cao tương ứng : ha, hb, hc; diện tích S của tam giác; nửa chu vi p của tam giác; bán kính đường tròn ngoại tiếp R của tam giác, v.v … cùng với các công thức liên hệ giữa chúng như định lý góc trong tam giác, định lý sin, định lý cosin, các công thức tính diện tích, … sẽ trở thành một đối tượng C-object khi ta tích hợp cấu trúc nầy 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. Như vậy ta có một đối tượng tam giác. Khi đối tượng tam giác nầy được yêu cầu cho một lời giải cho bài toán $\{a,B,C\} \Rightarrow S$ nó sẽ cung cấp một lời giải gồm 3 bước sau đây:

Bước 1: Xác định A bởi công thức $A = \pi - B - C$;

Bước 2: Xác định b bởi công thức $b = a.\sin(B)/\sin(A)$;

Bước 3: Xác định S bởi công thức $S = a.b.\sin(C)/2$;

Nếu yêu cầu là giải bài toán $\{a,B\} \Rightarrow S$ thì đối tượng sẽ trả lời rằng "không giải được" và nó có thể đề nghị cung cấp thêm thông tin như A, C, b hay c.

Ví dụ 3.2: Một cấu trúc tứ giác với cấu trúc gồm các yếu tố như : 4 cạnh a, b, c, d; 2 đường chéo; 4 góc, v.v … cùng với các công thức liên hệ giữa chúng và các sự kiện về các quan hệ sẽ trở thành một đối tượng C-object khi ta tích hợp cấu trúc nầy với các hành vi xử lý liên quan đến việc giải bài toán tứ 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. Như vậy ta có một đối tượng tứ giác.

3.1.2 Mô hình cho một C-object

Một C-Object có thể được mô hình hóa bởi một bộ:

(Attrs, F, Facts, Rules)

trong đó: 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 3.3: Đối tượng (C-Object) thuộc loại "TAM_GIAC" được biểu diễn theo mô hình trên gồm có:

- Attrs = { GocA, GocB, GocC, a, b, c, ha, hb, hc, ma, mb, mc, pa, pb, pc, S, p, R, r, ra, rb, rc }
- F = { GocA + GocB + GocC = Pi, a*sin(GocB) = b*sin(GocA),
  a^2 = b^2 + c^2 - 2*b*c*cos(GocA), . . . }
- Facts = {}
- Rules = { {GocA = GocB}⇒ {a = b},
  {a = b} ⇒ {GocA = GocB},
  {a^2 = b^2+c^2}⇒{GocA=pi/2},
  {GocA=pi/2} ⇒ {a^2 = b^2+c^2, b ⊥ c}, ...}

Ví dụ 3.4: Đối tượng (C-Object) thuộc loại "TU_GIAC" được biểu diễn theo mô hình trên gồm có:

- Attrs = { a, b, c, d, c1, c2, GA, GB, GC, GD, . . .}
- F = { GA + GB + GC + GD = 2*Pi, a+b+c+d = p,
  2*S = a*d*sin(GA)+ b*c*sin(GC),
  2*S = a*b*sin(GB)+ c*d*sin(GD), ...}
- Facts = {}
- Rules = { {a // c} ⇒ {GD=Pi-GA, GB=Pi-GC,
  GOC[A,B,D]=GOC[C,D,B], GOC[C,A,B]=GOC[A,C,D]},
  {GOC[C,A,B]=GOC[A,C,D]} ⇒ {a // c},
  {a=c, b=d} ⇒ {a // c, b // d}, ...}

Khảo sát các bài toán suy diễn và tính toán trên một C-Object và xây dựng các thuật giải thích hợp sẽ là cơ sở cho việc thiết kế các hệ hỗ trợ giải toán kết hợp với sự tra cứu kiến thức và học kiến thức. Tuy nhiên, mỗi loại C-Object 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 và các loại đối tượng khác nhau với những mối quan hệ hữu cơ. Ví dụ như 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 là một "đoạn thẳng". "đoạn thẳng" là một loại đối tượng có một thuộc tính giá trị thực chính là độ dài của nó và giá trị nầy sẽ được dùng trong các công thức tính toán hay các quan hệ tính toán; mặt khác đoạn thẳng cũng có các quan hệ phi tính toán như quan hệ song song hay quan hệ vuông góc. Các loại quan hệ tính toán và những quan hệ phi-tính toán giữa các đoạn thẳng cũng có những luật của riêng nó mà không phải là những luật nội tại trong một tam giác. Như vậy, để có một 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 C-Object ta cần phải xem xét khái niệm C-Object trong một hệ thống các khái niệm C-Object cùng với các loại sự kiện, các loại quan hệ khác nhau và các dạng luật khác nhau liên quan đến chúng. Ta sẽ xem xét một mô hình tri thức như thế và gọi nó là mô hình tri thức về các C-Object.

3.2 MÔ HÌNH TRI THỨC CÁC ĐỐI TƯỢNG TÍNH TOÁN

Trong phần nầy sẽ xem xét một 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. Mô hình nầy sẽ được gọi là mô hình tri thức về các C-Object (nghĩa là các đối tượng tính toán).

3.2.1 Mô hình tri thức

Ta gọi một mô hình tri thức các C-Object , viết tắt là một mô hình COKB (Computational Objects Knowledge Base), là một hệ thống (C, H, R, Ops, Rules) gồm:

1. <u>Một tập hợp C các khái niệm về các C-Object</u>.

   Mỗi khái niệm là một lớp C-Object có cấu trúc và được phân cấp theo sự thiết lập của cấu trúc đối tượng:

   [1] Các biến thực.

   [2] Các đối tượng cơ bản có cấu trúc rỗng hoặc có cấu trúc gồm một số thuộc tính thuộc kiểu thực (ví dụ như DIEM không có thuộc tính giá trị thực trong hình học phẳng). Các đối tượng loại nầy làm nền cho các đối tượng cấp cao hơn.

   [3] Các đối tượng C-Object cấp 1. Loại đối tượng nầy có một thuộc tính loại <real> và có thể được thiết lập từ một danh sách nền các đối tượng cơ bản. Ví dụ: DOAN[A,B] và GOC[A,B,C] trong đó A, B, C là các đối tượng cơ bản loại DIEM.

   [4] Các đối tượng C-Object cấp 2. Loại đối tượng nầy có các thuộc tính loại real và các thuộc tính thuộc loại đối tượng cấp 1, và đối tượng có thể được thiết lập trên một danh sách nền các đối tượng cơ bản. Ví dụ: TAM_GIAC[A,B,C] và TU_GIAC[A,B,C,D], trong đó A, B, C, D là các đối tượng cơ bản loại DIEM.

Cấu trúc bên trong của mỗi lớp đối tượng gồm:

- Kiểu đối tượng. Kiểu nầy có thể là loại kiểu thiết lập trên một danh sách nền các đối tượng cơ bản.
- Danh sách các thuộc tính, mỗi thuộc tính có kiểu thực, kiểu đối tượng cơ bản hay kiểu đối tượng cấp thấp hơn.
- Quan hệ trên cấu trúc thiết lập. Quan hệ nầy thể hiện các sự kiện về sự liên hệ giữa đối tượng và các đối tượng nền (tức là các đối tượng thuộc danh sách đối tượng nền).
- Tập hợp các điều kiện ràng buộc trên các thuộc tính.

- Tập hợp các tính chất nội tại liên quan đến các thuộc tính của đối tượng. Mỗi tính chất nầy cho ta một sự kiện của đối tượng.
- Tập hợp các quan hệ suy diễn - tính toán. Mỗi quan hệ thể hiện một qui luật suy diễn và cho phép ta có thể tính toán một hay một số thuộc tính nầy từ một số thuộc tính khác của đối tượng.
- Tập hợp các luật suy diễn trên các loại sự kiện khác nhau liên quan đến các thuộc tính của đối tượng hay bản thân đối tượng. Mỗi luật suy diễn có dạng:

  {các sự kiện giả thiết}⇒{các sự kiện kết luận}

Cùng với cấu trúc trên, đối tượng còn được trang bị các hành vi cơ bản trong việc giải quyết các bài toán suy diễn và 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 (nếu đối tượng được thiết lập trên một danh sách các đối tượng nền nào đó). Các hành vi cơ bản nầy của đối tượng C-Object sẽ được xem xét chi tiết hơn trong các mục sau.

2. <u>Một tập hợp H các quan hệ phân cấp giữa các loại đối tượng</u>.

   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.

3. <u>Một tập hợp R các khái niệm về các loại quan hệ trên các C-Object</u>.

   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. Ví dụ: Quan hệ cùng phương trên 2 đoạn thẳng có các tính chất phản xạ, đối xứng và bắc cầu.

4. Một tập hơp 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.

5. Một tập hơp Rules gồm các luật được phân lớp.

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. Như vậy, một luật r có thể được mô hình dưới dạng:

$$r : \{sk_1, sk_2, ..., sk_n\} \Rightarrow \{ sk_1, sk_2, ..., sk_m \}$$

Để mô hình luật dẫn trên có hiệu lực trong cơ sở tri thức và để có thể khảo sát các thuật giải để giải quyết các bài toán, ta cần định nghĩa các dạng sự kiện khác nhau trong các luật. Dưới đây là định nghĩa cho 6 loại sự kiện khác nhau được xem xét trong mô hình.

- **Định nghĩa 3.2**: (Các loại sự kiện)

(1) Sự kiện thông tin về loại của một đối tượng. Ta biểu diễn sự kiện nầy bởi cấu trúc danh sách: [<object>, <loại object>].

Ví dụ: [Ob1, "TAM_GIAC"] hay [Ob2, "TAM_GIAC[A,B,C]"].

(2) Sự kiện về tính xác định của một đối tượng (các thuộc tính coi như đã biết) hay của một thuộc tính. Ta biểu diễn sự kiện loại nầy bởi tên của đối tượng hay tên thuộc tính của đối tượng: <object> |

<object>.<thuoc_tinh>.

Ví dụ: Obj, Obj.a, Obj.DOAN[A,B].

* Ghi chú: trường hợp đối tượng được cấu thành từ các đối tượng khác thì <thuoc_tinh> có thể được viết theo phương thức cấu trúc.

Ví dụ: O1 : TAM_GIAC[A,B,C];

O1.GocA có thể viết là O1.GOC[C,A,B]

O1.a có thể viết là O1.DOAN[B,C]

(3) Sự kiện về sự xác định của một thuộc tính hay một đối tượng thông qua một biểu thức hằng. Ta viết: <object> = <bieu_thuc_hang>, hay <object>.<thuộc tính> = <bieu_thuc_hang>.

Ví dụ: Obj.a = 5, DOAN[A,B] = m, GOC[A,B,C] = π/3.

(4) Sự kiện về sự bằng nhau giữa một đối tượng hay một thuộc tính với một đối tượng hay một thuộc tính khác. Sự bằng nhau nầy giữa 2 đối tượng sẽ được hiểu theo nghĩa là các thuộc tính tương ứng của chúng bằng nhau. Sự kiện loại nầy sẽ được viết dưới dạng:

<bject> | <object>.<thuoc_tinh> = <bject> | <object>.<thuoc_tinh>

Ví dụ: Ob1.a = Ob2.a, Ob1.a = DOAN[C,D], Ob1 = Ob2.

(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. Loại sự kiện nầy có dạng:

<object> | <object>.<thuoc_tinh> =

<bieu thuc theo cac object hay thuoc tinh khac>.

Ví dụ: O1.a = O2.a + 2*O2.b

(6) Sự kiện về một quan hệ trên các đối tượng hay trên các thuộc tính của các đối tượng. Sự kiện loại nầy có thể được biểu diễn bởi cấu trúc danh sách có dạng:

[<ten quan he>, <object 1>, <object 2>, ...], hay

[<ten quan he>, {<object 1>, <object 2>, ... }]

(cách biểu diễn thứ 2 được sử dụng cho trường hợp quan hệ có tính đối xứng và truyền trên các phần tử khác nhau đôi một).

Ví dụ: [“SSONG”, DOAN[A,B], DOAN[C,D]],

[“THUOC”, M, DOAN[B,C]], [“VUONG”, Ob1.a, Ob2.b].

3.2.2 Ví dụ về một mô hình tri thức các C-Objects

Trong mục nầy chúng ta nêu lên một ví dụ áp dụng: Biểu diễn tri thức về các tam giác và tứ giác trong hình học phẳng theo mô hình tri thức về các C-Object. Một phần lớn kiến thức về hình học giải tích 3 chiều hay kiến thức về các phản ứng hóa học cũng có thể được biểu diễn theo mô hình nầy. Cách biểu diễn kiến thức theo mô hình nầy có nhiều ưu điểm thuận lợi cho việc thiết kế một cơ sở tri thức truy cập được dễ dàng bởi các môđun quản trị tri thức cũng như các môđun giải toán và tra cứu kiến thức. Đặc biệt là mô hình giúp ta có thể thiết kế các thuật giải để giải toán tự động. Phần cuối của chương nầy sẽ trình bày việc thiết kế các hành vi cho một C-Object, tức là thiết kế các thuật giải để giải các bài toán về một đối tượng C-Object.

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 COKB với các thành phần như dưới đây.

[1] Các khái niệm về các C-Object gồm:

- Khái niệm cơ bản là khái niệm *điểm*.
- Các đối tượng C-Object cấp 1: *đoạn*, *góc*. Mỗi đoạn có một thuộc tính giá trị thực, đó là độ dài của đoạn; Mỗi đoạn có thể được thiết lập từ 2 điểm. Mỗi góc có một thuộc tính giá trị thực, đó là số đo của góc; Mỗi góc có thể được thiết lập từ 3 điểm.

- Các đối tượng C-Object cấp 2: các loại tam giác và các loại tứ giác. Các loại tam giác bao gồm "tam giác", "tam giác cân", "tam giác vuông", "tam giác vuông cân" và "tam giác đều". Các loại tứ giác bao gồm "tứ giác", "hình thang", "hình thang vuông", "hình thang cân", "hình bình hành", "hình chữ nhật", "hình thoi" và "hình vuông".

Dưới đây là các ví dụ về một phần cấu trúc của các đối tượng thuộc lớp "TAM_GIAC" và lớp "HINH_VUONG" được thể hiện trong các văn bản có cấu trúc:

```
begin_object: TAM_GIAC[A,B,C];
   A, B, C : DIEM;
begin_variables
     GocA : GOC[C,A,B];
     GocB : GOC[A,B,C];
     GocC : GOC[B,C,A];
     a : DOAN[B,C];
     b : DOAN[A,C];
     c : DOAN[A,B];
     ha,hb,hc,ma,mb,mc,pa,pb,pc : DOAN;
     S,p,R,r,ra,rb,rc : real;
end_variables

begin_constraints
   S > 0;
     ...
end_constraints

begin_properties
end_properties

begin_computation_relations

   begin_relation
     flag = 1
     Mf    ={GocA,GocB,GocC}
     rf    =1
     vf    ={}
     expf =`  GocA + GocB + GocC = Pi `
     cost=2
   end_relation
```

```
     begin_relation
       flag = 0
       Mf    ={a, b, c, GocA}
       rf    =1
       vf    ={a}
       expf  =`  a^2 = b^2 + c^2 - 2*b*c*cos(GocA)`
       cost=19
     end_relation

     begin_relation
       flag = 1
       Mf  ={ a, b, GocA, GocB}
       rf    =1
       vf   ={}
       expf = `a*sin(GocB) = b*sin(GocA)  `
       cost=8
     end_relation
     ...
  end_computation_relations

  begin_rules

    begin_rule
       kind_rule = "";
       hypothesis_part:
         {GocA = GocB}
       goal_part:
         {a = b}
    end_rule

    begin_rule
       kind_rule = "";
       hypothesis_part:
         {a = b}
       goal_part:
         {GocA = GocB }
    end_rule

    begin_rule
       kind_rule = "";
       hypothesis_part:
         {a^2 = b^2+c^2}
       goal_part:
         {GocA=pi/2}
    end_rule

    begin_rule
```

```
      kind_rule = "";
      hypothesis_part:
        {GocA=pi/2}
      goal_part:
        {a^2 = b^2+c^2}
   end_rule

   begin_rule
      kind_rule = "xac_dinh_doi_tuong";
      hypothesis_part:
        {a, b, c}
      goal_part:
        {"Object"}
   end_rule
   ...

end_rules

end_object
```

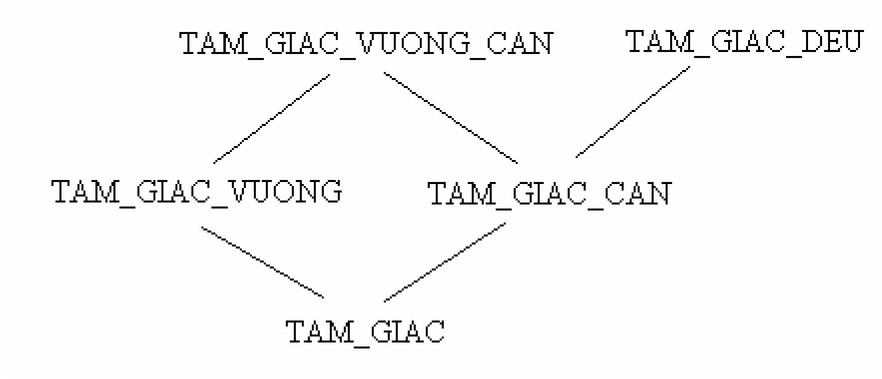
```
begin_object: HINH_VUONG[A,B,C,D]
   A, B, C, D : DIEM;

begin_variables
     a : DOAN[A,B]; # canh
     b : DOAN[B,C]; # canh
     c : DOAN[C,D]; # canh
     d : DOAN[D,A]; # canh
     c1 : DOAN[A,C]; # duong cheo
     c2 : DOAN[B,D]; # duong cheo
     GA : GOC[D,A,B]; # goc
     GB : GOC[A,B,C]; # goc
     GC : GOC[B,C,D]; # goc
     GD : GOC[C,D,A]; # goc
     S , p : real;
end_variables

begin_constraints
   S > 0;
   p > 0;
end_constraints

begin_properties
   GA = Pi / 2;
   GB = Pi / 2;
   GC = Pi / 2;
```

```
     GD = Pi / 2;
     b = a;
     c = a;
     d = a;
      ["VUONG", a, b];
      ["VUONG", b, c];
      ["VUONG", c, d];
      ["VUONG", d, a];
      ["SSONG", a, c];
      ["SSONG", b, d];
      ["VUONG", c1, c2];
  end_properties

  begin_computation_relations

      begin_relation 0
         flag=1
         Mf={a,b}
         rf=1
         vf={}
         expf= `b = a`
         cost = 1
      end_relation

      begin_relation 1
         flag=1
         Mf={a,c}
         rf=1
         vf={}
         expf=`c=a`
         cost=1
      end_relation

      begin_relation 2
         flag=1
         Mf={a,d}
         rf=1
         vf={}
         expf=`d=a`
         cost=1
      end_relation

      begin_relation 3
         flag=1
         Mf={a,c1}
         rf=1
         vf={}
```

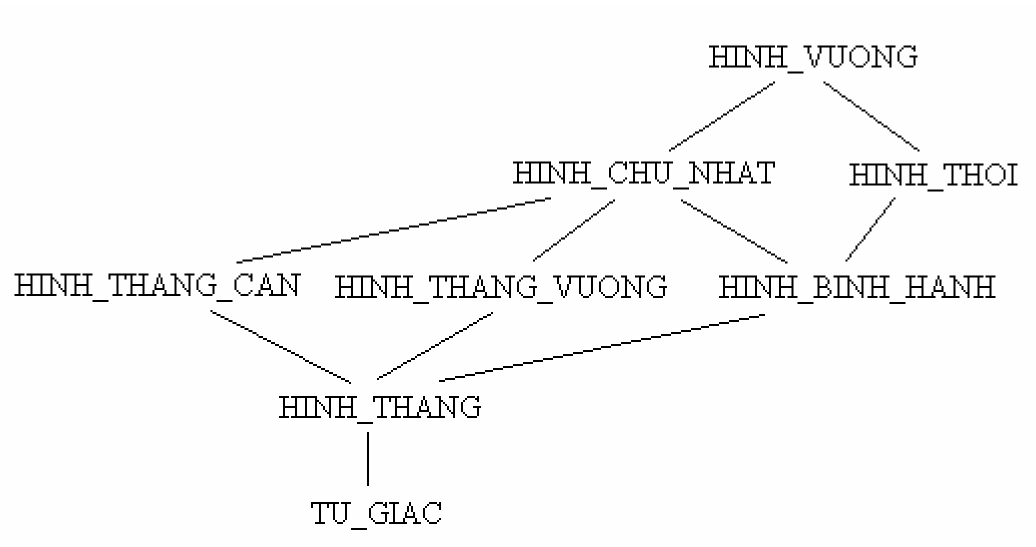
```
        expf=`c1 = a*sqrt(2)`
        cost=12
     end_relation

     begin_relation 4
        flag=1
        Mf={c1,c2}
        rf=1
        vf={}
        expf=`c1 = c2`
        cost=1
     end_relation
      ...
end_computation_relations

begin_rules

   begin_rule 1
      kind_rule = "xac_dinh_doi_tuong";
      hypothesis_part:
        {a}
      goal_part:
        {"Object"}
   end_rule
end_rules

end_object
```

[2] <u>Các quan hệ phân cấp giữa các loại đối tượng</u>:

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. Chẳng hạn, một tam giác cân cũng là một tam giác, một hình bình hành cũng là một tứ giác. Hệ thống quan hệ phân cấp các khái niệm hình học này có thể được thể hiện trên các biểu đồ thứ tự dưới đây.



Hình 3.1 Biểu đồ Hasse thể hiện quan hệ phân cấp của các khái niệm tam giác



Hình 3.2 Biểu đồ Hasse thể hiện quan hệ phân cấp của các khái niệm tứ giác

[3] <u>Các khái niệm về các loại quan hệ giữa các loại đối tượng</u>:

Các quan hệ giữa các khái niệm bao gồm các loại quan hệ như:

- Quan hệ thuộc về của 1 điểm đối với một đoạn thẳng.
- Quan hệ trung điểm của một điểm đối với một đoạn thẳng.
- Quan hệ song song giữa 2 đoạn thẳng.
- Quan hệ vuông góc giữa 2 đoạn thẳng.
- Quan hệ bằng nhau giữa 2 tam giác.

[4] 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".

[5] Các luật:

Các luật trên các loại sự kiện khác nhau chẳng hạn như các luật được liệt kê bên dưới. Các luật được viết dưới dạng văn bản có cấu trúc:

(1)
```
begin_rule
        kind_rule = "kieu_thiet_lap";
          A, B : DIEM;
        goal_part:
             {DOAN[A,B] = DOAN[B,A]}
        end_goal_part
     end_rule
```

Luật nầy nói rằng: Với 2 điểm A, B ta có đoạn AB = đoạn BA.

(2)
```
begin_rule
        kind_rule = "chuyen_kieu_doi_tuong";
          A, B, C : DIEM;
           Ob : TAM_GIAC[A,B,C];
       hypothesis_part:
          {Ob.DOAN[A,B] = Ob.DOAN[A,C]}
       end_hypothesis_part
        goal_part:
             {[Ob, TAM_GIAC_CAN[A,B,C]]}
        end_goal_part
     end_rule
```

Luật nầy nói rằng: Một tam giác ABC có 2 cạnh AB và AC bằng nhau thì tam giac là tam giác cân tại A.

(3)
```
begin_rule
     kind_rule = "SSONG,VUONG";
```

```
        a, b, c: DOAN;
      hypothesis_part:
           {["SSONG",a,b], ["VUONG",a,c]}
      end_hypothesis_part
      goal_part:
         {["VUONG",b,c]}
      end_goal_part
    end_rule
```

Luật nầy nói rằng: Với 3 đoạn thẳng a, b và c, nếu a // b và a ⊥ c thì ta có b ⊥ c.

(4)
```
begin_rule
       kind_rule = "";
         M, A, B: DIEM;
      hypothesis_part:
          {["THUOC",M,DOAN[A,B]]}
      end_hypothesis_part
       goal_part:
           {DOAN[A,B] = DOAN[A,M] + DOAN[M,B]}
       end_goal_part
    end_rule
```

Luật nầy nói rằng: Với 3 điểm M, A và B, nếu điểm M thuộc đoạn AB thì ta có (số đo đoạn AB) = (số đo đoạn AM) + (số đo đoạn MB).

(5)
```
begin_rule
  kind_rule = "TAM_GIAC = TAM_GIAC";
  Ob1, Ob2 : TAM_GIAC;
  hypothesis_part:
     {Ob1.a = Ob2.a, Ob1.b = Ob2.b, Ob1.c = Ob2.c }
  end_hypothesis_part
  goal_part:
     {Ob1 = Ob2}
```

```
    end_goal_part
  end_rule
```

Luật nầy nói rằng: Nếu 2 tam giác có 3 cạnh tương ứng bằng nhau thì chúng bằng nhau.

3.3 TỔ CHỨC CƠ SỞ TRI THỨC VỀ CÁC C-OBJECT

3.3.1 Các thành phần

Cơ sở tri thức về các C-Object theo mô hình COKB có thể được tổ chức bởi một hệ thống tập tin văn bản có cấu trúc thể hiện các thành phần trong mô hình tri thức. Có thể thiết kế hệ thống các tập tin nầy gồm những tập tin như sau:

[1] Tập tin "Objects.txt" lưu trữ các định danh (hay tên gọi) cho các khái niệm về các loại đối tượng C-Object.

[2] Tập tin "RELATIONS.txt" lưu trữ thông tin về các loại quan hệ khác nhau trên các loại C-Object.

[3] Tập tin "Hierarchy.txt" lưu lại các biểu đồ Hasse thể hiện quan hệ phân cấp đặc biệt hóa trên các khái niệm.

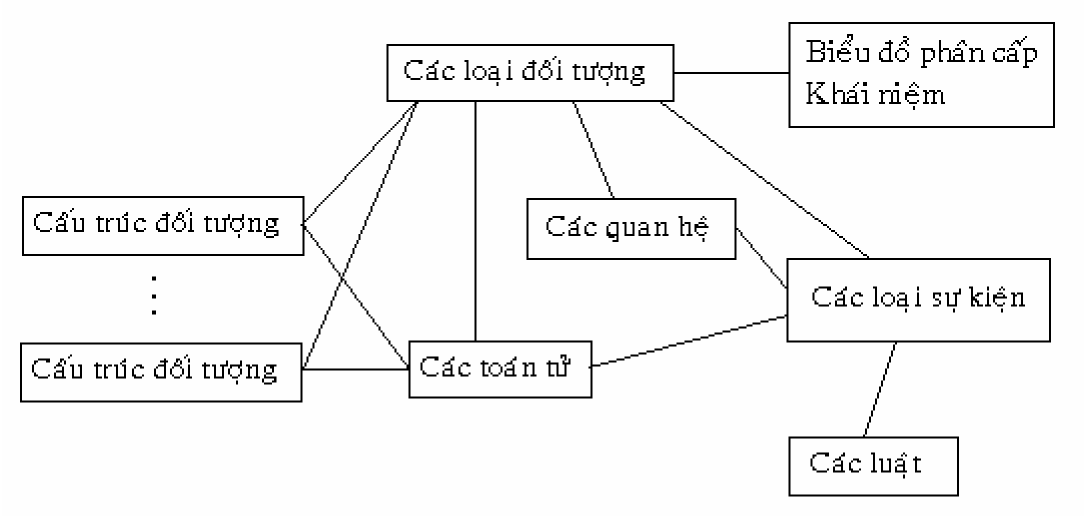
[4] Các tập tin với tên tập tin có dạng "<tên khái niệm C-Object>.txt" để lưu trữ cấu trúc của loại đối tượng <tên khái niệm C-Object>. Ví dụ: tập tin "TAM_GIAC.txt" lưu trữ cấu trúc của loại đối tượng tam giác.

[5] Tập tin "Operators.txt" lưu trữ các thông tin về các toán tử trên các đối tượng.

[6] Tập tin "FACTS.txt" lưu trữ thông tin về các loại sự kiện khác nhau.

[7] Tập tin "RULES.txt" lưu trữ hệ luật của cơ sở tri thức.

Mối liên hệ về cấu trúc thông tin trong cơ sở tri thức có thể được minh họa trên sơ đồ sau đây:



Hình 3.3 Biểu đồ liên hệ giữa các thành phần trong COKB

3.3.2 Cấu trúc của các tập tin lưu trữ các thành phần trong COKB

Các tập tin lưu trữ các thành phần trong cở sở tri thức các C-Object được ghi dưới dạng các văn bản có cấu trúc dựa trên một số từ khóa và qui ước về cú pháp khá đơn giản và tự nhiên. Dưới đây là phần liệt kê cấu trúc của các tập tin:

- <u>Cấu trúc tập tin "Objects.txt"</u>

```
begin_Objects
  <tên lớp đối tượng 1>
  <tên lớp đối tượng 2>
   ...
end_Objects
```

- <u>Cấu trúc tập tin "RELATIONS.txt"</u>

```
begin_Relations
  [<tên quan hệ>, <loại đối tượng>, <loại đối tượng>, ... ],
  {<tính chất>, <tính chất>, ...}
  [<tên quan hệ>, <loại đối tượng>, <loại đối tượng>, ... ],
  {<tính chất>, <tính chất>, ...}
```

```
  ...
end_Relations
```

- Cấu trúc tập tin “Hierarchy.txt”

```
begin_Hierarchy
  [<tên lớp đối tượng cấp cao>, <tên lớp đối tượng cấp thấp>]
  [<tên lớp đối tượng cấp cao>, <tên lớp đối tượng cấp thấp>]
  ...
end_Hierarchy
```

- Cấu trúc tập tin “<tên khái niệm C-Object>.txt”

```
begin_object: <tên khái niệm C-Object> [các đối tượng nền]
     <các đối tượng nền>  :  <kiểu>;
     <các đối tượng nền>  :  <kiểu>;
     ...
begin_variables
  <tên thuộc tính> : <kiểu>;
  <tên thuộc tính> : <kiểu>;
  ...
end_variables
begin_constraints
  ...
end_constraints
begin_properties
  <sự kiện>
  <sự kiện>
  ...
end_properties
```

```
begin_computation_relations
    begin_relation
        flag=<0 hoặc 1>
        Mf={các thuộc tính}
        rf=1
        vf={ghi thuộc tính kết quả nếu flag = 0}
        expf= `biểu thức tính toán`
        cost = <trọng số của sự tính toán>
    end_relation
    ...
end_computation_relations
begin_rules
  begin_rule
     kind_rule = "<loại luật>";
     hypothesis_part:
        {các sự kiện giả thiết của luật}
     goal_part:
        { các sự kiện kết luận của luật hoặc là "Object"}
  end_rule
  ...
end_rules
end_object
begin_inside_net
   parameters: ...
   objects:
     ...
   facts:
```

```
      ...
end_inside_net
```

- Cấu trúc tập tin “Operators.txt”

```
begin_Operators
  [<toán tử>, [các kiểu toán hạng], <kiểu kết quả>, <quitắc tính toán>]
  [<toán tử>, [các kiểu toán hạng], <kiểu kết quả>, <quitắc tính toán>]
   ...
end_Operators
```

- Cấu trúc tập tin “FACTS.txt”

```
begin_Facts
  1, <cấu trúc sự kiện>, <cấu trúc sự kiện>, ...
  2, <cấu trúc sự kiện>, <cấu trúc sự kiện>, ...
   ...
end_Facts
```

- Cấu trúc tập tin “RULES.txt”

```
begin_rules
  begin_rule
     kind_rule = "<loại luật>";
     <các tên đối tượng>  :  <kiểu đối tượng>;
     <các tên đối tượng>  :  <kiểu đối tượng>;
       ...
     hypothesis_part:
       {các sự kiện giả thiết của luật}
     goal_part:
       { các sự kiện kết luận của luật hoặc là  "Object"}
  end_rule
  ...
```

```
end_rules
```

Cách tổ chức cơ sở tri thức như thế cho ta một cấu trúc tri thức rõ ràng và tách bạch với đầy đủ các thông tin cùng với các liên hệ khác nhau rất đa dạng. Mô hình COKB được xây dựng có các ưu điểm sau đây:

- **Thích hợp cho việc thiết kế một cớ sở tri thức với các khái niệm có thể được biểu diễn bởi các C-Object.**
- **Cấu trúc tường minh giúp dễ dàng thiết kế các môđun truy cập cơ sở tri thức.**
- **Tiện lợi cho việc thiết kế các mô đun giải bài 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.**

3.4 GIẢI TOÁN C-OBJECT (COBJECT-SOLVER)

Trong phần nầy chúng ta sẽ thiết kế các xử lý cơ bản thể hiện các hành vi mà chúng ta cài đặt cho các đối tượng C-Object. Các hành vi nầy bao gồm:

- Khả năng giải quyết một bài toán có dạng:

$$GT \Rightarrow KL$$

  Trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của đối tượng. Khả năng nầy bao gồm việc xem xét tính giải được của bài toán, tìm lời giải cho bài toán, và thực hiện việc tính toán.
- Xét tính xác định của đối tượng dựa trên một số sự kiện cho trước.
- Khả năng giải quyết một bài toán có dạng mở rộng:

$$GT \Rightarrow KL$$

  Trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của đối tượng và trên các đối tượng thiết lập trên các đối tượng nền của đối tượng.

Như vậy, ta có các vấn đề cơ bản được đặt ra cho việc giải toán một đối tượng C-Object như sau:

- Vấn đề 1: Xét tính giải được của bài toán GT $\Rightarrow$ 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 $\Rightarrow$ 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 $\Rightarrow$ 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.

Đối với bài toán dạng mở rộng trên một đối tượng C-Object, ta sẽ giải quyết dựa trên một mô hình mạng các đối tượng C-Object sẽ được trình bày trong chương sau. Ngoài ra, khả năng giải quyết các vấn đề cơ bản trên một đối tượng C-Object sẽ được tăng cường khi ta thiết kế một mạng đối tượng liên quan trong một C-Object.

3.4.1 Giải quyết vấn đề cơ bản 1

Trong mục nầy chúng ta sẽ nêu lên thuật giải tổng quát cho vấn đề 1. Ý tưởng cơ bản là thực hiện một quá trính suy diễn tiến kết hợp với một số qui tắc heuristic nhằm tăng cường tốc độ giải quyết bài toán và đạt được một lời giải tốt nhanh hơn. Để thiết kế thuật giải nầy ta cần định nghĩa một số khái niệm liên quan bao gồm các khái niệm: "*sự hợp nhất*" của các sự kiện, một "*bước giải*", một "*lời giải*" và "*sự giải được*".

- **Định nghĩa 3.3**: Ta nói 2 sự kiện *fact1* và *fact2* là *hợp nhất* khi có các điều kiện sau đây:

(1) *fact1* và *fact2* là cùng loại k, và

(2) *fact1* = *fact2* nếu k = 1, 2, 6.

[*fact1*[1], {*fact1*[2..nops(*fact1*)]}] = [*fact2*[1], {*fact2*[2..nops(*fact2*)]}]

nếu k = 6 và quan hệ trong sự kiện *fact1* có tính "đối xứng".

lhs(*fact1*) = lhs(*fact2*) và compute(rhs(*fact1*)) = compute(rhs(*fact2*)) nếu k =3.

(lhs(*fact1*) = lhs(*fact2*) và rhs(*fact1*) = rhs(*fact2*)) hay

(lhs(*fact1*) = rhs(*fact2*) và rhs(*fact1*) = lhs(*fact2*)) nếu k = 4.

evalb(simplify(expand(lhs(*fact1*)-rhs(*fact1*)- lhs(*fact2*)+rhs(*fact2*))) = 0)

hay

evalb(simplify(expand(lhs(*fact1*)-rhs(*fact1*)+ lhs(*fact2*)-rhs(*fact2*))) = 0)

nếu k = 5.

Trong đó các ký hiệu có ý nghĩa như sau:

{. . .} : một tập hợp.

[. . .] : một danh sách (list).

L[1] : thành phần thứ nhất của danh sách L.

L[i..j] : dãy các thành phần từ vị trí i đến vị trí j trong danh sách L.

nops(L) : số thành phần của danh sách L.

lhs(eqn) : vế trái của đẳng thức <eqn>,

rhs(eqn) : vế phải của đẳng thức <eqn>,

compute(expr) : kết quả tính toán của biểu thức <expr>,

expand(expr) : khai triển biểu thức <expr>,

simplify(expr) : đơn giản biểu thức <expr>,

evalb(expr1 = expr2) : đánh giá việc so sánh bằng nhau giữa biểu thức <expr1> và biểu thức <expr2>.

Các ví dụ về các sự kiện hợp nhất với nhau:

[Ob1, “TAM_GIAC”] và [Ob1, “TAM_GIAC”].

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.

a = b và a = b.

Ob1 = Ob2 và Ob2 = Ob1.

a^2 = b^2 + c^2 và b^2 = a^2 – c^2.

a // b và b // a.

- **Định nghĩa 3.4**: Ta gọi 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* sau:

(1) “Deduce_from3”: Suy ra một sự kiện loại 2 từ một sự kiện loại 3, hay suy ra các sự kiện loại 2 từ các sự kiện loại 3.

(2) “Deduce_from453s”: Suy ra một sự kiện mới từ sự kiện <fact> loại 4 hay loại 5 và từ một số sự kiện loại 3 bằng phép thay thế các biến trong các sự kiện loại 3 vào sự kiện <fact>.

(3) “Deduce_from452s”: Từ một sự kiện <fact> loại 4 hay 5 và tập hợp <Fact2s> gồm một số sự kiện loại 2 ta suy ra một sự kiện loại 2 bằng cách lấy phần tử trong tập hợp S = SetVars(fact) – SetVars(Fact2s) trong trường hợp tập hợp S nầy có 1 phần tử. Ở đây ký hiệu SetVars(expr) có nghĩa là tập hợp tất cả các biến có mặt trong <expr>.

(4) “AppCRela_1” : Áp dụng một quan hệ tính toán f trên một số sự kiện loại 2 và loại 3 theo qui tắc tương tự như trong mạng suy diễn và tính toán để suy ra một sự kiện mới thuộc loại 3, 4, hoặc 5. Dạng suy luận nầy chỉ có thể suy ra sự kiện mới khi ta có điều kiện: trong f có đúng một biến không có mặt trong các sự kiện loại 2 và các sự kiện loại 3.

(5) "AppCRela_2" : Áp dụng một quan hệ tính toán f trên một số sự kiện loại 2 và loại 3 để suy ra một sự kiện mới thuộc loại 3, 4, hoặc 5 bằng cách thay thế một số biến trong f mà có mặt trong các sự kiện loại 3 rồi tính một biến theo các biến khác trong f. Dạng suy luận này không đòi hỏi điều kiện như trong dạng suy luận (4) mà vẫn có thể dẫn ra sự kiện mới.

(6) "AppCRela_3" : Áp dụng một quan hệ tính toán f trên một sự kiện loại loại 3 bằng cách thay thế một biến trong f mà có mặt trong sự kiện loại 3 để tạo ra một sự kiện mới với số biến ít hơn.

(7) Áp dụng một luật suy diễn r.

(8) Giải hệ phương trình đơn giản gồm n phương trình n ẩn.

Ví dụ 3.1: Một số ví dụ về các bước giải:

1/ $\{a^2 = m^2 + 2\,m + 1,\ GocA = \frac{1}{2}\pi,\ c = 4\} \Rightarrow \{a,\ c,\ GocA\}$ (Deduce_from3).

2/ $\{a^2 = m^2 + 2\,m + 1,\ GocA = \frac{1}{2}\pi,\ c = 4\}$, $b = \sin(GocA)\,a \Rightarrow b = \sin\left(\frac{1}{2}\pi\right)(m+1)$

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

$f := ["\ \ relation\ \ 0", 1, \{GocA, GocB, GocC\}, 1, \{\ \}, \ \ GocA + GocB + GocC = Pi, 2]$

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

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

$f := ["\ \ relation\ \ 0", 1, \{GocA, GocB, GocC\}, 1, \{\ \}, \ \ GocA + GocB + GocC = Pi, 2]$

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

5/ $b^2 = m^2 + 2\,m + 1,\ GocA = \frac{1}{2}\pi$,

$f := ["\ \ relation\ \ 1", 0, \{a, b, c, GocA\}, 1, \{a\}, \ \ a\verb|^|2 = b\verb|^|2 + c\verb|^|2 - 2*b*c*cos(GocA), 19]$

$\Rightarrow a = \sqrt{m^2 + 2\,m + 1 + c^2}$

6/ $S = 2\,a$, $f := ["\ \ relation\ \ 11", 1, \{a, ha, S\}, 1, \{\ \}, \ S = a*ha/2, 4] \Rightarrow ha = 4$

7/ $a^2 = b^2 + c^2$,

$f := ["\ \ \text{relation}\ \ 1", 0, \{a, b, c, GocA\}, 1, \{a\},\ \ a\verb|^|2 = b\verb|^|2 + c\verb|^|2 - 2*b*c*cos(GocA), 19]$

$\Rightarrow\ GocA = \frac{1}{2}\pi$

8/ $b^2 = a^2 - c^2$, rule : if $a^2 = b^2 + c^2$ then $GocA = \frac{1}{2}\pi$ $\Rightarrow$ $GocA = \frac{1}{2}\pi$

- **Định nghĩa 3.5**: Xét bài toán GT $\Rightarrow$ KL, trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của một đối tượng C-Object. Ta gọi một dãy các bước giải $\{s_1, s_2, \ldots, s_m\}$ là một *lời giải* của bài toán khi ta có KL $\subset$ FACTS, với FACTS là tập hợp tất cả các sự kiện có được sau khi ta lần lượt áp dụng các bước giải $s_1, s_2, \ldots, s_m$ xuất phát từ tập sự kiện GT và quan hệ $\subset$ được hiểu theo nghĩa là "bao hàm hợp nhất" (tức là mọi sự kiện trong KL đều hợp nhất được với một sự kiện nào đó trong FACTS). Ta sẽ nói rằng bài toán là *giải được* khi tồn tại một lời giải cho bài toán.

- **Thuật giải 3.1**: Cho bài toán GT $\Rightarrow$ KL, trong đó GT và KL là các tập hợp những sự kiện trên các thuộc tính của một đối tượng C-Object. Ta có thể xét tính giải được của bài toán trên theo thủ tục sau đây:

  Bước 1: Đặt trạng thái ban đầu cho tập sự kiện <FactSet>, và biến <found> có kiểu Bool cho biết kết quả xem xét sự giải được của bài toán:

  FactSet $\leftarrow$ GT;

  If KL $\subset$ FactSet then Found $\leftarrow$ true

  Else Found $\leftarrow$ false;

  Bước 2: Thực hiện một quá trình dò tìm cách suy diễn tính toán để phát sinh (hay suy ra) sự kiện mới:

  While not found do

  2.1: Tìm kiếm một dạng suy luận có thể áp dụng được trên tập sự kiện FactSet để sinh ra sự kiện mới.

  2.2: If (tìm kiếm ở bước 2.1 thất bại) then break; (ngắt lặp while)

2.3: Gọi tập các sự kiện mới là <news> trong trường hợp tìm kiếm ở bước 2.1 thành công. Cập nhật biến <FactSet> như sau:

FactSet ← FactSet ∪ news;

2.4: If KL ⊂ FactSet then found ← true;

end do;

Bước 3: If found then "Bài toán giải được"

Else "Bài toán không giải được";

Lưu ý: Trong thuật toán trên quan hệ KL ⊂ FactSet được xét theo nghĩa "bao hàm hợp nhất".

- **Các qui tắc heuristic**: Để quá trình tìm kiếm suy diễn và tính toán được nhanh chóng và hiệu quả hơn ta có thể sử dụng một số qui tắc sau đây trong việc tìm kiếm và chọn lựa các dạng suy luận có thể áp dụng được:
  - Qui tắc 1: Ưu tiên áp dụng các dạng suy luận "Deduce_from3", "Deduce_from453s" và "Deduce_from452s".
  - Qui tắc 2: Trong các dạng suy luận "AppCRela_1", "AppCRela_2" và "AppCRela_3" ta ưu tiên áp dụng dạng suy luận "AppCRela_1".
  - Qui tắc 3: Trong các luật ta ưu tiên áp dụng luật khác với luật xác định đối tượng trong trường hợp mục tiêu là sự kiện về tính xác định của các thuộc tính. Ngược lại thì ưu tiên xem xét các luật là luật xác định đối tượng.
  - Qui tắc 4: Thực hiện giải hệ phương trình để xác định giá trị của biến trong các trường hợp đơn giản của hệ n phương trình với n ẩn.

Ví dụ 3.2: Một số ví dụ về bài toán GT ⇒ KL trên đối tượng "TAM_GIAC" và kết quả khảo sát tính giải được của nó:

1/ GT = {a, b, c, GocA = m*(b+c)}, KL = {GocC}: Giải được.

2/ GT = {a, b=1, c, GocA = m*(b+c), GocA = 2*GocB},
KL = {GocB, GocC}: Giải được.

3/ GT = {GocB, c}, KL = {S}: Không giải được.

4/ GT = {GocC, GocA, a}, KL = {p, R}: Giải được.

5/ GT = {a, b, c=a^2+m*p}, KL = { S}: Giải được.

6/ GT = {b, c=a^2+m*p, a = b+1}, KL = { S}: Giải được.

7/ GT = {a, GocA = m*(b+c), GocA = 2*GocB, a^2=b^2+c^2},
KL = {GocB,GocC}: Giải được.

3.4.2 Giải quyết vấn đề cơ bản 2

Vấn đề 2 được đặt ra như sau: 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. Để giải quyết vấn đề nầy chúng ta cũng có thể thực hiện một quá trình suy diễn tiến với sự vận dụng một số qui tắc heuristic như đã trình bày ở thuật giải 3.1. Quá trính suy diễn nầy có thể sẽ đi đến kết luận rằng bài toán không giải được. Trong trường hợp bài toán giải được, thì trong quá trình tìm kiếm và áp dụng các luật suy diễn cũng như tính toán để phát sinh sự kiện mới ta đã có được một lời giải cho bài toán. Tuy nhiên lời giải nầy có thể có những bước giải dư thừa mà ta cần phải loại bỏ để có được một lời giải tự nhiên hơn. Từ lý giải trên ta có thể tìm một lời giải cho bài toán theo 2 giai đoạn được trình bày trong thuật giải tổng quát như sau:

- **Thuật giải 3.2**: Tìm một lời giải cho bài toán GT ⇒ KL.

- Giai đoạn 1: Tìm một lời giải (nếu có) cho bài toán như sau:

Bước 1: Đặt trạng thái ban đầu cho tập sự kiện đang có <FactSet>, danh sách lời giải <Sol>, và biến <found> có kiểu Bool cho biết kết quả xem xét sự giải được của bài toán:

FactSet ← GT; Sol ← [ ];

If KL ⊂ FactSet then Found ← true

Else Found ← false;

Bước 2: Thực hiện một quá trình dò tìm cách suy diễn tính toán để phát sinh (hay suy ra) sự kiện mới:

While not found do

2.1: Tìm kiếm một dạng suy luận có thể áp dụng được trên tập sự kiện FactSet để sinh ra sự kiện mới.

2.2: If (tìm kiếm ở bước 2.1 thất bại) then

break; (ngắt lặp while)

2.3: Gọi <rform> là dạng luật áp dụng được trên một tập sự kiện <fset> ⊂ <FactSet> và sẽ phát sinh tập các sự kiện mới là <news> trong trường hợp tìm kiếm ở bước 2.1 thành công. Cập nhật các biến như sau:

FactSet ← FactSet ∪ news;

Sol ← [op(Sol), [rform, fset, news]];

2.4: If KL ⊂ FactSet then found ← true;

end do;

Bước 3: If not found then "Bài toán không giải được"

Else tiếp tục sang giai đoạn 2;

Trong bước 2.3 ở trên, ký hiệu op(Sol) có nghĩa là dãy các phần tử (hay thành phần) trong danh sách <Sol>.

- <u>Giai đoạn 2</u>: Giả sử ta có một lời giải <Sol> của bài toán GT ⇒ KL được tìm thấy trong giai đoạn 1 ở trên khi bài toán là giải được. Ta có thể thực hiện việc xem xét và loại bỏ các bước dư thừa trong lời giải <Sol> bằng cách truy ngược theo lời giải, ứng với mỗi bước giải mà sự kiện mới được sinh ra nhưng không cần thiết thì ta loại bỏ. Quá trình nầy có thể được thể hiện một cách chi tiết trong thuật giải sau đây:

Bước 1: Khởi tạo biến <KL1> dùng để ghi lại những sự kiện cần được suy ra, và biến <Sol1> lưu lại những bước giải được giữ lại (không bị loại ra do dư thừa):

KL1 ← KL; Sol1 ← [];

Bước 2: Thực hiện quá trình lần ngược theo lời giải để xem xét bước giải nào cần được giữ lại trong lời giải:

```
For  i := nops(Sol)  downto  1  do
   Step ← Sol[i];
   If  step[3] ∩ KL1 ≠ ∅  then      Begin
      KL1 ← (KL1 – step[3]) ∪ step[2];
      Sol1 ← [step, op(Sol1)];
      End
End do;
```

Bước 3: Thiết lập lời giải mới: Sol ← Sol1;

<u>Ví dụ 3.3</u>: Giải 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 }.

Thực hiện giai đoạn 1 trong thuật giải trên ta sẽ có một lời giải như sau:

1. Suy ra $\{b\}$ từ $\{b = 5\}$ bởi "Deduce_from3"
2. Suy ra $\{GocA = m\,(5 + c)\}$ từ $\{b = 5,\ GocA = m\,(b + c)\}$ bởi "Deduce_from453s"
3. Suy ra $\{GocB = \frac{1}{2}GocA\}$ từ $\{GocA = 2\,GocB\}$ bởi "Deduce_from453s"
4. Suy ra $\{a^2 = 25 + c^2\}$ từ $\{b = 5,\ a^2 = b^2 + c^2\}$ bởi "Deduce_from453s"

5. Suy ra $\{GocA = \frac{1}{2}\pi\}$ từ $\{a^2 = b^2 + c^2\}$ bởi một luật suy diễn

6. Suy ra $\{GocA\}$ từ $\{GocA = \frac{1}{2}\pi\}$ bởi "Deduce_from3"

7. Suy ra $\{GocB = \frac{1}{4}\pi\}$ từ $\{GocB = \frac{1}{2} GocA, GocA = \frac{1}{2}\pi\}$ bởi "Deduce_from453s"

8. Suy ra $\{GocB\}$ từ $\{GocB = \frac{1}{4}\pi\}$ bởi "Deduce_from3"

9. Suy ra $\{GocC = \frac{1}{4}\pi\}$ từ $\{GocA = \frac{1}{2}\pi, GocB = \frac{1}{4}\pi\}$ bởi "AppCRela_1":

$["\text{relation } 0", 1, \{GocB, GocA, GocC\}, 1, \{\ \}, GocA + GocB + GocC = Pi, 2]$

10. Suy ra $\{GocC\}$ từ $\{GocC = \frac{1}{4}\pi\}$ bởi "Deduce_from3"

Thực hiện giai đoạn 2 trong thuật giải, từ lời giải trên ta sẽ có một lời giải rút gọn như sau:

1. Suy ra $\{GocB = \frac{1}{2} GocA\}$ từ $\{GocA = 2\ GocB\}$ bởi "Deduce_from453s"

2. Suy ra $\{GocA = \frac{1}{2}\pi\}$ từ $\{a^2 = b^2 + c^2\}$ bởi một luật suy diễn

3. Suy ra $\{GocB = \frac{1}{4}\pi\}$ từ $\{GocB = \frac{1}{2} GocA, GocA = \frac{1}{2}\pi\}$ bởi "Deduce_from453s"

4. Suy ra $\{GocB\}$ từ $\{GocB = \frac{1}{4}\pi\}$ bởi "Deduce_from3"

5. Suy ra $\{GocC = \frac{1}{4}\pi\}$ từ $\{GocA = \frac{1}{2}\pi, GocB = \frac{1}{4}\pi\}$ bởi "AppCRela_1":

$["\text{relation } 0", 1, \{GocB, GocA, GocC\}, 1, \{\ \}, GocA + GocB + GocC = Pi, 2]$

6. Suy ra $\{GocC\}$ từ $\{GocC = \frac{1}{4}\pi\}$ bởi "Deduce_from3"

<u>Ví dụ 3.4</u>: Giải bài toán GT $\Rightarrow$ KL trên đối tượng "TU_GIAC" với

GT = $\{["\text{SSONG}", b, d], GocA = 2\ GocB\}$

KL = $\{GocB, GocA\}$

Thuật giải 3.2 sẽ cho ta một lời giải như sau:

1/ Suy ra $\{GocB = \pi - GocA, GocD = \pi - GocC\}$ từ $\{["SSONG", b, d]\}$

bởi một luật suy diễn.

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

bởi giải hệ phương trình đơn giản.

Ghi chú: Trong ví dụ nầy, nếu yêu cầu tính diện tích S của tứ giác thì bài toán trở nên khó hơn. Khi đó cần phải xem xét thêm một số liên hệ khác trên các thuộc tính của tứ giác mà trong kiến thức về tứ giác của cơ sở tri thức có thể không thấy. Vấn đề nầy sẽ được giải quyết một cách hiệu quả khi ta xét một mạng các đối tượng “TAM_GIAC” của đối tượng “TU_GIAC” được trình bày trong chương sau.

3.4.3 Giải quyết vấn đề cơ bản 3

Trong mục nầy chúng ta xét đến vấn đề cơ bản 3 đối với việc giải bài toán trên một C-Object. Vấn đề được đặt ra là 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 nầy có thể được giải quyết bằng cách duyệt theo trình tự ngược các bước giải trong một lời giải của bài toán, tức là đi từ bước giải cuối trở về bước giải 1, và xác định các biến cần thay thế trong các sự kiện mới để thực hiện phép thay thế và tính toán trong biểu thức tính toán các thuộc tính mục tiêu.

- **Thuật giải 3.3**: Thực hiện tính toán cho bài toán GT ⇒ KL.

  Bước 1: Thực hiện việc tìm lời giải cho bài toán GT ⇒ KL.

  If (bài toán không có lời giải) then Kết thúc;

Else (tiếp tục sang bước 2).

Bước 2: Giả sử <Sol> là một lời giải tìm được trong bước 1. Khởi tạo các biến như sau:

n ← nops(Sol); // số bước giải

vars ← KL; // Các biến mục tiêu cần tính

exprs ← {}; // Các biểu thức hay công thức tính cho các biến mục tiêu

vars_thay ← {}; // Các biến trung gian cần được thay thế trong các biểu thức tính toán cho các biến mục tiêu.

Bước 3: Duyệt ngược theo các bước giải, xác định các biểu thức tính mục tiêu và thực hiện việc thay thế các biến trung gian.

For i := n downto 1 do

Duyệt qua các sự kiện mới <fact> ở bước giải thứ i, ứng với sự kiện nầy ta thực hiện kiểm tra và xử lý như sau:

Nếu sự kiện là một biến v được suy từ sự kiện có dạng v = bt trong giả thiết GT thì ta đặt cho fact là sự kiện: v = bt;

nếu sự kiện là một đẳng thức (tức là một biểu thức tính toán cho một biến có dạng v = bt) thì ta thực hiện kiểm tra và xử lý như dưới đây:

If v ∈ vars then

Begin

Thay thế biến v trong các biểu thức thuộc <exprs>;

Bổ sung đẳng thức v = bt vào <exprs>;

vars ← vars - {v};

vars_thay ← vars_thay ∪ (tập các biến trong biểu thức <bt>);

End

Else if v ∈ vars_thay then

Begin

Thay thế biến v trong các biểu thức thuộc <exprs>;

vars_thay ← (vars_thay - {v}) ∪

(tập các biến trong biểu thức <bt>);

End

End do;

Bước 4: Đơn giản các biểu thức tính trong tập hợp <exprs>.

Ví dụ 3.5: Trên một đối tượng "TAM_GIAC", cho bài toán GT ⇒ KL với

GT = $\{a, b = 1, GocA = \frac{1}{2}\pi\}$

KL = $\{R, S, c\}$

Hãy thực hiện tính toán.

Thuật giải 3.3 trên sẽ tìm lời giải rồi thực hiện tính toán và cho ta kết quả tính toán như sau:

$$\{c = \sqrt{a^2 - 1},$$
$$S = \frac{1}{4}\sqrt{-(a + 1 + \sqrt{a^2 - 1})(a - 1 - \sqrt{a^2 - 1})(a - 1 + \sqrt{a^2 - 1})(a + 1 - \sqrt{a^2 - 1})},$$
$$R = \frac{1}{2}a\}$$

Nếu sửa lại phần giả thiết là GT = $\{GocA = \frac{1}{2}\pi, a = 2, b = 1\}$ thì kết quả sẽ là:

$$\{R = 1, S = \frac{1}{2}\sqrt{3}, c = \sqrt{3}\}$$

3.4.4 Giải quyết vấn đề cơ bản 4

Bây giờ chúng ta xét đến vấn đề cơ bản 4 đối với việc giải bài toán trên một C-Object. Vấn đề được đặt ra là khảo sát tính xác định của đối tượng dựa trên một tập sự kiện cho trước trên các thuộc tính của đối tượng. Ta có thể giải quyết vấn đề nầy bằng một thuật giải khá đơn giản như sau:

- **Thuật giải 3.4**: Cho một đối tượng C-Object và một tập sự kiện GT. Khảo sát tính xác định của một đối tượng.

  Bước 1: Duyệt qua các luật xác định đối tượng (nếu có) trong tập hợp các luật trong đối tượng và chọn ra một luật có số thuộc tính vế trái chưa biết là thấp nhất. Đặt <vars> là tập hợp các thuộc tính như thế.

  Bước 2: Nếu không có luật xác định đối tượng, thì đặt <vars> là tập hợp tất cả các thuộc tính của đối tượng.

  Bước 3: Xét tính giải được của bài toán GT $\Rightarrow$ vars,

  If (bài toán giải được) then đối tượng xác định

  Else đối tượng chưa xác định;

Ví dụ 3.6: Trên một đối tượng "TAM_GIAC", xét tính xác định của đối tượng theo thuật giải 3.4 dựa trên một tập sự kiện GT cho trước ta có kết quả như sau:

1/ GT = {a, b, GocC}, Kết quả: đối tượng xác định.

2/ GT = { a, GocB, GocC}, Kết quả: đối tượng xác định.

3/ GT = { GocA, GocB, GocC}, Kết quả: đối tượng chưa xác định.

4/ GT = { GocA, GocB, a = b^2+1, b = 2}, Kết quả: đối tượng xác định.

CHƯƠNG 4
MẠNG CÁC C-OBJECT

Mô hình mạng các đối tượng có thể được dùng để biểu diễn các dạng bài toán tổng quát trong mô hình tri thức các đối tượng tính toán. Các kết quả về mô hình này và một số áp dụng được công bố trong các bài báo [58], [60], [63], [64] và [65]. Chương này sẽ trình bày chi tiết về mô hình và các thuật giải để giải bài toán tự động.

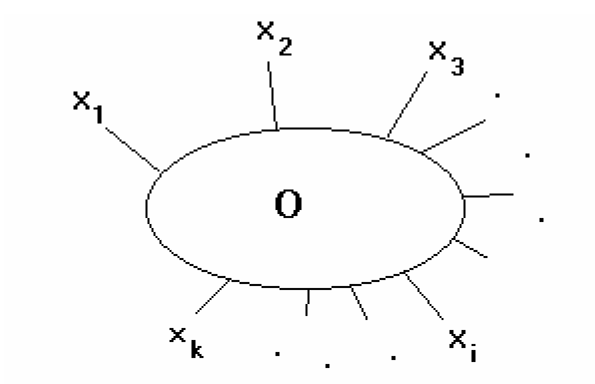
4.1 MẠNG CÁC ĐỐI TƯỢNG TÍNH TOÁN CƠ BẢN:

4.1.1 Mô hình

Trong mục nầy ta xét các đối tượng tính toán (C-Object) cơ bản với 2 thành phần chính là tập thuộc tính và tập các quan hệ tính toán. Mỗi đối tượng tính toán có một tập biến và các quan hệ tính toán nội tại làm cơ sở cho sự thiết lập các hành vi của đối tượng. Tập các biến và tập các quan hệ của đối tượng O lần lượt được ký hiệu là M(O), F(O). Từ đó ta có thể viết :

$$O = (M(O), F(O))$$

Hình vẽ dưới đây biểu diễn cho một đối tượng O, trong đó tập $\{x_1, ..., x_k\} \subseteq$ M(O) là một tập biến đang được quan tâm xem xét của đối tượng O.



Hình 4.1 Đối tượng tính toán cơ bản O.

Ngoài ra đối tượng tính toán O còn có khả năng đáp ứng lại một số thông điệp yêu cầu từ bên ngoài. Trong các khả năng đó của đối tượng tính toán ta có thể kể đến những chức năng sau đây:

(1) Xác định bao đóng (trong đối tượng O) của một tập $A \subseteq M(O)$.

(2) Xác định tính giải được của một bài toán $A \to B$, trong đó $A \subseteq M(O)$, $B \subseteq M(O)$.

(3) Xét tính giải được và tìm một lời giải cho bài toán $A \to B$ trên mạng suy diễn tính toán $(M(O), F(O))$, trong đó $A \subseteq M(O)$, $B \subseteq M(O)$.

(4) Thực hiện tính toán cho bài toán $A \to B$ trên mạng suy diễn tính toán $(M(O), F(O))$ trong trường hợp bài toán giải được.

- **Định nghĩa 4.1**: Một mạng các đối tượng tính toán cơ bản là một bộ (O, M, F) gồm:

(1) $O = \{O_1, O_2, \dots , O_n\}$ là một tập hợp các đối tượng C-Object cơ bản.

(2) M là một tập hợp các thuộc tính của các đối tượng thuộc O.

(3) $F = \{f_1, f_2, \dots , f_m\}$ là một tập hợp các quan hệ tính toán trên các thuộc tính thuộc M.

Đặt $M(O_i)$ = tập hợp tất cả các thuộc tính của đối tượng O_i

$$M(O) = \bigcup_{i=1}^{n} M(O_i)$$

$M(f_i)$ = tập hợp các biến trong quan hệ f_i.

$$M(F) = \bigcup_{i=1}^{m} M(f_i).$$

$$M_i = M \cap M(O_i), \qquad i=1,2, \dots , m.$$

Ta có:

$$\bigcup_{i=1}^{n} M(O_i) \supseteq M \supseteq \bigcup_{i=1}^{m} M(f_i), \text{ hay } M(O) \supseteq M \supseteq M(F).$$

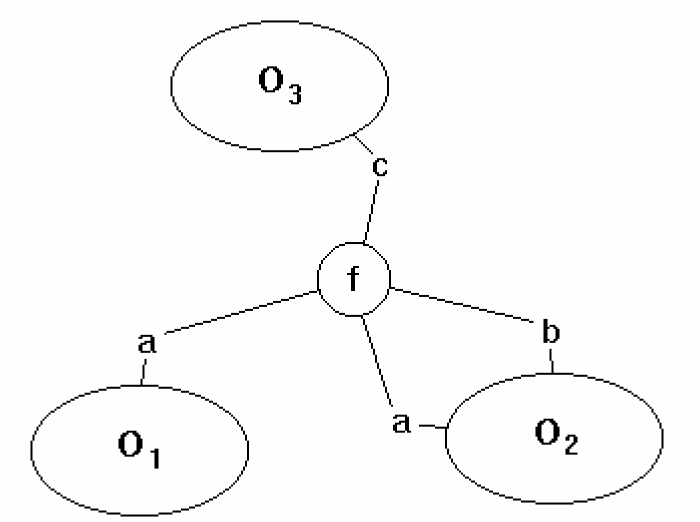
Nhận xét rằng (M, F) là một mạng suy diễn tính toán.

Hai ví dụ dưới đây sẽ minh họa cho một quan hệ tính toán $f \in F$ và một mạng các đối tượng C-Object cơ bản.

<u>Ví dụ 4.1</u>: Giả sử có 3 đối tượng O_1, O_2, O_3. Giữa thuộc tính a của O_1, các thuộc tính a và b của O_2, thuộc c của O_3 có một quan hệ f xác định bởi hệ thức:

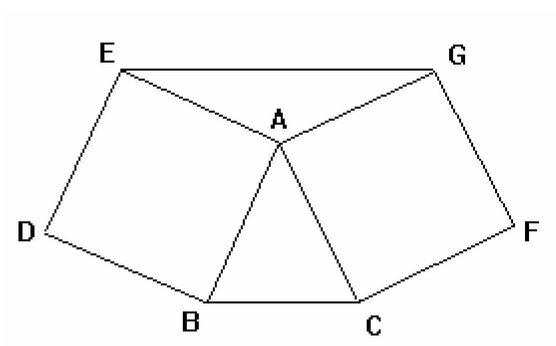
$$O_3.c = (O_1.a)^2 + O_2.a * O_2.b.$$

Ta có hệ thức f xác định một quan hệ tính toán giữa các đối tượng O_1, O_2, O_3.



Hình 4.2 f là một quan hệ tính toán giữa $O_1.a$, $O_2.a$, $O_2.b$, $O_3.c$

<u>Ví dụ 4.2</u>: Cho tam giác cân ABC, cân tại A, và cho biết trước góc đỉnh α, cạnh đáy a. Bên ngoài tam giác có hai hình vuông ABDE và ACFG. Tính độ dài EG.



Hình 4.3 Một bài toán tính toán hình học.

Bài toán có dạng một mạng các đối tượng tính toán bao gồm :

1. Bốn đối tượng :

O_1 : tam giác cân ABC,

O_2 : tam giác AEG,

O_3 : hình vuông ABDE,

O_4 : hình vuông ACFG,

trong đó mỗi tam giác có các biến: a, b, c, GocA, GocB, GocC, h_a, h_b, h_c, S, p, R, r, ... và mỗi hình vuông có các biến: a (cạnh), c (đường chéo), S (diện tích), ...

2. Các quan hệ giữa các đối tượng :

f_1 : $O_1.c = O_3.a$ // cạnh c của tam giác ABC = cạnh của hình vuông ABDE

f_2 : $O_1.b = O_4.a$ // cạnh b của tam giác ABC = cạnh của hình vuông ACFG

f_3 : $O_2.b = O_4.a$ // cạnh b của tam giác AEG = cạnh của hình vuông ACFG

f_4 : $O_2.c = O_3.a$ // cạnh c của tam giác AEG = cạnh của hình vuông ABDE

f_5 : $O_1.GocA + O_2.GocA = \pi$

Trong ví dụ nầy ta có :

$M(f_1) = \{ O_1.c , O_3.a \}$,

$M(f_2) = \{ O_1.b , O_4.a \}$,

$M(f_3) = \{ O_2.b , O_4.a \}$,

$M(f_4) = \{ O_2.c , O_3.a \}$,

$M(f_5) = \{ O_1.GocA , O_2.GocA \}$,

$M = \{ O_1.b, O_1.c, O_1.GocA, O_2.b, O_2.c, O_2.GocA, O_3.a, O_4.a, O_2.a\}$.

Lưu ý rằng $O_2.a$ (cạnh EG của tam giác AEG) là biến cần tính.

4.1.2 Các bài toán trên mạng các C-Object cơ bản

Cho một mạng các C-Object cơ bản (O, M, F). Giả sử có một tập biến $A \subseteq M$ đã được xác định (tức là tập gồm các biến đã biết trước giá trị), và B là một tập biến bất kỳ trong M.

Các vấn đề cơ bản được đặt ra là:

1. Có thể xác định được tập B từ tập A nhờ các quan hệ trong F và các đối tượng thuộc O hay không? Nói cách khác, ta có thể tính được giá trị của các biến thuộc B với giả thiết đã biết giá trị của các biến thuộc A hay không?
2. Nếu có thể xác định được B từ A thì quá trình tính toán giá trị của các biến thuộc B như thế nào?
3. Tìm một lời giải tốt nhất (hay lời giải tối ưu) của bài toán tính toán B từ giả thiết A?

Tương tự như đối với một mạng suy diễn-tính toán, bài toán xác định B từ A trên mạng (O, M, F) được viết dưới dạng:

$$A \rightarrow B$$

trong đó A được gọi là giả thiết, B được gọi là mục tiêu tính toán (hay tập biến cần tính) của bài toán. Trường hợp tập B chỉ gồm có một phần tử b, ta viết vắn tắt bài toán trên là $A \rightarrow b$.

Có thể nhận thấy rằng nếu gộp lại tất cả các biến của các đối tượng O_i (i=1,2,...,n) thành một tập biến lớn và gộp tất cả các quan hệ nội bộ của từng đối tượng cùng với các quan hệ thuộc F thành một tập các quan hệ thì ta có một mạng suy diễn-tính toán như đã xét trong chương 2. Như vậy nếu đặt:

$$\mathscr{M}(O, F) = M(O),$$

$$\mathscr{F}(O, F) = \bigcup_{i=1}^{n} F(O_i) \bigcup F,$$

thì ($\mathscr{M}$, $\mathscr{F}$) là một mạng suy diễn-tính toán; mạng nầy được gọi là mạng suy diễn-tính toán tương ứng của mạng các đối tượng tính toán (O, M, F).

Bài toán $A \rightarrow B$ trên mạng các đối tượng tính toán (O, M, F) được gọi là *giải được* khi bài toán đó là giải được trên ($\mathscr{M}$, $\mathscr{F}$), hay nói cách khác ta có thể tính toán được giá trị các biến thuộc B xuất phát từ giả thiết A. Tất nhiên một lời giải của bài toán trên trên mạng $(\mathscr{M}, \mathscr{F})$ cũng được xem là một lời giải trên mạng các đối tượng. Tuy nhiên lời giải đó có thể có chứa các quan hệ nội bộ bên trong của các đối tượng mà nhiều khi ta không cần quan tâm chi tiết. Do đó ta gọi một lời giải như thế là một *lời giải chi tiết* của bài toán trên mạng các đối tượng tính toán. Chẳng hạn như trong tình huống nêu trong ví dụ sau đây:

Ví dụ 4.3 : Giả sử đang xét bài toán $A \rightarrow B$ trên mạng các đối tượng (O, M, F), và khi giải bài toán trên mạng tính toán $(\mathscr{M}, \mathscr{F})$ tương ứng ta tìm được một lời giải gồm 10 quan hệ (thuộc $\mathscr{F}$) là $\{f_1, f_2, ..., f_{10}\}$, trong đó ta có:

$$\{f_1, f_4, f_7, f_8, f_{10}\} \subseteq F, \qquad \{f_2, f_3\} \subseteq F(O_2),$$

$$\{f_5, f_6\} \subseteq F(O_1), \qquad \{f_9\} \subseteq F(O_2).$$

Theo khái niệm nêu ở trên thì $\{f_1, f_2, ..., f_{10}\}$ là một lời giải chi tiết của bài toán $A \rightarrow B$. Quá trình tính toán theo lời giải nầy có thể được biểu diễn như sau:

$$A = A_0 \xrightarrow{f_1} A_1 \xrightarrow{f_2} A_2 \xrightarrow{f_3} \ . \ . \ . \xrightarrow{f_8} A_8 \xrightarrow{f_9} A_9 \xrightarrow{f_{10}} A_{10}$$

trong đó ta có : $A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_8 \subseteq A_9 \subseteq A_{10} \subseteq \mathscr{M}$,

$A_{10} \supseteq B$.

Trong nhiều trường hợp, đặc biệt là khi tri thức tính toán trên từng đối tượng không cần phải quan tâm chi tiết, ta thay thế mỗi dãy con gồm các quan hệ kế tiếp nhau thuộc cùng một tập hợp các quan hệ $F(O_i)$ trong lời giải chi tiết bởi đối tượng O_i tương ứng. Từ đó ta được một dãy chỉ gồm các quan hệ giữa các thuộc tính của các đối tượng (tức là các quan hệ thuộc F) và các đối tượng; dãy nầy được gọi là một *lời giải gọn (hay vắn tắt là một lời giải)* của bài toán trên mạng các đối tượng tính toán (O, M, F).

Trong ví dụ trên $\{f_1, O_2, f_4, O_1, f_7, f_8, O_2, f_{10}\}$ là một lời giải (gọn) của bài toán $A \rightarrow B$. Quá trình tính toán theo lời giải nầy được biểu diễn như sau :

$$A = A'_0 \xrightarrow{f_1} A'_1 \xrightarrow{O_2} A'_2 \xrightarrow{f_4} \ . \ . \ . \xrightarrow{f_8} A'_6 \xrightarrow{O_2} A'_7 \xrightarrow{f_{10}} A'_8$$

trong đó ta có : $A'_0 \subseteq A'_1 \subseteq A'_2 \subseteq \ldots \subseteq A'_6 \subseteq A'_7 \subseteq A'_8 \subseteq M$,

$A'_8 \supseteq B$.

Tóm lại ta có thể định nghĩa một lời giải như sau:

- **Định nghĩa 4.2**: Một dãy $\{t_1, t_2, ..., t_k\}$ gồm các phần tử thuộc F hay thuộc O được gọi là một *lời giải* của bài toán $A \to B$ trên mạng (O, M, F) nếu như ta lần lượt áp dụng các t_i (i=1,...,k) xuất phát từ giả thiết A thì sẽ suy ra được (hay tính được) các biến thuộc B. Lời giải $\{t_1, t_2, ..., t_k\}$ được gọi là *lời giải không thừa* nếu không thể bỏ bớt một số "bước tính toán" trong quá trình giải, theo nghĩa là không thể bỏ bớt một số quan hệ hay đối tượng trong lời giải.

Việc tìm lời giải cho bài toán là việc tìm ra một dãy các quan hệ hay các đối tượng để có thể áp dụng tính ra được B từ A. Điều nầy cũng có nghĩa là tìm ra được một quá trình tính toán để giải quyết bài toán.

4.2 CÁC THUẬT GIẢI

Trong phần nầy sẽ trình bày một các thuật giải để giải quyết các vấn đề cơ bản đã được nêu trên mà ta gọi là bài toán về tính giải được , bài toán tìm lời giải và bài toán tìm lời giải tối ưu.

4.2.1 Tính giải được của bài toán

Để xét tính giải được của bài toán $A \to B$ trên mạng các đối tượng tính toán (O, M, F), ta có thể khảo sát bài toán trên mạng suy diễn tính toán $(\mathscr{M}, \mathscr{F})$ tương ứng của mạng các đối tượng. Theo cách nầy, ta tìm bao đóng $\tilde{A}$ của A trên mạng $(\mathscr{M}, \mathscr{F})$ rồi xem bao đóng nầy có chứa B không. Tuy nhiên, trong $\tilde{A}$ có thể chứa các biến của các đối tượng mà ta không cần quan tâm; đó là các biến thuộc tập hợp $\tilde{A} \setminus M$. Ở đây, trên mạng các đối tượng (O, M, F), ta chỉ cần quan tâm đến tập hợp biến lớn nhất trong M có thể tính được từ giả thiết A; và có thể thấy rằng tập hợp biến lớn nhất nầy là tồn tại do tính hữu hạn của tập hợp M. Từ đó ta định nghĩa bao đóng của của một tập hợp biến trên mạng các đối tượng tính toán như sau:

- **Định nghĩa 4.3**: Cho (O, M, F) là một mạng các đối tượng tính toán cơ bản, A là một tập hợp con của M. Ta gọi *bao đóng của A trên mạng* là tập hợp lớn nhất trong M gồm các biến có thể suy ra được (hay tính được) từ A, và ký hiệu bao đóng nầy là $\overline{A}$.

Lưu ý rằng bao đóng $\overline{A}$ của A trên mạng các đối tượng tính toán không phải là bao đóng $\tilde{A}$ của A trên mạng tính toán tương ứng. Tuy nhiên ta có thể thấy rằng giữa $\tilde{A}$ và $\overline{A}$ có một sự liên hệ rất tự nhiên được nêu lên trong mệnh đề dưới đây.

- <u>**Mệnh đề 4.1**</u>: Bao đóng $\overline{A}$ của một tập hợp biến A trên một mạng các đối tượng tính toán (O, M, F) bằng phần giao giữa bao đóng $\tilde{A}$ của tập biến đó trong mạng suy diễn tính toán tương ứng và tập biến M được xem xét của mạng các đối tượng, tức là ta có :

$$\overline{A} = \tilde{A} \cap M.$$

- <u>**Định lý 4.1**</u>: Trên một mạng các đối tượng (O, M, F), bài toán $A \to B$ là giải được khi và chỉ khi $B \subseteq \overline{A}$.

Từ định lý nầy, ta có thể kiểm tra tính giải được của bài toán $A \to B$ bằng cách tính bao đóng của tập A rồi xét xem B có bao hàm trong $\overline{A}$ hay không. Để tìm bao đóng của tập hợp $A \subseteq M$, tất nhiên là phải tìm tất cả các biến trong M tính được từ A. Một cách trực quan ta có thể nói rằng việc tìm bao đóng của A là việc mở rộng tối đa tập A trên mạng các đối tượng. Để thực hiện điều nầy, không những phải áp dụng các quan hệ giữa các đối tượng mà ta còn phải áp dụng chính các đối tượng; bởi vì chính bản thân đối tượng có khả năng tính toán thêm được một số thuộc tính nào đó.

Giả sử ta đang có một tập biến được xác định $A \subseteq M$. Đối với một quan hệ giữa các đối tượng $f \in F$, ta có f áp dụng được khi và chỉ khi:

Card $(M(f) \setminus A) \leq r(f)$ nếu f là quan hệ đối xứng,

$M(f) \setminus A \subseteq v(f)$ nếu f là quan hệ không đối xứng.

Trong đó r(f) là số phần tử sẽ được suy ra bởi việc áp dụng quan hệ f và v(f) là tập hợp các phần tử sẽ được suy ra. Việc áp dụng f sẽ mở rộng A thành tập hợp $A \cup M(f)$, được ký hiệu là $\{f\}(A)$ hay ký hiệu vắn tắt là f(A). Như vậy, ta có thể viết : $A \xrightarrow{f} f(A)$. Đối với một đối tượng $O_i \in O$, ta cũng ký hiệu $\{O_i\}(A)$, hoặc $O_i(A)$, là tập hợp biến trong M mở rộng từ A nhờ áp dụng đối tượng O_i. Tập biến $O_i(A)$ có thể được xác định như trong mệnh đề sau đây:

- **Mệnh đề 4.2**: Cho $A \subseteq M$ là một tập hợp biến của mạng các đối tượng (O, M, F), $O_i \in O$. Gọi A' là bao đóng của $A \cap M_i$ trong đối tượng O_i khi xét O_i như một mạng suy diễn tính toán, ta có :

$$O_i(A) = A \cup (A' \cap M).$$

Nhận xét :

(1) Với mọi $t \in F \cup O$ áp dụng được trên A, ta có $t(A) \supseteq A$.

(2) Không giống như các quan hệ giữa các đối tượng, mỗi đối tượng O_i coi như áp dụng được trên một tập biến bất kỳ $A \subseteq M$. Tuy nhiên rất có thể xảy ra trường hợp $O_i(A) = A$, tức là việc áp dụng đối tượng O_i không cho ta thêm thông tin gì mới.

Tương tự như trong chương 2 ta có thể định nghĩa khái niệm về tính "*áp dụng được*" của một dãy D các phần tử trong $F \cup O$ trên một tập biến $A \subseteq M$, và cũng ký hiệu D(A) là tập hợp biến mở rộng từ A nhờ áp dụng dãy D trên mạng. Từ đó, chúng ta có thể kiểm chứng dễ dàng mệnh đề sau:

- **Mệnh đề 4.3** : Trên mạng các đối tượng tính toán (O, M, F) cho $A \subseteq M$, một dãy $D = \{t_1, t_2, ..., t_m\} \subseteq F \cup O$ áp dụng được trên tập hợp A. Đặt :

  $$A_0 = A,\ A_1 = t_1(A_0),\ ...,\ A_m = t_m(A_{m-1}),$$

  ta có :

  $$A_0 \subseteq A_1 \subseteq \ldots \subseteq A_m = D(A) \subseteq M.$$

- **Thuật toán xác định D(A)** :

  1. $A' \leftarrow A$;
  2. **for** $t \in D$ **do**

     if (t áp dụng được trên A') **then** $A' \leftarrow t(A')$;
  3. $D(A) \leftarrow A'$;

- **Định lý 4.2**. Cho một mạng các đối tượng (O, M, F), A và B là hai tập con của M. Ta có các điều sau đây là tương đương:

  (1) $B \subseteq \overline{A}$.

  (2) Có một dãy $D \subseteq F \cup O$ thỏa các điều kiện :

  (a) D áp được trên A.

  (b) $D(A) \supseteq B$.

Cuối cùng, liên quan đến tính giải được của bài toán ta nêu lên thuật toán tìm bao đóng của một tập biến trên mạng các đối tượng tính toán.

- **Thuật toán 4.1**: tìm bao đóng của tập $A \subseteq M$ trên mạng các đối tượng tính toán (O, M, F).

  Nhập : Mạng các đối tượng tính toán (O, M, F), và $A \subseteq M$.

  Xuất : $\overline{A}$

  Thuật toán :

1. A’ ← A;
2. **for** ($t \in F \cup O$) **do**
   if (t áp dụng được trên A) **then** A ← t(A);
3. **if** ($A \neq A'$) **then**
   goto 1;
4. $\overline{A} \leftarrow A'$

Thuật toán trên có thể được viết theo cách khác như sau :

1. **Repeat**
   A’ ← A;
   for ($f \in F$) **do**
     if (f đối xứng **and** Card $(M(f) \setminus A) \leq r(f)$) **or**
       (f không đối xứng **and** $M(f) \setminus A \subseteq v(f)$) **then**
       begin
       A ← t(A); F ← F \ {f};
       end;
   for ($t \in O$) **do**
     if ($t(A) \neq A$) **then**
       A ← t(A);
   Until (A = A’);
2. $\overline{A} \leftarrow A'$;

4.2.2 Tìm lời giải của bài toán

Xét bài toán A → B trên mạng các đối tượng (O, M, F). Trong mục này sẽ nêu lên cách tìm một lời giải cho bài toán theo cách tương tự như đối với một mạng suy diễn tính toán. Theo cách nầy công việc sẽ được tiến hành qua các giai đoạn sau đây :

Giai đoạn 1: Đầu tiên ta tìm một lời giải (có thể có dư thừa bước giải) cho bài toán.

Giai đoạn 2: Xuất phát từ một lời giải đã tìm được, tìm cách trích ra một lời giải không có dư thừa bước giải.

- **<u>Mệnh đề 4.4</u>**: Dãy D gồm các phần tử thuộc $F \cup O$ là một lời giải của bài toán $A \rightarrow B$ khi và chỉ khi D áp dụng được trên A và $D(A) \supseteq B$.

Do mệnh đề trên, để tìm một lời giải ta có thể làm như sau: Xuất phát từ giả thiết A, ta thử áp dụng các quan hệ giữa các đối tượng cùng với các đối tượng để mở rộng dần tập các biến có được xác định cho đến khi đạt đến tập biến B. Tuy nhiên để định hướng nhanh hơn đến mục tiêu, quá trình trên có thể được tiến hành theo thứ tự ưu tiên xem xét như sau :

1/ xét các quan hệ $f \in F$ trước, rồi đến

2/ các đối tượng có chứa yếu tố cần xác định, và cuối cùng là

3/ các đối tượng O_i khác mà tập M_i chưa được xác định hết.

Dưới đây là thuật toán tìm một lời giải cho bài toán $A \rightarrow B$ trên mạng các đối tượng tính toán (O, M, F).

- **<u>Thuật giải 4.2:</u>** tìm một lời giải cho bài toán $A \rightarrow B$:

  <u>Nhập</u> : Mạng các đối tượng tính toán (O, M, F), tập giả thiết $A \subseteq M$, và tập biến cần tính $B \subseteq M$.

  <u>Xuất</u> : lời giải cho bài toán $A \rightarrow B$

  <u>Thuật toán</u> :

  1. Solution ← empty; // Solution *là danh sách các quan hệ*
     // *hay các đối tượng sẽ áp dụng*
  2. **if** $B \subseteq A$ **then**

     begin

```
        Solution_found ← true;
        // biến Solution_found = true khi bài toán là  giải được
        goto  5;
        end
    else
        Solution_found ← false;
3.  Repeat
        Aold ← A;
        Chọn ra một  f ∈  F chưa xem xét (trong bước 3 lần nầy);
        while  not Solution_found  and (chọn được f)  do
            begin
            if  ( f đối xứng  and  0 < Card (M(f) \ A) ≤  r(f) )  or
                ( f không đối xứng  and  ∅ ≠ M(f) \ A ⊆ v(f) )   then
                begin
                A ← f(A);
                Solution ← Solution ∪ { f }; // thêm f vào Solution
                end;
            if  B ⊆ A  then
                Solution_found ← true;
            Chọn ra một  f ∈  F chưa xem xét (trong bước 3 lần nầy);
            end;   { while }
    Until   Solution_found  or  (A = Aold);
4.  if  not Solution_found  then
        begin
        Chọn ra một  Oi ∈  O (theo thứ tự ưu tiên đã nói ở trên) sao cho
```

```
            Oi(A) ≠ A;
        if  (chọn được Oi)  then
            begin
            A ← Oi(A);   Solution ← Solution ∪ { Oi };
            if  (B ⊆ A)  then     begin
                Solution_found ← true;
                goto  5;
                end;
            else
                goto  3;
            end;
        end;
    5.  if  not Solution_found  then
            Bài toán không có lời giải;
        else
            Begin
            5.1: Lần ngược theo các bước giải trong <Solution> để xem xét
                 loại bỏ những bước giải dư thừa
            5.2: <Solution> là một lời giải.
            End
```

<u>Ghi chú</u> : Về sau, khi cần trình bày quá trình giải (hay bài giải) ta có thể xuất phát từ lời giải tìm được dưới dạng một dãy $D \subseteq F \cup O$ để xây dựng bài giải. Các bước 1-4 trong thuật toán là giai đoạn tìm một lời giải (có thể có sự dư thừa bước giải). Trong trường hợp bài toán có lời giải, bước 5.1 là giai đoạn loại bỏ các bước giải dư thừa trong lời giải.

Ví dụ 4.4 : Bây giờ chúng ta khảo sát bài toán đã nêu trong ví dụ 4.2.

Cho tam giác cân ABC, cân tại A, và cho biết trước góc A, cạnh đáy a. Bên ngoài tam giác có hai hình vuông ABDE và ACFG. Tính độ dài EG.

Như đã trình bày trong ví dụ 4.2, bài toán có thể được biểu diễn trong một mạng các đối tượng (O, M, F) bởi GT $\Rightarrow$ KL với

$O = \{O_1, O_2, O_3, O_4\}$

$M = \{ O_1.a, O_1.b, O_1.c, O_1.GocA, O_2.b, O_2.c, O_2.GocA , O_2.a, O_3.a, O_4.a \}$

$F = \{f_1, f_2, f_3, f_4, f_5\}$

$GT = \{ O_1.a , O_1.GocA \}$

$KL = \{ O_2.a \}$

Trong đó đối với các quan hệ $f_i \in F$ ta có:

$M(f_1) = \{ O_1.c , O_3.a \}$,

$M(f_2) = \{ O_1.b , O_4.a \}$,

$M(f_3) = \{ O_2.b , O_4.a \}$,

$M(f_4) = \{ O_2.c , O_3.a \}$,

$M(f_5) = \{ O_1.GocA , O_2.GocA \}$.

Thuật toán tìm lời giải ở trên sẽ cho ta một lời giải cho bài toán là dãy:

$D = \{ f_5, O_1, f_1, f_2, f_3, f_4, O_2\}$,

và quá trình mở rộng tập biến được xác định theo lời giải nầy như sau:

$$GT = A_0 \xrightarrow{f_5} A_1 \xrightarrow{O_1} A_2 \xrightarrow{f_1} A_3 \xrightarrow{f_2} A_4$$
$$\xrightarrow{f_3} A_5 \xrightarrow{f_4} A_6 \xrightarrow{O_2} A_7$$

trong đó :

$A_0 = A = \{O_1.a , O_1.GocA\}$,

$A_1 = \{O_1.a , O_1.GocA, O_2. GocA \}$,

$A_2 = \{ O_1.a , O_1.GocA, O_2. GocA, O_1.b, O_1.c \}$,

$A_3 = \{O_1.a\ , O_1.GocA, O_2.\ GocA, O_1.b, O_1.c, O_3.a\}$,

$A_4 = \{O_1.a\ , O_1.GocA, O_2.\ GocA, O_1.b, O_1.c, O_3.a, O_4.a\}$,

$A_5 = \{O_1.a\ , O_1.GocA, O_2.\ GocA, O_1.b, O_1.c, O_3.a, O_4.a, O_2.b\}$,

$A_6 = \{O_1.a\ , O_1.GocA, O_2.\ GocA, O_1.b, O_1.c, O_3.a, O_4.a, O_2.b, O_2.c\}$,

$A_7 = \{O_1.a\ , O_1.GocA, O_2.\ GocA, O_1.b, O_1.c, O_3.a, O_4.a, O_2.b, O_2.c, O_2.a\}$.

4.2.3 Định lý về sự phân tích quá trình giải

Cũng như đối với mạng suy diễn tính toán, ta cũng cần xem xét quá trình áp dụng các quan hệ giữa các đối tượng, và các đối tượng trong một lời giải cho một bài toán trên mạng các đối tượng (O, M, F). Từ đó thiết lập quá trình tính toán các biến dựa theo lời giải. Dưới đây là một định lý tương tự như định lý 2.4 trong chương 2.

- **Định lý 4.3**. Cho $\{t_1, t_2, ..., t_m\}$ là một lời giải tốt cho bài toán $A \rightarrow B$ trên một mạng các đối tượng tính toán (O, M, F). Đặt :

  $A_0 = A$, $A_i = \{t_1, t_2, ..., t_i\}(A)$, với mọi i=1,...,m.

  Khi đó có một dãy $\{B_0, B_1, ..., B_{m-1}, B_m\}$ các tập con của M, thỏa các điều kiện sau đây:

  (1) $B_m = B$.

  (2) $B_i \subseteq A_i$, với mọi i=0,1,...,m.

  (3) Với mọi i=1,...,m, $\{t_i\}$ là lời giải của bài toán $B_{i-1} \rightarrow B_i$ nhưng không phải là lời giải của bài toán $G \rightarrow B_i$, trong đó G là một tập con thật sự tùy ý của B_{i-1}.

Tuy nhiên, khi xây dựng dãy $\{B_0, B_1, ..., B_{m-1}, B_m\}$ ta cần lưu ý đến trường hợp t_i là một đối tượng thì cách xây dựng không giống như đối với trường hợp t_i là một quan hệ. Chẳng hạn, nếu t_m là một đối tượng O_i thì B_{m-1} được xác định như sau :

$$B_{m-1} = (B_m \cap A_{m-1}) \cup A'_{m-1},$$

trong đó A'_{m-1} là một tập hợp con của $(M_i \cap A_{m-1}) \setminus (B_m \cap A_{m-1})$ có ít phần tử nhất sao cho $\{O_i\}$ là lời giải của bài toán $B_{m-1} \rightarrow B_m$.

4.2.4 Lời giải tối ưu

Trong mục nầy chúng ta định nghĩa một lời giải tối ưu của một bài toán A $\rightarrow$ B trên mạng các đối tượng (O, M, F) dựa trên các trọng số tính toán của các quan hệ tính toán trên mạng.

- **Định nghĩa 4.4**: Cho (O, M, F) là một mạng các đối tượng tính toán cơ bản. Giả sử rằng ứng với mỗi f $\in$ F ta có một trọng số dương tương ứng c(f) và với mỗi bài toán có dạng GT $\rightarrow$ KL trong một đối tượng O_i ta cũng có một trọng số dương tương ứng c'(O_i, GT, KL) khi bài toán là giải được. Khi đó ta nói rằng mạng các đối tượng là mạng có trọng số và ký hiệu là (O, M, F, c, c').
- **Định nghĩa 4.5**: Xét một bài toán A $\rightarrow$ B trên một mạng có trọng số (O, M, F, c, c'), với A và B là các tập hợp con của M. Giả sử S = $\{t_1, t_2, ..., t_k\}$ là một lời giải của bài toán, ứng với mỗi $t_i \in$ S ta đặt:

  $w(t_i) = c(t_i)$ nếu $t_i \in F$,

  $w(t_i) = c'(t_i, GT, KL)$ nếu $t_i \in O$ và trong đó

  $GT = \{t_1, t_2, ..., t_{i-1}\}(A) \cap M(t_i) \cap M$

  KL = bao đóng của GT trong đối tượng t_i.

  $$w(S) = \sum_{i=1}^{m} w(t_i)$$

  Ta gọi w(S) là trọng số của lời giải S, và S được gọi là một *lời giải tối ưu* của bài toán A $\rightarrow$ B khi và chỉ khi lời giải S có trọng số nhỏ nhất, tức là khi

  $$w(S) = \min \{w(S') : S' \text{ là một lời giải của bài toán } A \rightarrow B\}$$

Ta có thể tìm một lời giải tối ưu cho bài toán A $\rightarrow$ B trên mạng (O, M, F, c, c') bằng cách xây dựng một mạng suy diễn tính toán có trọng số (Attrs, D, w) tương ứng sao cho việc tìm lời giải tối ưu trên mạng các đối tượng tương đương với việc tìm một lời giải tối ưu trên mạng (Attrs, D, w). Mạng suy diễn tính toán nầy được xây dựng như sau:

(1) Attrs = M.

(2) D = F $\cup$ {(ob, A', B') : ob $\in$ O, A'$\subset$ M, B'$\subset$ M, A'$\rightarrow$ B' giải được trong ob}

(3) Với mỗi t $\in$ D, w(t) = c(t) nếu t $\in$ F và w(t) = c'(t) nếu t $\notin$ F.

Mạng suy diễn tính toán xây dựng như trên thỏa mãn điều ta mong muốn. Điều nầy được phát biểu trong mệnh đề sau đây:

- **Mệnh đề 4.5**. Mỗi lời giải tối ưu của bài toán A $\rightarrow$ B trên mạng (O, M, F, c, c') thì tương ứng với một lời giải tối ưu trên mạng (Attrs, D, w) và ngược lại.

4.3 MẠNG CÁC C-OBJECT TỔNG QUÁT

Trong phần nầy chúng ta sẽ khảo sát một mạng các C-Object tổng quát xét trong một cơ sở tri thức các đối tượng mà ta gọi là các C-Object (xem chương 3), trong đó ngoài những quan hệ tính toán còn có nhiều luật khác trên các loại sự kiện khác nhau. Ở đây không phải chỉ có các loại sự kiện tính toán được quan tâm khảo sát mà ta còn đề cập đến những sự kiện quan hệ khác trên các thuộc tính và trên các đối tượng. Mô hình mạng các C-Object tổng quát sẽ cho ta một phương pháp biểu diễn các dạng bài toán tổng quát trong hệ cơ sở tri thức các C-Object, làm cơ sở cho việc thiết kế các môđun giải toán và trợ giúp giải toán trong các hệ giải toán thông minh.

4.3.1 Mô hình mạng các C-Object tổng quát

Mô hình mạng các C-Object tổng quát được được xét trong một cơ sở tri thức các C-Object được trình bày trong chương 3. Mô hình nầy được dùng để biểu diễn cho một dạng bài toán tổng quát trong cơ sở tri thức mà ta sẽ xem xét trong phần nầy.

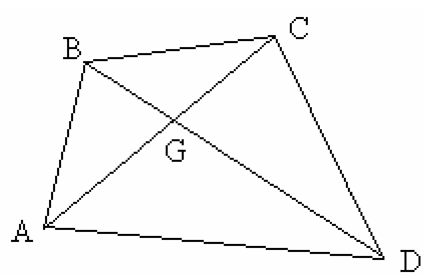
- **Định nghĩa 4.6**: Giả sử ta có một mô hình (cơ sở) tri thức các C-Object COKB = (C, H, R, Ops, Rules). Ta gọi một mạng các C-Object trong mô hình COKB, viết vắn tắt bởi CO-Net, là một bộ (O, F) với:
  (1) O là một tập hợp các C-Object (hay các đối tượng), mỗi đối tượng có một tên cụ thể và thuộc một khái niệm được biết trong COKB.
  (2) F là một tập hợp sự kiện, mỗi sự kiện thể hiện một tính chất hay một liên hệ nào đó trên các đối tượng hay trên các thuộc tính của các đối tượng.

Đối với một CO-Net (O, F), khi ta phải xem xét một tập sự kiện mục tiêu G và muốn khảo sát những vấn đề suy diễn và tính toán (hay giải toán) các sự kiện trong G từ mạng thì ta nói rằng ta có một bài toán trên CO-Net. Bài toán nầy sẽ được ký hiệu là:

$$(O, F) \Rightarrow G$$

Ví dụ 4.5: Các ví dụ về các CO-Net và các bài toán.

1 - Mạng các đối tượng (O, F) trong một tứ giác ABCD với G là giao điểm của 2 đường chéo:



Hình 4.4 Một tứ giác ABCD

O = {O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8, O_9}

Trong đó các đối tượng O_1, O_2, ..., O_9 lần lượt là tứ giác ABCD, các tam giác ABD, CBD, BAC, DAC, GAB, GBC, GCD, và tam giác GDA (được xây dựng trên các điểm A, B, C, D, G).

F = { O_2.a = O_1.BD, O_2.b = O_1.d, O_2.c = O_1.a, O_2.GocA = O_1.GocA,

O_3.a = O_1.BD, O_3.b = O_1.c, O_3.c = O_1.b, O_3.GocA = O_1.GocC,

O_4.a = O_1.AC, O_4.b = O_1.b, O_4.c = O_1.a, O_4.GocA = O_1.GocB,

O_5.a = O_1.AC, O_5.b = O_1.c, O_5.c = O_1.d, O_5.GocA = O_1.GocD,

O_1.GocA = O_4.GocB + O_5.GocB, O_1. GocB = O_2. GocB + O_3. GocB,

O_1.GocC = O_4.GocC + O_5.GocC, O_1.GocD = O_2.GocC + O_3.GocC,

O_1.S = O_2.S + O_3.S, O_1.S = O_4.S + O_5.S,

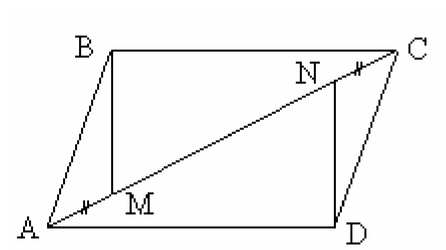
G $\in$ DOAN[B,D], G $\in$ DOAN[A,C],

O_6.GocA = O_7.GocA, O_8.GocA = O_9.GocA,

O_6.GocA = π - O_8.GocA, O_7.GocA = π - O_9.GocA,

. . . }

2 - Xét bài toán: Cho hình bình hành ABCD. Giả sử M và N là 2 điểm trên AC sao cho AM = CN. Chứng minh rằng tam giác ABM bằng tam giác CDN.



Hình 4.5 Một bài toán chứng minh tam giác bằng nhau

Bài toán nầy có thể biểu diễn dưới dạng bài toán (O, F) $\Rightarrow$ G như sau:

O = { O_1, O_2, O_3}

Trong đó O_1 là hình bình hành ABCD, O_2 là tam giác ABM và O_3 là tam giác CDN (được xây dựng trên các điểm A, B, C, D, M và N).

F = { O_2.b = O_3.b (cạnh AM = cạnh CN),

M $\in$ AC, N $\in$ AC,

O_2.c = O_1.a (cạnh AB = cạnh AB),

$O_3.c = O_1.a$ (cạnh CD = cạnh CD)

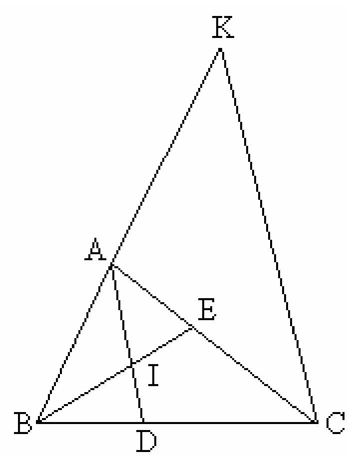
}

$G = \{ O_2 = O_3 \}$

3 - Xét bài toán: Cho tam giác ABC và 2 đường phân giác trong của 2 góc A và B là AD và BE (D nằm trên cạnh BC và E nằm trên cạnh AC). Gọi I là giao điểm của AD và BE. Từ C dựng đường thẳng song song với AD cắt đường AB ở K.

(a) Chứng minh rằng trong tam giác ACK có góc C bằng góc K.

(b) Tính Góc AIB nếu biết góc ACB = 24^o.



Hình 4.6

Bài toán nầy có thể biểu diễn dưới dạng các bài toán $(O, F) \Rightarrow G$ và $(O, F') \Rightarrow G'$ như sau:

$O = \{ O_1, O_2\}$

Trong đó O_1 là tam giác ABC, O_2 là tam giác AKC (được xây dựng trên các điểm A, B, C, D, E và K).

F = { Góc BAD = Góc DAC,

Góc ABE = Góc CBE,

A ∈ Đoạn BK, Đoạn AD // Đoạn CK

}

G = { Góc AKC = Góc ACK }

F’ = F ∪ { Góc ACB = 24^o }

G’ = { Góc AIB}

4.3.2 Phương pháp giải tự động

Chiến lược suy diễn cơ bản được sử dụng ở đây là phương pháp suy diễn tiến/lùi kết hợp với một số qui tắc heuristic. Tuy nhiên ở mỗi bước giải ta không chỉ áp dụng các luật suy diễn mà còn thực hiện các tính toán thích hợp và áp dụng các đối tượng. Các đối tượng tham gia vào bước giải như một tác nhân có khả năng thực hiện các hành vi nhất định để phát sinh sự kiện mới. Trong quá trình thực hiện việc suy luận ta phải xem xét sự hợp nhất giữa các sự kiện tương tự như đã trình bày trong phương pháp giải toán một C-Object (chương 3). Sự hợp nhất của các sự kiện dựa trên sự phân loại của các sự kiện làm 6 loại. Ngoài ra, ta cũng sử dụng các dạng suy luận khác nhau có thể phát sinh được các sự kiện mới từ các sự kiện đã biết.

- Các dạng suy luận nầy bao gồm:
  - (1) Suy ra các sự kiện mặc nhiên từ các cấu trúc của sự kiện và các đối tượng (tự phát sinh sự kiện mà không cần dùng luật hay qui tắc tính toán nào).
  - (2) Áp dụng các luật để xác định đối tượng hay thuộc tính.
  - (3) Áp dụng các luật để nhận dạng đối tượng.
  - (4) Luật suy diễn trên các sự kiện không có sự tính toán hay thay thế ra biểu thức mới. Luật có dạng: {Các sự kiện} ⇒ { Các sự kiện }. Trong đó ta

có thể chú ý đến các trường hợp sau đây khi xem xét sự liên hệ giữa các sự kiện thể hiện một quan hệ và các sự kiện thể hiện một sự tính toán:

(a) {Các sự kiện quan hệ } ⇒ { Các sự kiện quan hệ}

(b) {Các sự kiện quan hệ } ⇒ { Các sự kiện tính toán}

(c) {Các sự kiện tính toán } ⇒ { Các sự kiện quan hệ}

(d) {Các sự kiện tính toán } ⇒ { Các sự kiện tính toán}

(5) Các luật suy diễn đơn giản trên các sự kiện như các luật "Deduce" trong phương pháp giải một C-Object.

(6) Các luật thay thế, tính toán để suy ra sự kiện mới (dưới dạng một biểu thức mới).

(7) Áp một đối tượng.

(8) Giải một hệ phương trình.

(9) Thiết lập một đối tượng mới.

Thuật giải cơ bản cho quá trình suy diễn tiến như sau:

- **Thuật giải 4.3**

Bước 1: Ghi nhận mô hình của bài toán bao gồm các đối tượng, các sự kiện đã có và các sự kiện mục tiêu.

Bước 2: Thực hiện việc phát sinh các sự kiện mặc nhiên theo dạng suy luận (1), và ghi nhận thêm sự kiện mới (nếu có) cùng với một bước giải đã được thực hiện.

Bước 3: Kiểm tra mục tiêu, nếu đạt được thì chuyển sang bước 7.

Bước 4: Tìm một luật hay một dạng suy luận để phát sinh một số sự kiện mới hay một số đối tượng mới.

Bước 5: Nếu tìm được luật hay dạng suy luận ở bước 4 thì ghi nhận các thông tin của một bước giải: Ghi nhận đối tượng mới vào tập hợp các đối tượng, ghi nhận các sự kiện mới vào tập các sự kiện đã biết, và quay lại bước 3.

Bước 6: Nếu không tìm thấy được luật hay dạng suy luận ở bước 4 thì kết thúc với kết quả là không tìm thấy lời giải cho bài toán.

Bước 7: Rút gọn lời giải tìm được bằng cách loại bỏ dư thừa và phân tích quá trình giải để xác định các sự kiện mới cần thiết trong mỗi bước giải.

Đối với bài toán trong ví dụ 4.2, thì thuật giải trên sẽ dễ dàng cho ta một lời giải khá tốt, chẳng hạn như một lời giải gồm các bước giải sau:

Bước giải 1: Xác định góc A của tam giác AEG.

Bước giải 2: Trong tam giác cân ABC, xác định cạnh AB.

Bước giải 3: Xác định cạnh của hình vuông ABDE.

Bước giải 4: Xác định cạnh của hình vuông ACFG.

Bước giải 5: Xác định cạnh AE của tam giác AEG.

Bước giải 6: Xác định cạnh AG của tam giác AEG.

Bước giải 7: Trong tam giác cân AEG, xác định cạnh EG.

Nhận xét:

1. Do số khái niệm về các C-Object là hữu hạn, nên nếu các đối tượng phát sinh trong quá trình suy luận chỉ thiết lập trên một số lượng bị chặn các đối tượng cơ sở thì số đối tượng của bài toán trong quá trình suy luận là hữu hạn và bị chặn, và do đó số sự kiện loại 1 của bài toán cũng hữu hạn và bị chặn.
2. Khi số đối tượng của bài toán là hữu hạn bị chặn trong quá trình suy luận thì số sự kiện loại 2, 3, 4 và 6 cũng hữu hạn bị chặn trong quá trình suy luận.
3. Nếu các qui tắc hay luật phát sinh sự kiện mới loại 5 đều thuộc các loại đơn giản là thay thế hay giải phương trình để dẫn tới các sự kiện mới với số biến

liên hệ trong sự kiện là thấp hơn so với sự kiện có trước, thì số sự kiện loại 5 cũng là hữu hạn và bị chặn.

Từ những nhận xét trên ta có tính chất sau đây:

Trong quá trình suy luận, nếu các đối tượng phát sinh chỉ thiết lập trên một số lượng bị chặn các đối tượng cơ sở như trong nhận xét 1 và nếu các dạng suy luận phát sinh sự kiện loại 5 thuộc loại đơn giản như trong nhận xét 3 thì thuật giải sẽ dừng sau một số hữu hạn bước.

Thuật giải cơ bản trên sẽ được tăng cường tính hiệu quả khi ta sử dụng các heuristics. Quá trình suy diễn với heuristics có dạng thuật giải như sau:

- **<u>Thuật giải 4.4:</u>**

Bước 1: Ghi nhận mô hình của bài toán bao gồm các đối tượng, các sự kiện đã có và các sự kiện mục tiêu.

Bước 2: Thực hiện việc phát sinh các sự kiện mặc nhiên theo dạng suy luận (1), và ghi nhận thêm sự kiện mới (nếu có) cùng với một bước giải đã được thực hiện.

Bước 3: Kiểm tra mục tiêu, nếu đạt được thì chuyển sang bước 8.

Bước 4: Sử dụng các heuristic để chọn luật hay dạng suy luận thích hợp để phát sinh sự kiện mới.

Bước 5: Nếu sự chọn lựa ở bước 4 không thành công thì ta tìm một luật hay một dạng suy luận bất kỳ có thể được để phát sinh một số sự kiện mới hay một số đối tượng mới.

Bước 6: Nếu tìm được luật hay dạng suy luận ở bước 4 hay bước 5 thì ghi nhận các thông tin của một bước giải: Ghi nhận đối tượng mới vào tập hợp các đối tượng, ghi nhận các sự kiện mới vào tập các sự kiện đã biết, và quay lại bước 3.

Bước 7: Nếu không tìm thấy được luật hay dạng suy luận ở bước 4 hay bước 5 thì kết thúc với kết quả là không tìm thấy lời giải cho bài toán.

Bước 8: Rút gọn lời giải tìm được bằng cách loại bỏ dư thừa và phân tích quá trình giải để xác định các sự kiện mới cần thiết trong mỗi bước giải.

Việc áp dụng các qui tắc heuristic là rất quan trọng trong sự suy luận. Các heuristic giúp ta có thể tìm được lời giải nhanh chóng hơn và cho một lời giải rất tự nhiên như sự suy nghĩ và cho lời giải của con người. Dưới đây là một số heuristic có thể được sử dụng:

(H1) Ưu tiên sử dụng các qui tắc xác định đối tượng và các thuộc tính của đối tượng.

(H2) Chuyển đổi đối tượng (nhận dạng đối tượng thuộc khái niệm mức cao hơn) sang khái niệm mức cao hơn trong biểu đồ phân cấp các khái niệm.

(H3) Sử dụng các qui tắc phát sinh đối tượng mới để liên kết các yếu tố trên mạng các đối tượng.

(H4) Khi phát sinh đối tượng thì ưu tiên tạo ra đối tượng có liên quan đến các đối tượng đang có nhất là liên quan đến các sự kiện mục tiêu.

(H5) Ưu tiên sử dụng luật hay dạng suy luận để phát sinh ra sự kiện liên quan đến các sự kiện mục tiêu.

(H6) Nếu không thể phát sinh sự kiện mới hay các đối tượng mới ta có thể đặt tham biến và giải các phương trình hay hệ phương trình.

(H7) Luôn luôn có sự kiện mới khi thiết lập đối tượng mới.

Dưới đây là sự minh họa cho việc sử dụng các heuristics trong suy luận giải bài toán được nêu trong ví dụ 4.5-2. Theo đề bài, đầu tiên ta có 4 đối tượng O_1 (hình bình hành ABCD), O_2 (tam giác ABM), O_3 (tam giác CDN) và “đoạn AC”. Các sự kiện ban đầu gồm:

$O_2.b = O_3.b$ (cạnh AM = cạnh CN),

$M \in AC$, $N \in AC$,

$O_2.c = O_1.a$ (cạnh AB = cạnh AB),

$O_3.c = O_1.a$ (cạnh CD = cạnh CD)

Thực hiện suy luận trên các sự kiện thông thường liên quan đến các thuộc tính của hình bình hành ABCD sẽ không dẫn đến được kết luận vì chưa liên hệ được thêm các yếu tố khác giữa tam giác ABM và tam giác CDN. Bởi việc sử dụng các heuristic 4 và 7 sẽ dẫn đến việc phát sinh các đối tượng mới là O_4 (tam giác ABC) và O_5 (tam giác CDA) với những sự kiện mới liên quan:

$O_4 = O_5$ (sự bằng nhau của 2 tam giác),

$O_4.GocA = O_2.GocA$ (sự bằng nhau của 2 góc thứ nhất),

$O_5.GocA = O_3.GocA$ (sự bằng nhau của 2 góc thứ nhất),

$O_4.GocB = O_1.GocB$, . . .

Từ đó ta có đủ các sự kiện liên quan đến các thuộc tính của O_2 và O_3 :

$O_2.b = O_3.b$,

$O_2.c = O_3.c$,

$O_2.GocA = O_3.GocA$

và có thể suy ra được

$O_2 = O_3$

theo một trường hợp bằng nhau của 2 tam giác (c.g.c).

Ghi chú: Đối với bài toán nầy ngoài cách phát sinh đối tượng mới ta có thể sử dụng các luật và tính năng mở rộng của đối tượng bằng cách trang bị thêm cho đối tượng một số luật trên các đối tượng thiết lập trên danh sách các đối tượng nền của nó và một mạng đối tượng bên trong để tăng cường sức mạnh suy luận và tính toán của đối tượng. Điều nầy sẽ được trình bày chi tiết hơn trong việc thiết kế mở rộng cho tính năng giải toán của một tứ giác trong phần kế tiếp.

4.4 TỨ GIÁC VỚI TÍNH NĂNG MỞ RỘNG

Trong phần nầy chúng ta sẽ mở rộng khả năng giải toán của một C-Object thông qua việc bổ sung các luật nội bộ liên quan đến các đối tượng thiết lập trên danh sách các đối tượng nền của tứ giác: 4 điểm ở 4 đỉnh của tứ giác. Một mạng đối tượng nội bộ cũng được đưa vào để liên kết các thuộc tính cũng như các đối tượng liên quan trong tứ giác. Kỹ thuật nầy làm cho đối tượng “tứ giác” có khả năng xử lý và giải quyết nhiều bài toán hơn so với phương pháp giải C-Object đã được trình bày trước đây trong chương 3.

4.4.1 Cấu trúc của đối tượng tứ giác

Xét một tứ giác ABCD với danh sách đối tượng nền gồm 4 điểm A, B, C và D. Cấu trúc cơ bản của tứ giác gồm:

(1) Một tập hợp Attrs các thuộc tính:

Attrs = { a, b, c, d, c1, c2, GA, GB, GC, GD, S, p}

Trong đó a, b, c, d lần lượt là các cạnh AB, BC, CD, DA; c1 và c2 lần lượt là các đường chéo AC và BD; GA, GB, GC, GD lần lượt là các góc A, B, C, D của tứ giác; S là diện tích của tứ giác và p là chu vi của tứ giác.

(2) Một tập hợp F gồm các quan hệ tính toán cơ bản:

F = { GA + GB + GC + GD = 2*Pi, a+b+c+d = p,

```
2*S = a*d*sin(GA)+ b*c*sin(GC),
2*S = a*b*sin(GB)+ c*d*sin(GD), ...}
```

(3) Một tập hợp rỗng Facts các sự kiện.

(4) Một tập hợp Rules gồm các luật liên hệ trên các sự kiện liên quan đến các thuộc tính của đối tượng, đến bản thân đối tượng cùng với một số sự kiện liên quan đến một số đối tượng thiết lập trên các đối tượng nền:

Rules = { { ["SSONG", a, c] } ⇒ {GD=Pi-GA, GB=Pi-GC,
GOC[A,B,D]=GOC[C,D,B], GOC[C,A,B]=GOC[A,C,D] },
{GOC[C,A,B]=GOC[A,C,D] } ⇒ { ["SSONG", a, c] },
{a=c, b=d} ⇒ { ["SSONG", a, c], ["SSONG", b, d] },
...}

Với cấu trúc như thế, các thuật giải để giải toán một C-Object được trình bày trong chương 3 sẽ trang bị cho đối tượng khả năng giải toán trên đối tượng. Tuy nhiên có nhiều bài toán trên đối tượng sẽ không giải được nếu không trang bị thêm cho đối tượng những sự kiện liên hệ trên các đối tượng thiết lập trên nền của các đối tượng cơ bản. Có thể lấy ví dụ như tính diện tích của tứ giác khi biết 4 cạnh và 1 góc. Vì thế chúng ta sẽ thiết kế một mạng các đối tượng trong tứ giác để có thể mở rộng khả năng giải bài toán của đối tượng.

4.4.2 Mạng các đối tượng trong tứ giác

Như đã nêu lên trong ví dụ 4.5-1 ở trên, một tứ giác ABCD có thể được xem xét cùng với một số các đối tượng tam giác liên quan để tạo thành một mạng đối tượng (O, F) với:

$O = \{O_1, O_2, O_3, O_4, O_5, O_6, O_7, O_8, O_9\}$

Trong đó các đối tượng $O_1, O_2, ..., O_9$ lần lượt là tứ giác ABCD, các tam giác ABD, CBD, BAC, DAC, GAB, GBC, GCD, và tam giác GDA (được xây dựng trên các điểm A, B, C, D, G).

$F = \{ O_2.a = O_1.BD, O_2.b = O_1.d, O_2.c = O_1.a, O_2.GocA = O_1.GocA,$

$O_3.a = O_1.BD, O_3.b = O_1.c, O_3.c = O_1.b, O_3.GocA = O_1.GocC,$

O_4.a = O_1.AC, O_4.b = O_1.b, O_4.c = O_1.a, O_4.GocA = O_1.GocB,

O_5.a = O_1.AC, O_5.b = O_1.c, O_5.c = O_1.d, O_5.GocA = O_1.GocD,

O_1.GocA = O_4.GocB + O_5.GocB, O_1. GocB = O_2. GocB + O_3. GocB,

O_1.GocC = O_4.GocC + O_5.GocC, O_1.GocD = O_2.GocC + O_3.GocC,

O_1.S = O_2.S + O_3.S, O_1.S = O_4.S + O_5.S,

G $\in$ DOAN[B,D], G $\in$ DOAN[A,C],

O_6.GocA = O_7.GocA, O_8.GocA = O_9.GocA,

O_6.GocA = π - O_8.GocA, O_7.GocA = π - O_9.GocA,

. . . }

Như thế khi áp dụng phương pháp giải bài toán trên mạng các C-Object nầy để trang bị khả năng giải toán cho đối tượng tứ giác thì nó có thể giải được nhiều bài toán hơn. Ngoài ra ta có thể xem xét một mạng các đối tượng nhỏ hơn để giảm bớt những bước "suy nghĩ" không cần thiết. Dưới đây chúng ta xét hai ví dụ để minh họa.

Ví dụ 4.6 : Trong một tứ giác ABCD, cho biết 4 cạnh AB, BC, CD, DA, và góc A. Hãy tính diện tích S của tứ giác.

Theo đề bài ta có các sự kiện giả thiết là :

{O_1.a, O_1.b, O_1.c, O_1.d, O_1.GA},

mục tiêu cần tính toán là :

{ S }.

Đặt tứ giác O_1 (tứ giác ABCD) trong mạng tính toán liên hệ với 4 đối tượng tam giác :

O_2 : tam giác ABD,

O_3 : tam giác CBD,

O_4 : tam giác BAC,

O_5 : tam giác DAC.

Ta có mạng tính toán gồm 5 đối tượng O_1, O_2, O_3, O_4, O_5. Trong O_1 ta có 4 quan hệ $O_1.f_1$, $O_1.f_2$, $O_1.f_3$, $O_1.f_4$. Về mối liên hệ giữa các đối tượng trên ta có các quan hệ sau đây :

f_1 : $O_2.a = O_1.BD$

f_2 : $O_2.b = O_1.d$

f_3 : $O_2.c = O_1.a$

f_4 : $O_2.GocA = O_1.GA$

f_5 : $O_3.a = O_1.BD$

f_6 : $O_3.b = O_1.c$

f_7 : $O_3.c = O_1.b$

f_8 : $O_3.GocA = O_1.GC$

f_9 : $O_4.a = O_1.AC$

f_{10} : $O_4.b = O_1.b$

f_{11} : $O_4.c = O_1.a$

f_{12} : $O_4.GocA = O_1.GB$

f_{13} : $O_5.a = O_1.AC$

f_{14} : $O_5.b = O_1.c$

f_{15} : $O_5.c = O_1.d$

f_{16} : $O_5.GocA = O_1.GD$

f_{17} : $O_1.GA = O_4.GocB + O_5.GocB$

f_{18} : $O_1.GB = O_2.GocB + O_3.GocB$

f_{19} : $O_1.GC = O_4.GocC + O_5.GocC$

f_{20} : $O_1.GD = O_2.GocC + O_3.GocC$

f_{21} : $O_1.S = O_2.S + O_3.S$

f_{22} : $O_1.S = O_4.S + O_5.S$

Như thế trong mô hình mạng các đối tượng cơ bản của bài toán đặt ra ta có :

1/ tập các đối tượng :

$O = \{ O_1, O_2, O_3, O_4, O_5\}$.

2/ tập các quan hệ (giữa các đối tượng) :

$F = \{ f_1, f_2, \ldots, f_{21}, f_{22}\}$.

3/ tập các biến được xem xét :

M = { O_1.a, O_1.b, O_1.c, O_1.d, O_1.GA, O_1.GB, O_1.C, O_1.D, O_1.S, O_1.BD, O_1.AC,

O_2.a, O_2.b, O_2.c, O_2.GocA, O_2.GocB, O_2.GocC, O_2.S,

O_3.a, O_3.b, O_3.c, O_3.GocA, O_3.GocB, O_3.GocC, O_3.S,

O_4.a, O_4.b, O_4.c, O_4.GocA, O_4. GocB, O_4.GocC, O_4.S,

O_5.a, O_5.b, O_5.c, O_5.GocA, O_5.GocB, O_5. GocC, O_5.S

}

4/ Giả thiết (các sự kiện đã biết):

A = { O_1.a, O_1.b, O_1.c, O_1.d, O_1.GA }

5/ Mục tiêu tính toán (tập biến cần tính) :

B = { O_1.S }

Giải theo thuật giải suy diễn tiến ta có quá trình xem xét các quan hệ để tìm lời giải như sau :

Giả thiết : { O_1.a, O_1.b, O_1.c, O_1.d, O_1.GA }

<u>Lần lượt thử áp dụng các quan hệ giữa các đối tượng ta tính được</u> :

O_2.b,	nhờ áp dụng f_2
O_2.c,	nhờ áp dụng f_3
O_2.GocA,	nhờ áp dụng f_4
O_3.b,	nhờ áp dụng f_6
O_3.c,	nhờ áp dụng f_7
O_4.b,	nhờ áp dụng f_{10}
O_4.c,	nhờ áp dụng f_{11}

O_5.b, nhờ áp dụng f_{14}

O_5. c, nhờ áp dụng f_{15}

Lần lượt xét các đối tượng theo thứ tự O_1, O_2, O_3, O_4, O_5 ta tính được :

O_2.a, O_2.GocB, O_2.GocC, O_2.S, nhờ áp dụng O_2.

Lại xét các quan hệ giữa các đối tượng ta tính được :

O_1.BD, nhờ áp dụng f_1

O_3.a, nhờ áp dụng f_5

Lại xét các đối tượng theo thứ tự O_1, O_2, O_3, O_4, O_5 ta tính được :

O_3.GocA, O_3.GocB, O_3.GocC, O_3.S, nhờ áp dụng O_3.

Lại xét các quan hệ giữa các đối tượng ta tính được :

O_1.C, nhờ áp dụng f_8

O_1.D, nhờ áp dụng f_{20}

O_1.S, nhờ áp dụng f_{21}

Đến đây ta đã đạt được mục tiêu cần tính toán, và có một lời giải như sau :

{ f_2, f_3, f_4, f_6, f_7, f_{10}, f_{11}, f_{14}, f_{15}, O_2, f_1, f_5, O_3, f_8, f_{20}, f_{21} }.

Sau khi rút gọn lời giải trên ta suy ra được một lời giải tốt như sau :

{ f_2, f_3, f_4, f_6, f_7, O_2, f_1, f_5, O_3, f_{21} }.

Theo lời giải nầy, quá trình tính toán diện tích S của tứ giác như sau :

Tính O_2.b, (cạnh AD) áp dụng f_2

Tính O_2.c, (cạnh AB) áp dụng f_3

Tính O_2.GocA, (góc A) áp dụng f_4

Tính O_3.b, (cạnh CD) áp dụng f_6

Tính O_3.c, (cạnh CB) áp dụng f_7

Tính O_2.a,O_2.S, (cạnh BD,diện tích tam giác ABD) áp dụng O_2

Tính O_1.BD, (đường chéo BD của tứ giác) áp dụng f_1

Tính O_3.a, (cạnh BD của tam giác CBD) áp dụng f_5

Tính $O_3.S$, (diện tích tam giác CBD) áp dụng O_3

Tính $O_1.S$, (diện tích tứ giác ACBD) áp dụng f_{21}.

Ví dụ 4.7: Trong ví dụ nầy ta xét lại bài toán trong ví dụ 4.5-2: Cho hình bình hành ABCD. Giả sử M và N là 2 điểm trên AC sao cho AM = CN. Chứng minh rằng tam giác ABM bằng tam giác CDN (xem hình 4.5 ở trên).

Bài toán nầy có thể biểu diễn dưới dạng bài toán (O, F) $\Rightarrow$ G như sau:

O = { O_1, O_2, O_3}

Trong đó O_1 là hình bình hành ABCD, O_2 là tam giác ABM và O_3 là tam giác CDN (được xây dựng trên các điểm A, B, C, D, M và N).

F = { $O_2.b = O_3.b$ (cạnh AM = cạnh CN),

$M \in AC$, $N \in AC$,

$O_2.c = O_1.a$ (cạnh AB = cạnh AB),

$O_3.c = O_1.a$ (cạnh CD = cạnh CD)

}

G = { $O_2 = O_3$ }

Quá trình suy luận giải bài toán nầy có thể đạt được theo thuật giải suy diễn tiến với các heuristics như sau:

Bước giải 1: Phát sinh thêm sự kiện mới

$O_2.c = O_3.c$

Bước giải 2: Phát sinh sự kiện mới cùng với các đối tượng mới:

Góc BAM = Góc BAC

Góc DCN = Góc DCA

Bước giải 3: Đối tượng O_1 sẽ cho ta sự kiện

Góc BAC = Góc DCA

Bước giải 4: Phát sinh sự kiện mới

Góc BAM = Góc DCN

Bước giải 5: Phát sinh sự kiện mới

O_2.GocA = O_3.GocA

Bước giải 6: Phát sinh sự kiện mới

$O_2 = O_3$

Ghi chú: Kỹ thuật mở rộng tính năng cho đối tượng như đã trình bày ở trên cũng có thể áp dụng cho các loại đối tượng khác như các loại đối tượng tam giác và các loại tứ giác khác. Chẳng hạn như một tam giác ABC khi xét các tính toán có liên quan đến đường trung tuyến AM (xem hình 4.7) có thể được suy luận tính toán trong một mạng các tam giác (O, F) với:

O = { O_1, O_2, O_3 }

Trong đó O_1, O_2, O_3 lần lượt là các tam giác ABC, ABM, AMC.

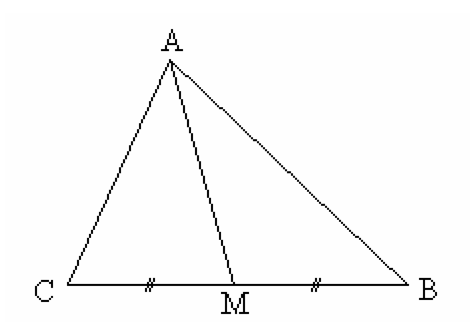
F = { [“TRUNG”, M, DOAN[B,C]],

O_1.a = O_2.a + O_3.a, O_1.b = O_3.b, O_1.c = O_2.c, O_2.b = O_3.c,

O_1.GocA = O_2.GocA + O_3.GocA, O_1.GocB = O_2.GocB,

O_1.GocC = O_3.GocC, O_2.GocC = π - O_3.GocB

}



Hình 4.7 Tam giác ABC và đường trung tuyến AM

CHƯƠNG 5
CÁC ỨNG DỤNG

Trong chương nầy trình bày một số ứng dụng của mạng suy diễn tính toán, mô hình tri thức các C-Object và mô hình mạng các C-Object được trình bày trong các chương 2, 3 và 4. Các ứng dụng nầy bao gồm:

- Chương trình giải toán một C-Object.
- Chương trình giải các bài toán Hình học phẳng.
- Chương trình giải các bài toán Hình học giải tích 3 chiều.
- Chương trình giải một số bài toán về các phản ứng hóa học.

Ngoài ra, luận văn còn thực hiện việc cài đặt một package về mạng suy diễn tính toán tổng quát với đầy đủ các thủ tục giải quyết các vấn đề cơ bản được trình bày trong chương 2. Phần chính của cài đặt package nầy được để trong phần phụ lục.

5.1 CHƯƠNG TRÌNH GIẢI TOÁN C-OBJECT:

Trong phần nầy chúng ta trình bày sự cài đặt trong MAPLE một package "Cobject_Solver" giải toán C-Object tổng quát trong một cơ sở tri thức các C-Object. Như thế với package nầy chúng ta có thể giải tự động các bài toán trên các loại tam giác cũng như trên các loại tứ giác khác nhau vì các tam giác và các tứ giác cùng với những kiến thức về suy diễn và tính toán liên quan có thể được biểu diễn dưới dạng các đối tượng tính toán (C-Object). MAPLE là một phần mềm đại số tính toán (Computer Algebra) khá mạnh với sự hỗ trợ lập trình trên những cấu trúc dữ liệu trừu tượng rất phức tạp thích hợp cho việc cài đặt thử nghiệm các mô hình và phương pháp giải toán.

5.1.1 Sơ đồ hoạt động giải toán của chương trình

Hoạt động giải toán một C-Object của chương trình dựa trên một cơ sở tri thức các C-Object được tổ chức theo mô hình tri thức các C-Object. Tổ chức cơ sở tri thức các C-Object như đã trình bày trong chương 3, bao gồm các tập tin (file) văn bản có cấu trúc chứa đựng các kiến thức mà chương trình cần sử dụng trong việc suy luận và tính toán. Nhắc lại rằng tổ chức các tập tin chính trong cơ sở tri thức các C-Object là:

[1] Tập tin "Objects.txt" lưu trữ các định danh (hay tên gọi) cho các khái niệm về các loại đối tượng C-Object.

[2] Tập tin "RELATIONS.txt" lưu trữ thông tin về các loại quan hệ khác nhau trên các loại C-Object.

[3] Tập tin "Hierarchy.txt" lưu lại các biểu đồ Hasse về quan hệ phân cấp trên các khái niệm.

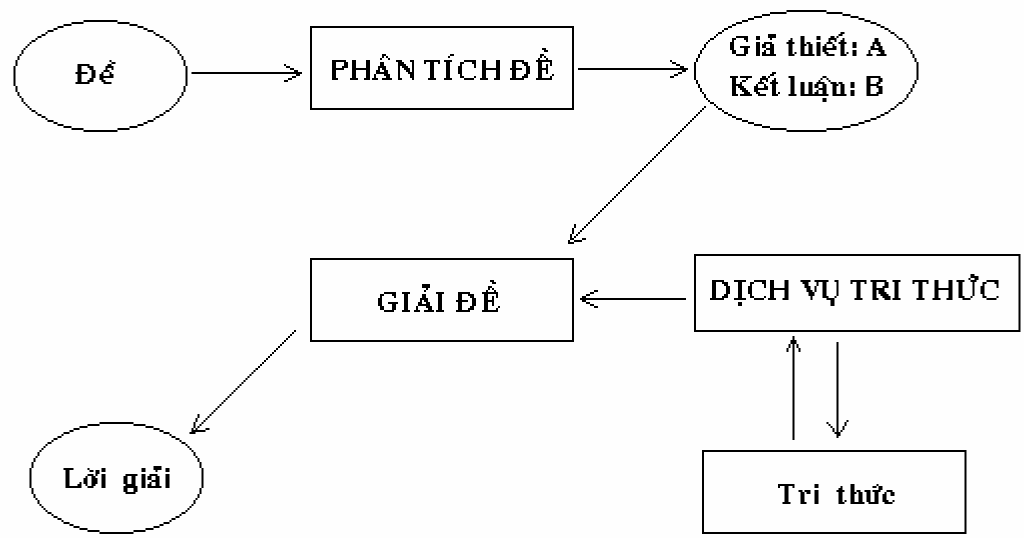
[4] Các tập tin với tên tập tin có dạng "<tên khái niệm C-Object>.txt" để lưu trữ cấu trúc của loại đối tượng <tên khái niệm C-Object>. Ví dụ: tập tin "TAM_GIAC.txt" lưu trữ cấu trúc của loại đối tượng tam giác.

[5] Tập tin "Operators.txt" lưu trữ các thông tin về các toán tử trên các đối tượng.

[6] Tập tin "FACTS.txt" lưu trữ thông tin về các loại sự kiện khác nhau.

[7] Tập tin "RULES.txt" lưu trữ hệ luật của cơ sở tri thức.

Sơ đồ hoạt động tổng quát của chương trình được thể hiện trong hình 5.1 bên dưới. Quá trình hoạt động giải một bài toán có thể được diễn giải như sau: Khi có yêu cầu giải một bài toán, chương trình sẽ đọc đề bài và phân tích đề bài toán. Việc phân tích đề sẽ xác lập mô hình bài toán gồm: các đối tượng, các thuộc tính được quan tâm, các tham biến, các sự kiện (gồm các sự kiện thể hiện quan hệ tính toán, các sự kiện thể hiện quan hệ khác, các sự kiện phân loại đối tượng, . . .) và mục tiêu của bài toán. Sau đó bộ phận giải đề sẽ thực hiện việc tìm kiếm lời giải dựa trên cơ sở tri thức đã có, và thể hiện lời giải tìm được.



Hình 5.1 Sơ đồ tổng quát của hoạt động giải một đề bài toán

5.1.2 Qui ước về đề bài

Đề bài có thể được cho dưới dạng một tập tin văn bản có cấu trúc khá tự nhiên và đơn giản dựa trên một vài từ khóa như "`begin_hypothesis`", "`end_hypothesis`", "`parameters`", "`objects`", "`facts`", "`begin_goal`", "`end_goal`" cùng với một số qui ước khai báo các tham biến, các đối tượng, các sự kiện và yêu cầu của bài toán. Đây cũng là qui ước chung cho việc khai báo một bài toán tổng quát theo mô hình mạng các C-Object trong một COKB. Cấu trúc của đề bài toán có dạng như sau:

```
begin_hypothesis
    parameters:    <các tham biến>
    objects:
        <các đối tượng>  :  <kiểu đối tượng>
    facts:
      <các sự kiện>
```

```
    ...
end_hypothesis
begin_goal
    <mục tiêu của bài toán>
end_goal
```

Ta có thể gọi các chức năng giải toán C-Object và cho biết tập tin đề bài. Tuy nhiên ta cũng có thể yêu cầu giải một bài toán trên một C-Object bằng cách cung cấp trực tiếp những thông tin của đề bài toán bao gồm: <kiểu đối tượng>, <tập hợp gồm các sự kiện giả thiết> và < tập hợp gồm các sự kiện mục tiêu>.

Ví dụ 5.1: Trong một tam giác giả sử cho trước cạnh a, và biết b = 5. Ngoài ra giả thiết rằng ta có các sự kiện sau đây: GocA = 2 GocB, $a^2 = b^2 + c^2$. Hãy xác định góc B và góc C của tam giác.

Bài toán nầy được khai báo theo cấu trúc qui ước trên như sau:

```
begin_hypothesis
   objects:
      Obj : TAM_GIAC;
   facts:
     Obj.a, Obj.b = 5
     Obj.GocA = 2*Obj.GocB
     (Obj.a)^2 = (Obj.b)^2 + (Obj.c)^2
end_hypothesis
begin_goal
     Obj.GocB, Obj.GocC
end_goal
```

Nếu ta muốn yêu cầu giải bài toán trên bằng cách gọi trực tiếp các thủ tục giải và cung cấp cho thủ tục các thông tin về đề bài thì ta có thể viết dòng lệnh gọi

Solvable1s("TAM_GIAC", {a,b = 5, GocA = 2*GocB, a^2 = b^2 + c^2}, {GocB, GocC})

khi muốn xét tính giải được của bài toán, hay là viết dòng lệnh gọi

ComputeVars("TAM_GIAC", {a,b = 5, GocA = 2*GocB, a^2 = b^2 + c^2}, {GocB, GocC})

khi muốn tính GocB và GocC.

<u>Ví dụ 5.2</u>: Xét một tứ giác (lồi) ABCD. Giả sử biết các cạnh a, b, c, d và góc A. Hãy tính diện tích của tứ giác.

Bài toán nầy được khai báo theo cấu trúc qui ước trên như sau:

```
begin_hypothesis
   objects:
      Obj : TU_GIAC;
   facts:
     Obj.a, Obj.b, Obj.c, Obj.d
     Obj.GocA
end_hypothesis
begin_goal
     Obj.S
end_goal
```

Nếu ta muốn yêu cầu giải bài toán trên bằng cách gọi trực tiếp các thủ tục giải và cung cấp cho thủ tục các thông tin về đề bài thì ta có thể viết dòng lệnh gọi

Solvable1s("TU_GIAC", {a, b, c, d, GocA},{ S})

khi muốn xét tính giải được của bài toán, hay là viết dòng lệnh gọi

ComputeVars("TU_GIAC", {a, b, c, d, GocA},{ S})

khi muốn tính diện tích S.

5.1.3 Một số thủ tục chính

Dưới đây là trích dẫn của 3 thủ tục chính được cài đặt trong MAPLE 6.0 với chức năng chính của thủ tục như sau:

- Thủ tục **Solvable1s**(<kiểu đối tượng>, <GT>, <KL>)

  Trong đó GT là tập hợp gồm các sự kiện giả thiết, KL là tập hợp gồm các sự kiện mục tiêu. Thủ tục nầy có nhiệm vụ xác định xem bài toán có giải được hay không, và trong trường hợp giải được thì nó cho ta một lời gải (bao gồm một danh sách các bước giải).
- Thủ tục **ComputeVars**(<kiểu đối tượng>, <GT>, <KL>)

  Trong đó GT là tập hợp gồm các sự kiện giả thiết, KL là tập hợp gồm các biến cần tính toán. Thủ tục nầy có nhiệm vụ thực hiện tính toán và cho ta giá trị các biến cần tính hay một biểu thức tính toán theo các biến và các sự kiện đã biết trong trường hợp bài toán là giải được.
- Thủ tục **IsDetermine**(<kiểu đối tượng>, <Facts>)

  Trong đó Facts là tập hợp gồm các sự kiện đã biết của đối tượng. Thủ tục nầy sẽ khảo sát và xem đối tượng có xác định hay không dựa vào tập các sự kiện đã biết. Nó cho ta kết quả *true* khi đối tượng là xác định và cho kết quả *false* trong trường hợp ngược lại.

```
CONet_Solver[Solvable1s] := proc(nameO, facts, vars::set)
local  factlist,  sol, sol_found, goalvars, varset, set1, set2, Aset, news,
     ilist,
     init_vars, add_properties2, test_goal,
     add_properties, deduce_2s, deduce_32s, deduce_31s,
     ApplyC1, ApplyRules, ApplyC3a, ApplyC2,
     deduce_h2, deduce_h3;

init_vars := proc()
  # khoi tao:  factlist, Aset, goalvars, varset, set1, set2, ilist, sol, sol_found
local  fact;
  factlist := ClassifyFacts(nameO, facts);
```

```
  Aset := convert( ObjStruct(nameO)[2], set);
  set1 := factlist[1];  set2 := map(s->op(SetVars(lhs(s))), factlist[2]);
  ilist := [];
  sol := [];  sol_found := false;
  varset := vars;
  for  fact in facts  do
    varset := varset union `intersect`(SetVars(fact), Aset);
  end do;
  goalvars := vars minus set1;
end:  #  init_vars

add_properties2 := proc()
local fact, newfs, vars;
  newfs := {};  vars := {};
  for  fact in ObjStruct(nameO)[5]  do
      if  KindFact(nameO, fact)=2  then
        newfs := `union`(newfs, {fact});  vars := `union`(vars, SetVars(lhs(fact)) );
      fi;
  end do;
  if  newfs <> {}  then
    factlist[2] := `union`(factlist[2], newfs);  sol := [op(sol), ["properties2",newfs]];
    set2 := `union`(set2, vars);  varset := `union`(varset, vars);
  fi;
end:  #  add_properties2

test_goal := proc()
   goalvars := goalvars minus set1;
   if  goalvars = {}  then  RETURN (true);
   else  RETURN (false);
   fi;
end:

deduce_2s := proc()
local  nfs;
    nfs := `minus`(Deduce_from2s(factlist[2]), factlist[1]);
    if  nfs <> {} then
      factlist[1] := `union`(factlist[1], nfs);   sol := [op(sol), ["Deduce_from2s", nfs]];
      set1 := `union`(set1, nfs);  # <varset>, <set2>  nhu cu
      RETURN (nfs);
    fi;
    RETURN ({});
end:  #  deduce_2s

deduce_32s := proc()
  local  fact, fact1, k1;
    for  fact in factlist[3]  do
```

```
        fact1 := Deduce_from32s(nameO, factlist[2], fact, goalvars);
        k1 := KindFact(nameO, fact1);
        if  k1 = 2 and not Unify_in(nameO, fact1, factlist[2])  then
          factlist[2] := `union`(factlist[2], {fact1});
          sol := [op(sol), ["Deduce_from32s", {fact1}]];
          set2 := `union`(set2, SetVars(lhs(fact1)) );
          RETURN ({fact1});
        elif  k1 = 3 and not Unify_in(nameO, fact1, factlist[3])  then
          factlist[3] := `union`(factlist[3], {fact1});
          sol := [op(sol), ["Deduce_from32s", {fact1}]];
          RETURN ({fact1});
        fi;
      end do;  #  for
      RETURN ({});
end: #  deduce_32s

deduce_31s := proc()
   local  fact, set_1;
      for  fact in factlist[3]  do
        set_1 := Deduce_from31s(nameO, factlist[1], fact);
        if  set_1 <> {}  then
          factlist[1] := `union`(factlist[1], set_1);
          sol := [op(sol), ["Deduce_from31s", set_1]];
          set1 := `union`(set1, set_1);
          RETURN (set_1);
        fi;
      end do;  #  for
      RETURN ({});
end: #  deduce_31s

ApplyC1 := proc()
          . . .
end:  #  ApplyC1

ApplyRules := proc()
          . . .
end:  #  ApplyRules

ApplyC3a := proc()
          . . .
end:  #  ApplyC3a

deduce_h2 := proc()
   # xet cac quan he tinh toan f giong nhu ApplyC2, nhung uu tien su dung nhung quan he
f thoa
```

```
  # dieu kien:  f[3] chua 2 bien khong nam trong (set1 union set2) va co kha nang ket hop
voi
  # mot su kien loai 3 cung co dung 2 bien nhu tren khong thuoc (set1 union set2) de giai
ra
  # su kien moi loai 2, hoac loai 3 nhung tinh theo duoc nhung bien thuoc set1 va set2.
        . . .
end:  #  deduce_h2

deduce_h3 := proc()
  # xet cac quan he tinh toan f giong nhu ApplyC2, nhung uu tien su dung nhung quan he
f thoa
  # dieu kien:  f[3] chua 3 bien khong nam trong (set1 union set2) va co kha nang ket hop
voi
  # mot su kien loai 3 cung co dung 3 bien nhu tren khong thuoc (set1 union set2) de giai
ra
  # su kien moi loai 2, hoac loai 3 nhung tinh theo duoc nhung bien thuoc set1 va set2.
        . . .
end:  #  deduce_h3

ApplyC2 := proc()
        . . .
end:  #  ApplyC2

#  test_Solvable1s 's body

  init_vars();
# Truong hop muc tieu nam trong gia thiet hay trong tinh chat cua object
  add_properties2();
  sol_found := test_goal();
# Duyet cac qui tac co the phat sinh su kien moi
  news := {"khoi_tao_news"};
  while  not sol_found and news <> {}  do
    news := {};
    # thu ap dung qui tac mac nhien Deduce_from2s
    news := deduce_2s();
    if  news <> {} then  sol_found := test_goal();   next;
    fi;
    # thu ap dung qui tac mac nhien Deduce_from32s
    news := deduce_32s();
    if  news <> {} then  sol_found := test_goal();   next;
    fi;
    # thu ap dung qui tac mac nhien Deduce_from31s
    news := deduce_31s();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot quan he tinh toan f theo qui tac 1
```

```
    news := ApplyC1();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot luat khac voi luat dang [{GT}, "Object"]
    news := ApplyRules();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot quan he tinh toan f theo theo heuristic <deduce_h2>
    news := deduce_h2();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot quan he tinh toan f theo theo heuristic <deduce_h3>
    news := deduce_h3();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot quan he tinh toan f theo qui tac C3a
    news := ApplyC3a();
    if  news <> {} then   sol_found := test_goal();   next;
    fi;
    # thu ap dung mot quan he tinh toan f theo qui tac 2
     news := ApplyC2();
     if  news <> {} then   sol_found := test_goal();   next;
     fi;
   end do; # while
  RETURN ([sol_found, sol]);
end:  #  CONet_Solver[Solvable1s]

CONet_Solver[ComputeVars] := proc(nameO::string, facts::set, goalvars::set)
  # Tinh cac bien trong <goalvars> theo cac su kien trong <facts>
  #  RETRUN tap cac su kien loai 2 hay 3 the hien bieu thuc tinh cac bien,
  #  hoac RETURN ({})  neu khong tinh duoc
local  Aset, result, n, i, vars, vars_thay, exprs, fact;

  result := Solvable1s(nameO, facts, goalvars);
  Aset := convert( ObjStruct(nameO)[2], set);
  if  result [1] = false  then  RETURN ({});
  fi;
  n := nops(result[2]);  vars := goalvars;  vars_thay := {}; exprs := {};
  for  i from n to 1 by -1  do
    for  fact in (result[2] [i] [2])  do  # duyet cac su kien moi o buoc suy dien thu i
      if  type(fact, `=`)  then
        if  member(lhs(fact), vars)  then
          exprs := subs (fact, exprs);
          exprs := {op(exprs), fact};  vars := vars minus {lhs(fact)};
          vars_thay := vars_thay union `intersect`(SetVars(rhs(fact)), Aset);
        elif  member(lhs(fact), vars_thay)  then
```

```
            exprs := subs (fact, exprs);
            vars_thay := (vars_thay minus {lhs(fact)}) union
                  `intersect`(SetVars(rhs(fact)), Aset);
          fi;
        fi;
      end do;  #  for fact
    end do;  #  for  i
    RETURN (simplify(exprs));
end:  #  CONet_Solver[ComputeVars]

CONet_Solver[IsDetermine] := proc(nameO::string, facts::set)
  # Xet xem doi tuong co duoc xac dinh tu tap su kien da cho hay khong
  # su dung Solvable1s va cac luat (rule) xac dinh doi tuong
local  factlist, Aset, nvars, vars, rules, i, j, n;
  factlist := ClassifyFacts(nameO, facts);
  Aset := convert( ObjStruct(nameO)[2], set);
  rules := ObjStruct(nameO)[7];  n := nops(rules); # so luat trong Object
  nvars := nops(Aset);
  for  i from 1 to n  do
    if  rules[i] [2] = {"Object"}  then
      vars := `minus`(rules[i] [1], factlist[1]);  nvars := nops(vars);
      break;
    fi;
  end do;
  for  j from (i+1) to n  do
    if  rules[j] [2] = {"Object"} and nops(`minus`(rules[j] [1], factlist[1])) < nvars  then
      vars := `minus`(rules[j] [1], factlist[1]);  nvars := nops(vars);  i := j;
    fi;
  end do;
  # truong hop khong co luat xac dinh doi tuong
  if  nvars = nops(Aset)  then  vars := Aset;
  fi;
  # Xet tinh giai duoc cua:  facts => vars
  RETURN ( Solvable1s(nameO, facts, vars) [1] );
end: #  CONet_Solver[IsDetermine]
```

5.1.4 Lời giải

Chương trình có thể cho lời giải ngắn gọn hoặc chi tiết tùy theo yêu cầu, lời giải có thể được thể hiện trực tiếp trên màn hình hoặc là ghi thành tập tin trên đĩa dưới dạng văn bản có cấu trúc. Đối với bài toán trong ví dụ 5.1 ở trên lời giải tìm được sẽ gồm các bước giải như sau:

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

2. Suy ra $\{GocA = \frac{1}{2}\pi\}$ từ $\{a^2 = b^2 + c^2\}$ bởi một luật suy diễn

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

4. Suy ra $\{GocB\}$ từ $\{GocB = \frac{1}{4}\pi\}$ bởi "Deduce_from3"

5. Suy ra $\{GocC = \frac{1}{4}\pi\}$ từ $\{GocA = \frac{1}{2}\pi, GocB = \frac{1}{4}\pi\}$ bởi áp dụng quan hệ tính toán:

$$["relation\ \ 0", 1, \{GocB, GocA, GocC\}, 1, \{\ \}, GocA + GocB + GocC = Pi, 2]$$

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

Đối với bài toán trong ví dụ 5.2 ở trên, giả sử tứ giác là ABCD, lời giải tìm được sẽ gồm các bước giải như sau:

Tính TAM_GIAC[A,B,D].b, (cạnh AD)

Tính TAM_GIAC[A,B,D].c, (cạnh AB)

Tính TAM_GIAC[A,B,D].GocA, (góc A)

Tính TAM_GIAC[C,B,D].b, (cạnh CD)

Tính TAM_GIAC[C,B,D].c, (cạnh CB)

Tính TAM_GIAC[A,B,D].a, TAM_GIAC[A,B,D].S,

(cạnh BD,diện tích tam giác ABD)

Tính TU_GIAC[A,B,C,D].BD, (đường chéo BD của tứ giác)

Tính TAM_GIAC[C,B,D].a, (cạnh BD của tam giác CBD)

Tính TAM_GIAC[C,B,D].S, (diện tích tam giác CBD)

Tính TU_GIAC[A,B,C,D].S, (diện tích tứ giác ABCD)

5.2 CHƯƠNG TRÌNH GIẢI TOÁN HÌNH HỌC PHẲNG:

Trong phần nầy chúng ta sẽ trình bày về một ứng dụng của mô hình COKB và mạng các C-Object: package giải các bài toán hình học phẳng dựa trên mô hình tri thức các C-Object. Sơ đồ hoạt động giải toán của bài toán theo mô hình mạng các C-Object cũng giống như sơ đồ được trình bày trong phần trên đối với việc giải một C-Object (xem hình 5.1). Kỹ thuật thiết kế các thuật giải đã được trình bày trong chương 3 và chương 4. Phần cài đặt cụ thể tương tự như phần cài đặt mạng suy diễn tính toán và phần cài đặt module giải toán C-Object. Ở đây chúng ta sẽ nêu lên phần ngôn ngữ đặc tả cho bài toán và trình bày một số ví dụ minh họa.

5.1 Ngôn ngữ đặc tả bài toán

- Các từ khóa khai báo cấu trúc đề bài toán: "`begin_hypothesis`", "`end_hypothesis`", "`parameters`", "`objects`", "`facts`", "`begin_goal`", "`end_goal`", "`prove`", "`compute`", "find", "determine". Bài toán được khai báo theo cấu trúc sau đây:

```
begin_hypothesis
    parameters:    <các tham biến>
    objects:
        <các đối tượng>  :  <kiểu đối tượng>
    facts:
      <các sự kiện>
      ...
end_hypothesis
begin_goal
      <mục tiêu của bài toán>
end_goal
```

- Các từ khóa khai báo các đối tượng: đó là các kiểu đối tượng được ghi trong tập tin "Objects.txt". Chẳng hạn như: "DIEM", "DOAN", "GOC",

"TAM_GIAC", "TAM_GIAC_CAN", "TU_GIAC", "HINH_BINH_HANH", . . .

Khai báo các đối tượng có thể viết theo cú pháp sau:

<dãy các tên đối tượng> : <kiểu đối tượng>;

Kiểu đối tượng có thể được ghi dưới dạng một tên kiểu đơn. Kiểu đối tượng còn có thể được ghi dưới dạng một tên cấu trúc thiết lập trên một danh sách nền các đối tượng cơ bản. Chẳng hạn với A, B, C là 3 điểm (các đối tượng kiểu "DIEM") ta có thể khai báo một đối tượng Obj thuộc kiểu "TAM_GIAC" và là tam giác ABC như sau:

Obj : TAM_GIAC[A,B,C];

Hơn nữa trong phần khai báo các sự kiện và mục tiêu ta có thể sử dụng một tên kiểu cấu trúc thiết lập cho một đối tượng mà không cần khai báo trước đối tượng nầy. Chẳng hạn, với A, B, C, D là các điểm (đối tượng kiểu "DIEM") thì các đối tượng là đoạn thẳng AB, đoạn thẳng AC, đoạn thẳng BC. … là không cần thiết phải khai báo trước nhưng ta có thể khai báo sự kiện liên quan đến các đối tượng nầy trong phần sự kiện. Ví dụ như các sự kiện:

["VUONG", DOAN[A,B], DOAN[C,D]]

DOAN[A,C] = DOAN[A,B] + DOAN[B,C]

- Các từ khóa trong cấu trúc khai báo các sự kiện thể hiện một quan hệ theo kiểu quan hệ được ghi trong tập tin "Relations.txt". Chẳng hạn như các loại quan hệ sau đây:

  ["VUONG", "DOAN", "DOAN"]

  ["SSONG", "DOAN", "DOAN"]

  ["=", "TAM_GIAC", "TAM_GIAC"]

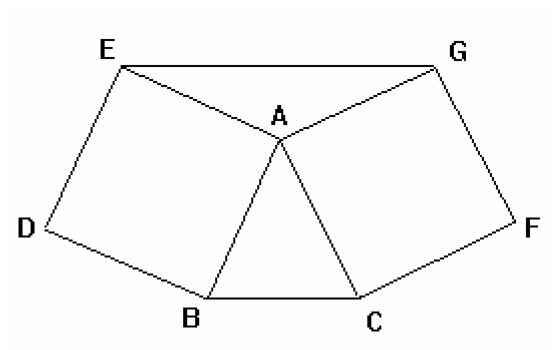
  ["DONGDANG", "TAM_GIAC", "TAM_GIAC"]

["THUOC", "DIEM", "DOAN"]

- Các ký hiệu toán tử (phép toán số học và phép toán so sánh) và các hàm, toán tử "." Để truy cập thành phần của cấu trúc đối tượng.
- Các tên đại diện cho các hằng hay các đối tượng hằng. Ví dụ: số Pi, gốc tọa độ O trong hình học giải tích 2 chiều hay 3 chiều.

5.2 Các ví dụ

Ví dụ 5.3: Xét bài toán: Cho tam giác cân ABC, cân tại A, và cho biết trước góc đỉnh A bằng α, cạnh đáy a bằng m. Bên ngoài tam giác có hai hình vuông ABDE và ACFG. Tính độ dài EG.



- Đặc tả bài toán:

```
begin_hypothesis
objects:
     A, B, C, D, E, F, G : DIEM;
     O1 : TAM_GIAC_CAN[A,B,C];
     O2 : TAM_GIAC[A,G,E];
     O3 : HINH_VUONG[A,E,D,B];
     O4 : HINH_VUONG[A,C,F,G];
facts:
     O1.GOC[C,A,B];
     O1.DOAN[B,C];
     O1.A = Pi - O2.A;
```

```
    end_hypothesis
    begin_goal
        determine: O2.DOAN[E,G]
    end_goal
```

- Lời giải:

Bước 1: determine O2.A, tức là góc A của tam giác AGE.

Bước 2: determine O1.DOAN[A,B].

// trong đối tượng O1

Bước 3: determine O3.DOAN[A, B].

Bước 4: determine O4.DOAN[A, C].

Bước 5: determine O2.DOAN[A, E].

Bước 6: determine O2.DOAN[A, G].

Bước 7: determine O2. DOAN[E, G].

// trong đối tượng O2

Ghi chú: Nếu ta khai báo 2 tham số alpha, m và các sự kiện

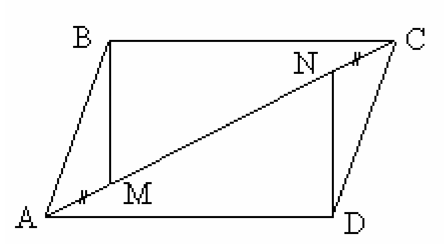
```
    O1.GOC[C,A,B] = alpha,
    O1.DOAN[B,C] = m,
```

và phần mục tiêu sửa lại là

compute: `O2.DOAN[E,G]`

thì chương trình sẽ thực hiện tính toán và cho ta một biểu thức tính cạnh EG của tam giác AGE theo các tham số alpha và m.

Ví dụ 5.4: Xét bài toán: Cho hình bình hành ABCD. Giả sử M và N là 2 điểm trên AC sao cho AM = CN. Chứng minh rằng tam giác ABM bằng tam giác CDN.



- Đặc tả bài toán:

```
begin_hypothesis
objects:
     A, B, C, D, E, M, N : DIEM;
     O1 : HINH_BINH_HANH[A,B,C,d];
     O2 : TAM_GIAC[A,B,M];
     O3 : TAM_GIAC[C,D,N];
facts:
     ["THUOC", M, DOAN[A,C]];
     ["THUOC", N, DOAN[A,C]];
     DOAN[A,M] = DOAN[C,N];
end_hypothesis
begin_goal
     prove: O2 = O3
end_goal
```

Ghi chú: Trong đặc tả trên ta có thể không khai báo 2 đối tượng O1 và O2, nhưng nếu như thế thì trong phần mục tiêu phải viết lại như sau:

```
prove: TAM_GIAC[A,B,M] = TAM_GIAC[C,D,N]
```

- Lời giải:

Bước 1: $\Rightarrow$ $O_2.c = O_3.c$

Vì: DOAN[A,B] = DOAN[C,D]

Bước 2: $\Rightarrow$ GOC[B,A,M] = GOC[B,A,C],

GOC[D,C,N] = GOC[D,C,A].

Vì: ["THUOC", M, DOAN[A,C]],

["THUOC", N, DOAN[A,C]].

Bước 3: $\Rightarrow$ GOC[B,A,C] = GOC[D,C,A].

Vì: đối tượng O1.

Bước 4: $\Rightarrow$ GOC[B,A,M] = GOC[D,C,N].

Vì: GOC[B,A,M] = GOC[B,A,C],

GOC[D,C,N] = GOC[D,C,A],

GOC[B,A,C] = GOC[D,C,A].

Bước 5: $\Rightarrow$ O_2.GocA = O_3.GocA.

Vì: GOC[B,A,M] = GOC[D,C,N].

Bước 6: $\Rightarrow$ $O_2 = O_3$

Vì: luật về tam giác bằng nhau.

5.3 CHƯƠNG TRÌNH GIẢI TOÁN HÌNH HỌC GIẢI TÍCH 3 CHIỀU

Trong phần nầy chúng ta trình bày thiết kế một chương trình giải toán Hình Học Giải Tích 3 chiều dựa trên mô hình tri thức các C-Object (mô hình COKB)và mạng các C-Object. Chương trình đã được cài đặt thử nghiệm dùng Visual C++. Ngoài ra ta cũng có một chương trình tra cứu kiến thức tương ứng được thiết kế theo mô hình COKB được cài đặt thử nghiệm dùng Visual Basic và MS-Access.

5.3.1 Cấu Trúc hệ thống của chương trình

Chương trình gồm 3 thành phần chính: phần giao diện, cở sở tri thức và module giải bài toán. Chương trình có các thực đơn cho người dùng thực hiện việc chọn lựa chức năng, cho phép ta nhập vào đề bài toán. Đề bài toán sẽ được nhập vào dưới dạng một văn bản có cấu trúc dựa trên một ngôn ngữ khai báo bài toán tương tự như đối với chương trình giải bài toán Hình học phẳng ở trên.

Khi có yêu cầu giải bài toán chương trình sẽ tìm lời giải và thể hiện lời giải cũng như hình vẽ trong các cửa sổ trên màn hình.

Cơ sở tri thức của chương trình được thiết kế theo mô hình tri thức các C-Object bao gồm: các khái niệm về các loại đối tượng với các công thức, các khái niểm về các loại quan hệ khác nhau trên các đối tượng, các luật suy diễn, v.v... Cách tổ chức cơ sở tri thức theo mô hình COKB làm cho tổ chức tri thức có cấu trúc tường minh, dễ dàng truy cập và dễ dàng sử dụng trong việc suy luận và giải tự động các bài toán. Chúng ta có thể thiết kế 2 module xử lý tri thức khá đơn giản và hiệu quả. Một module cho phép việc quản lý và tra cứu kiến thức để ta có thể thêm vào hay loại bỏ bớt các khái niệm, các công thức cũng như các luật. Module thứ hai đảm nhiệm chức năng suy luận và giải toán: nó có một mô-tơ suy diễn và dựa trên các sự kiện đã cho trong bài toán cùng với các kiến thức trong cơ sở tri thức để tìm ra lời giải cho bài toán. Quá trình giải bài toán có thể được tóm tắt trong các bước sau đây:

Bước 1: Nhận vào đề bài toán thông qua phần giao diện của chương trình.

Bước 2: Phân tích bài toán và xác định các thông tin trong bài toán gồm: các đối tượng và tham biến, các sự kiện và mục tiêu của bài toán. Thông tin của bài toán nầy được biểu diễn theo mô hình mạng các C-Object và được chuyển cho bộ phận giải.

Bước 3: Dựa trên cơ sở tri thức của hệ thống bộ mô-tơ suy diễn sẽ "suy nghĩ" và tìm cách giải bài toán. Quá trình suy nghĩ nầy cần có một vùng nhớ tạm để ghi nhận những sự kiện phát sinh ra trong quá trình tìm kiếm lời giải cũng như các bước giải (một sự áp dụng luật, một sự tính toán, một sự thiết lập đối tượng mới hay một qui tắc phát sinh sự kiện mới).

Bước 4: Kết quả sẽ là một lời giải (gồm một danh sách bước giải thích hợp để đạt đến mục tiêu) hay là một kết luận về bài toán.

5.3.2 Cơ sở tri thức của chương trình

Tổ chức cơ sở tri thức của chương trình được thực hiện theo mô hình COKB. Một phần trong kiến thức của chương trình được liệt kê dưới đây:

- Các khái niệm về các đối tượng C-Object gồm "DIEM", "DOAN", "VECTOR", "DUONG_THANG", "MAT_PHANG", ... Mỗi đối tượng có các thuộc tính và các tính năng nhất định trong việc giải các bài toán trên một C-Object. Có thể thấy rằng các đối tượng trên trong Hình học giải tích 3 chiều đơn giản hơn các đối tượng hình học phẳng.
- Hệ tọa độ Descartes trực chuẩn gồm điểm gốc O(0,0,0), 3 trục tọa độ Ox, Oy, Oz và 3 mặt phẳng tọa độ Oxy, Oxz, Oyz.
- Các khái niệm về các quan hệ như "SONG_SONG", "VUONG_GOC", "CAT", ...
- Các phép toán như phép tính tích vô hướng của 2 vector, phép chiếu một điểm lên một mặt phẳng, ...
- Các luật suy diễn trên các loại sự kiện khác nhau như các luật sau đây

  [1] "Với 3 vector u, v, w, nếu u // v và $u \perp w$ thì $v \perp w$"

  [2] "Với 2 vector u,v ta có: $u \perp v$ khi và chỉ khi $u.v = 0$"

Theo cách biểu diễn trên của kiến thức Hình học giải tích 3 chiều ta có thể tổ chức các thành phần của cơ sở tri thức như sau:

1. Tự điển các khái niệm về các loại đối tượng, các thuộc tính, các phép toán, các loại quan hệ, các hàm và công thức tính toán, ...
2. Bảng mô tả cấu trúc và tính năng của các đối tượng. Ví dụ: khi một đối tượng mặt phẳng đã được xác định thì ta có thể yêu cầu nó tính toán và

cho ta biết phương trình của mặt phẳng, hay một pháp vector của mặt phẳng.

3. Các luật xác định đối tượng.
4. Các qui tắc hay luật tính toán các giá trị hay các đối tượng.
5. Các luật thể hiện các tính chất của các phép toán.
6. Bảng mô tả các loại sự kiện khác nhau. Ví dụ: sự kiện ["VUONG_GOC", u, v] với u và v là các vector.
7. Bảng mô tả các dạng luật. Ví dụ: Một luật suy diễn sẽ có dạng cấu trúc với 2 thành phần chính là phần giả thiết và phần kết luận. Hai phần nầy đều là các tập hợp gồm một số sự kiện.
8. Danh mục luật.
9. Danh mục các đối tượng và sự kiện chuẩn.
10. Các mẫu bài toán có thể sử dụng trong suy luận để giải các bài toán.

Dựa trên cách tổ chức cơ sở tri thức như trên chúng ta sẽ dễ dàng thực hiện sự tìm kiếm và truy cập vào cơ sở tri thức: các khái niệm, các sự kiện và các luật có thể được bổ sung, thay đổi, hay loại bỏ.

5.3.3 Kỹ thuật giải bài toán

Kỹ thuật giải bài toán ở đây được phát triển dựa trên mô hình mạng các đối tượng trong một COKB. Dạng tổng quát của bài toán có thể được mô hình bởi các tập hợp sau:

$O = \{O_1, O_2, \ldots, O_n\}$,

$R = \{r_1, r_2, \ldots, r_m\}$,

$F = \{f_1, f_2, \ldots, f_p\}$.

Trong đó O là tập hợp các đối tượng, R là tập hợp các sự kiện cho ta các quan hệ về phương diện hình học giữa các đối tượng, và F cũng là tập hợp các sự kiện nhưng gồm các công thức thể hiện các quan hệ tính toán trên các đối tượng và các thuộc tính của chúng. Mục tiêu của bài toán có thể là một hay một số trong các dạng yêu cầu sau:

- Xác định một đối tượng.
- Xác định một thuộc tính của đối tượng.
- Xem xét một quan hệ giữa các đối tượng và/hay giữa các thuộc tính của các đối tượng.
- Tìm kiếm một quan hệ giữa các đối tượng và/hay giữa các thuộc tính của các đối tượng.
- Tìm một biểu thức liên hệ trên các đối tượng và/hay trên các thuộc tính.
- Tính toán một thuộc tính, một đối tượng hay một tham biến.
- Tính toán một kết quả liên quan đến một số đối tượng. Chẳng hạn như tính khoảng cách giữa một điểm và một mặt phẳng, khoảng cách giữa một điểm và một đường thẳng.

<u>Ví dụ 5.5</u>: Cho các điểm E và F, và một đường thẳng (d). Giả sử E, F và (d) đã được xác định. (P) là một mặt phẳng thỏa mãn các sự kiện quan hệ: E $\in$ (P), F $\in$ (P), và (d) // (P). Tìm phương trình tổng quát của mặt phẳng (P).

Trong bài toán nầy các đối tượng, các sự kiện và mục tiêu được liệt kê trong các bảng sau đây:

Loại đối tượng	Tên đối tượng	Sự xác định
Point	E	Yes
Point	F	Yes
Line	D	Yes
Plane	P	No

Bảng 1. Các đối tượng và sự xác định của chúng

Loại quan hệ	Tên đối tượng	Tên đối tượng
$\in$	E	P
$\in$	F	P
//	D	P

Bảng 2. Các quan hệ hình học giữa các đối tượng

Mục tiêu	Đối tượng	Tên đối tượng	Thuộc tính
Attribute	Plane	P	*Equation*

Bảng 3. Mục tiêu của bài toán.

Để tìm được lời giải một cách nhanh chóng chúng ta có thể sử dụng một số heuristic trong phương pháp suy diễn tiến. Khảo sát cách giải của con người đối với nhiều bài toán khác nhau chúng tôi đã sử dụng một số qui tắc heuristic được liệt kê dưới đây:

- Ưu tiên sử dụng các qui tắc xác định đối tượng hay các thuộc tính của đối tượng.
- Ưu tiên sử dụng các qui tắc phát sinh sự kiện quan hệ mới có liên quan đến mục tiêu.
- Ưu tiên sử dụng các qui tắc thiết lập đối tượng mới có liên quan đến mục tiêu.
- Ưu tiên sử dụng bước suy diễn ngược trong trường hợp chỉ có một hướng truy ngược.
- Ưu tiên xác định các đối tượng có quan hệ với mục tiêu.
- Nếu có các tham số cần được xác định thì nên chú ý sử dụng các qui tắc tính toán và các tính chất của các phép toán.

Thuật giải suy diễn tiến sử dụng các heuristic gồm các bước sau đây:

Bước 1: Ghi nhận thông tin của bài toán theo mô hình mạng các đối tượng C-Object.

Bước 2: Khởi tạo lời giải rỗng.

Bước 3: Phân loại các sự kiện giả thiết:

(1) Sự xác định của các đối tượng.

(2) Sự kiên quan hệ hình học giữa các đối tượng.

(3) Sự kiện quan hệ tính toán hay các biểu thức tính toán và các công thức.

(4) Các thuộc tính (của các đối tượng) đã được xác định.

Bước 4: Kiểm tra mục tiêu, nếu đạt được mục tiêu thì chuyển sang bước 10.

Bước 5: Sử dụng các qui tắc heuristics để chọn lựa một bước giải thích hợp nhằm phát sinh thêm các sự kiện mới, các đối tượng mới và đạt đến một trạng thái mới của quá trình suy luận.

Bước 6: Nếu bước 5 thành công thì ghi nhận lại một bước giải gồm qui tắc suy luận, các sự kiện được sử dụng và các sự kiện mới; rồi quay trở lại bước 4.

Bước 7: Tìm một luật có thể áp dụng được để phát sinh ra thêm đối tượng mới hay sự kiện mới.

Bước 8: Nếu bước 7 thất bại (không tìm thấy luật) thì kết thúc và kết luận rằng không tìm thấy lời giải.

Bước 9: Nếu bước 7 thành công thì ghi nhận lại một bước giải gồm qui tắc suy luận, các sự kiện được sử dụng và các sự kiện mới; rồi quay trở lại bước 4.

Bước 10: Thu gọn lời giải được tìm thấy để có một lời giải tốt hơn.

5.3.4 Ngôn ngữ khai báo bài toán

Tương tự như đối với chương trình giải bài toán Hình học phẳng được trình bày trong phần trước, chúng ta thiết kế chương trình theo cách biểu diễn tri thức về các C-Object (mô hình COKB) và mô hình bài toán dạng mạng các C-Object, ta cũng có một ngôn ngữ qui ước khá tự nhiên và đơn giản cho việc đặc tả bài toán. Ngôn ngữ đặc tả bài toán bao gồm các từ khóa, các ký hiện, các toán tử, v.v... Để yêu cầu chương trình giải một bài toán ta chỉ cần khai báo các đối tượng trong bài toán, các sự kiện và mục tiêu của bài toán theo cấu trúc gồm 2 phần như sau:

(1) Khai báo phần giả thiết bao gồm các đối tượng, các sự kiện quan hệ, các quan hệ tính toán (hay các biểu thức), và các yếu tố đã được xác định trước (đối tượng, thuộc tính hay tham biến).

(2) Khai báo phần mục tiêu của bài toán.

Chương trình sẽ "suy nghĩ" tìm lời giải và cho ta lời giải của bài toán. Ta cũng có thể thiết kế khả năng hướng dẫn giải toán của chương trình: chương trình cho những chỉ thị hướng dẫn giải bài toán thay vì cho lời giải của bài toán.

Ví dụ 5.6: Cho các mặt phẳng (Q1), (Q2) và một đường thẳng (d). Giả sử (Q1), (Q2) và (d) được xác định bởi các phương trình:

(Q1) : $2x + y + z = 0$

(Q2) : $x + y + 2z + 1 = 0$

(d) : $(1+t, 2t, 2 + 3t)$

Cho (P) là một mặt phẳng thỏa các điều kiện: (d) // (P), và (P) chứa phần giao (giao tuyến) của 2 mặt phẳng (Q1) và (Q2).Tìm phương trình tổng quát của mặt phẳng (P).

Khai báo bài toán:

```
Hypothesis
  Objects
       Plane Q1(2*x+y+z=0), Q2(x+y+2*z+1=0);
       Plane P;
       Line d(1+t,2*t,2+3*t), d';
  Relations
       d' bel Q1;  d' bel Q2;
       d' bel P;
       d par P;
```

```
EndHypothesis
Goal
      P.gequation    (general equation of plane P)
EndGoal
```

<u>Lời giải của chương trình</u>:

1. d // P suy ra vector u(1,2,3) // P.
2. d' là xác định.
3. Suy ra d'(2*x+y+z=0, x+y+2*z+1=0)
4. Suy ra điểm M(1,-2,0) thuộc P, và vector v(1,-3,1) // P.
5. P được xác định
6. Suy ra P.gequation: 11*x+2*y-5*z-7 = 0

5.4 GIẢI MỘT SỐ BÀI TOÁN VỀ CÁC PHẢN ỨNG HÓA HỌC:

5.4.1 Các bài toán

Chúng ta biết rằng trong hóa học, việc xem xét các phản ứng hóa học là một trong những vấn đề quan trọng. Về mặt tri thức người ta đã biết được nhiều chất và các phản ứng hóa học có thể chuyển hóa từ một số chất nầy thành các chất khác. Tạm thời bỏ qua một số điều kiện phản ứng, ta có thể xem tri thức đó như một mạng tính toán mà mỗi phản ứng là một quan hệ của mạng. Ví dụ như phản ứng điều chế Clo từ axít Clohidric và đioxit mangan :

$$MnO_2 + HCl \rightarrow MnCl_2 + Cl_2\uparrow + H_2O.$$

Phản ứng trên có thể được xem như một quan hệ cho chúng ta có được các chất Cl_2, $MnCl_2$, H_2O từ các chất MnO_2, HCl.

Trong phần nầy chúng ta dùng mạng tính toán để giải 2 bài toán sau :

1. Cho một số chất, hỏi có điều chế được một vài chất nào đó không?
2. Tìm các phương trình phản ứng để biểu diễn dãy các biến hóa, chẳng hạn như các dãy :

$Zn \rightarrow ZnO \rightarrow ZnSO_4$

$S \rightarrow SO_2 \rightarrow SO_3 \rightarrow H_2SO_4$

5.4.2 Một số phản ứng hóa học

Kiến thức ta cần có đối với 2 bài toán trên là tập hợp tất cả các chất cùng với các phản ứng hóa học được phân loại thành các nhóm phản ứng khác nhau. Dưới đây chúng ta liệt kê một số nhóm phản ứng hóa học được lưu trữ trong phần kiến thức của chương trình.

Một số phản ứng liên quan đến khí Clo :

1/ Natri nóng chảy cháy trong Clo cho phản ứng tạo thành natri clorua :

$Na + Cl_2 \rightarrow NaCl$

2/ Bột sắt nóngchảy trong Clo cho phản ứng:

$Fe + Cl_2 \rightarrow FeCl_3$

3 Nung đỏ dây đồng cho vào khí Clo, ta có phản ứng :

$Cu + Cl_2 \rightarrow CuCl_2$

4/ Clo tác dụng với nước :

$Cl_2 + H_2O \rightarrow HCl + HClO$

5/ Điều chế Clo từ axít Clohidric và đioxit mangan :

$MnO_2 + 4HCl = MnCl_2 + Cl_2\uparrow + 2H_2O$

6/ Điều chế Clo bằng axit clohidric và Kali pemanganat :

$HCl + KMnO_4 \rightarrow KCl + MnCl_2 + H_2O + Cl_2\uparrow$

7/ Điện phân dung dịch đậm đặc muối ăn trong nước :

$NaCl + H_2O \rightarrow Cl_2\uparrow + H_2\uparrow + NaOH$

Một số phản ứng khác :

8/ $K + Cl_2 \rightarrow KCl$

9/ $Na + Cl_2 \rightarrow NaCl$

10/ $Al + Cl_2 \rightarrow AlCl_3$

11/ $Ca + Cl_2 \rightarrow CaCl_2$

12/ $FeCl_2 + Cl_2 \rightarrow FeCl_3$

13/ $Cl_2 + KBr \rightarrow KCl + Br_2$

14/ $Cl_2 + NaBr \rightarrow NaCl + Br_2$

Các phản ứng liên quan đến Hidro clorua HCl :

15/ Cho Natri Clorua tinh thể tác dụng với axit sunfuric đậm đặc, đung nóng (phương pháp sunfat), tùy theo nhiệt độ ta có các phản ứng sau đây :

$NaCl + H_2SO_4 \xrightarrow{t^o} NaHSO_4 + HCl\uparrow$

$NaCl + H_2SO_4 \xrightarrow{t^o} Na_2SO_4 + HCl\uparrow$

16/ Phản ứng điều chế HCl bằng phương pháp tổng hợp :

$H_2 + Cl_2 \rightarrow HCl$

Một số phản ứng khác :

17/ $KCl + H_2SO_4 \rightarrow KHSO_4 + HCl$
18/ $KCl + H_2SO_4 \rightarrow K_2SO_4 + HCl$
19/ $CaCl_2 + H_2SO_4 \rightarrow Ca(HSO_4)_2 + HCl$
20/ $CaCl_2 + H_2SO_4 \rightarrow CaSO_4 + HCl$

Các phản ứng của xit clohidric và muối clorua :

21/ $Zn + HCl \rightarrow ZnCl_2 + H_2\uparrow$
22/ $Fe + HCl \rightarrow FeCl_2 + H_2\uparrow$
23/ $Al + HCl \rightarrow AlCl_3 + H_2\uparrow$
24/ $Ca + HCl \rightarrow CaCl_2 + H_2\uparrow$
25/ $Na + HCl \rightarrow NaCl + H_2\uparrow$
26/ $K + HCl \rightarrow KCl + H_2\uparrow$
27/ $Mg + HCl \rightarrow MgCl_2 + H_2\uparrow$
28/ $CuO + HCl \rightarrow CuCl_2 + H_2O$
29/ $ZnO + HCl \rightarrow ZnCl_2 + H_2O$
30/ $MgO + HCl \rightarrow MgCl_2 + H_2O$
31/ $MnO_2 + HCl \rightarrow MnCl_2 + Cl_2\uparrow + H_2O$
32/ $Al_2O_3 + HCl \rightarrow AlCl_3 + H_2O$
33/ $Al(OH)_3 + HCl \rightarrow AlCl_3 + H_2O$
34/ $NaOH + HCl \rightarrow NaCl + H_2O$
35/ $KOH + HCl \rightarrow KCl + H_2O$
36/ $Zn(OH)_2 + HCl \rightarrow ZnCl_2 + H_2O$
37/ $Cu(OH)_2 + HCl \rightarrow CuCl_2 + H_2O$
38/ $Cl_2 + KOH \rightarrow KClO_3 + KCl + H_2O$
39/ $Cl_2 + NaOH \rightarrow NaClO + NaCl + H_2O$
40/ $AgNO_3 + HCl \rightarrow HNO_3 + AgCl\downarrow$
41/ $CaCO_3 + HCl \rightarrow CaCl_2 + H_2O + CO_2\uparrow$
42/ $MgCO_3 + HCl \rightarrow MgCl_2 + H_2O + CO_2\uparrow$
43/ $BaSO_3 + HCl \rightarrow BaCl_2 + H_2SO_3$
44/ $AgNO_3 + NaCl \rightarrow NaNO_3 + AgCl\downarrow$
45/ $AgNO_3 + KCl \rightarrow KNO_3 + AgCl\downarrow$
46/ $H_2S + HClO_3 \rightarrow HCl + H_2SO_4$

Nước Javen : dẫn Clo vào dung dịnh NaOH :

47/ $Cl_2 + NaOH \rightarrow NaCl + NaClO + H_2O$
(Natri hipoclorit)

tương tự ta cũng có :

48/ $Cl_2 + KOH \xrightarrow{t^o} KCl + KClO + H_2O$

Kali Clorat $KClO_3$

Clo đi vào dung dịch kiềm đung nóng đến 100^oC sẽ cho phản ứng :

49/ $Cl_2 + KOH \xrightarrow{t^o} KCl + KClO_3 + H_2O$

Kali Clorat bị phân hủy khi đung nóng theo các phương trình :

50/ $KClO_3 \xrightarrow{t^o} KCl + O_2 \uparrow$

51/ $KClO_3 \xrightarrow{t^o} KCl + KClO_4$

(Kali peclorat)

Clorua vôi : Clo tác dụng với vôi :

52/ $Ca(OH)_2 + Cl_2 \rightarrow CaOCl_2 + H_2O$

Các phản ứng của Brom và Iot :

53/ $Al + Br_2 \rightarrow AlBr_3$

54/ $Al + I_2 \rightarrow AlI_3$

55/ $H_2 + Br_2 \xrightarrow{t^o} HBr$

56/ $H_2 + I_2 \xrightarrow{t^o} HI$

(hidro iotua)

57/ $HI \xrightarrow{t^o} H_2 + I_2$

5.4.3 Các ví dụ

Ví dụ 5.7: Viết phương trình phản ứng biểu diễn các biến hóa sau :

$$Zn \rightarrow ZnO \rightarrow ZnSO_4$$

Giải : Trên cở sở dò tìm các phản ứng (xem là các quan hệ của mạng tính toán các chất hóa học) đã biết ta có thể tìm thấy được các phản ứng sau đây :

$$Zn + O_2 \rightarrow ZnO$$

$$ZnO + H_2SO_4 \rightarrow ZnSO_4 + H_2O$$

Ví dụ 5.8: Từ lưu huỳnh (S) và nước (H_2O) ta có thể điều chế được axit sunfuaric (H_2SO_4) không ?

Giải : Áp dụng các thuật toán tìm lời giải cho mạng tính toán các chất hóa học, dò theo các phản ứng liên quan đến lưu huỳnh và nước ta tìm ra được quá trình điều chế như sau :

$$H_2O \xrightarrow{\text{điện phân}} H_2\uparrow + O_2\uparrow$$

$$S + O_2 \rightarrow SO_2$$

$$SO_2 + O_2 \rightarrow SO_3$$

$$SO_3 + H_2O \rightarrow H_2SO_4$$

Tương tự như 2 ví dụ trên, với tri thức gồm các phản ứng hóa học đã biết dưới dạng một mạng các chất hóa học chúng ta cũng có thể dễ dàng giải các bài toán sau đây :

Ví dụ 5.9: Viết các phương trình phản ứng để thực hiện các biến hóa theo các sơ đồ sau đây :

$$ZnS \rightarrow SO_2 \rightarrow H_2SO_4$$

$$ZnS \rightarrow ZnO \rightarrow ZnCl_2$$

Ví dụ 5.10 : Hoàn thành các phương trình phản ứng sau đây :

$$Mg + H_2SO_4 \rightarrow \ldots$$

$$Fe(OH)_3 + H_2SO_4 \rightarrow \ldots$$

$$K_2CO_3 + H_2SO_4 \rightarrow \ldots$$

$$Ba(NO_3)_2 + H_2SO_4 \rightarrow \ldots$$

CHƯƠNG 6
KẾT LUẬN

Luận án đã tập trung nghiên cứu và phát triển các mô hình biểu diễn tri thức để làm cơ sở và là công cụ cho việc thiết kế cơ sở tri thức, bộ suy luận tự động giải toán cũng như giao diện của các hệ giải toán thông minh. Các hệ thống này có hoạt động tư duy giải toán tương tự như người và có khả năng cho các lời giải tường minh, tự nhiên và phù hợp với cách nghĩ và cách viết của con người. Luận án đã đạt được một số kết quả chính như sau:

Về mặt lý thuyết luận án đã góp phần trong việc phát triển các mô hình biểu diễn tri thức mới. Luận án phân tích và đánh giá các phương pháp biểu diễn tri thức đã biết, khảo sát những kết quả nghiên cứu về lý thuyết cũng như thực hành từ đó xây dựng được một số mô hình biểu diễn tri thức khá tốt có thể sử dụng trong thiết kế các hệ giải toán dựa trên tri thức. Các mô hình này gồm:

1. Mô hình mạng suy diễn tính toán trong đó tích hợp các kỹ thuật biểu diễn mạng ngữ nghĩa, biểu diễn dạng luật dẫn, biểu diễn cấu trúc đặc biệt là biểu diễn trên cơ sở các đối tượng cùng với các kỹ thuật tính toán symbolic trên máy tính.
2. Mô hình tri thức về các C-Object là một mô hình biểu diễn tri thức theo cách tiếp cận hướng đối tượng. Mô hình cho ta một thể hiện tương đối đầy đủ với các cấu trúc tường minh cho một lớp kiến thức tổng quát bao gồm một hệ thống các khái niệm về các C-Object, các quan hệ định tính cũng như định lượng trên các khái niệm C-Object, các loại sự kiện khác nhau liên quan đến các loại quan hệ cùng với các dạng luật khá phong phú.

3. Mô hình mạng các C-Object là một mô hình thể hiện được các dạng bài toán tổng quát trong tri thức về các C-Object.

Trên cơ sở các mô hình biểu diễn tri thức trên luận văn cũng nêu lên cách tổ chức cơ sở tri thức về các C-Object và phát triển các thuật giải để giải quyết các vấn đề cơ bản được đặt ra trên mô hình.

Đối với mô hình mạng suy diễn và tính toán, các vấn đề cơ bản được giải quyết bao gồm:

1. Khảo sát tính giải được của bài toán suy diễn.
2. Tìm bao đóng của một tập thuộc tính.
3. Tìm lời giải cho bài toán và thực hiện các tính toán bao gồm tính toán số và tính toán ký hiệu.
4. Tìm lời giải tối ưu trên mạng suy diễn tính toán có trọng số.
5. Tìm một tập hợp sinh trên mạng suy diễn tính toán và giải quyết vấn đề bổ sung giả thiết cho bài toán.

Đối với mô hình tri thức về các C-Object ta có một tổ chức cơ sở tri thức chặc chẽ và tiện lợi cho việc hiệu chỉnh, truy cập cũng như cho việc sử dụng tri thức trong giải tự động các bài toán.

Đối với mô hình mạng các C-Object, luận văn cũng nêu lên các thuật giải để giải quyết các vấn đề cơ bản đối với bài toán tương tự như các yêu cầu cơ bản đối với bài toán trên mạng suy diễn tính toán nhưng với mức độ khái quát và phạm vi áp dụng rộng và sâu hơn.

Cuối cùng các mô hình biểu diễn tri thức đề ra trong luận văn được chỉ rõ ưu thế và lợi ích của chúng trong việc thiết kế các chương trình giải bài toán thông minh dựa trên tri thức thông qua việc thiết kế và cài đặt một số chương trình ứng dụng cụ thể gồm:

1. Chương trình giải toán một C-Object tổng quát. Chương trình này có khả năng giải các bài toán trên một C-Object bất kỳ có trong cơ sở tri thức của chương trình như: giải các bài toán trên tam giác, trên tứ giác.
2. Chương trình giải các bài toán hình học phẳng trong đó có thể giải các bài toán tính toán trên ký hiệu cũ như giải các bài toán suy diễn trên các sự kiện quan hệ hình học và các bài toán suy diễn liên quan đến cả tính toán lẫn quan hệ.
3. Chương trình giải bài toán hình học giải tích 3 chiều có khả năng thực hiện các suy diễn và tính toán để giải được nhiều dạng bài toán khác nhau.
4. Chương trình giải một số bài toán suy diễn trên các phản ứng hóa học.

Hướng phát triển của luận văn:

1. Tiếp tục phát triển và hoàn thiện mô hình tri thức về các C-Object tiến tới xây dựng một hệ cơ sở tri thức cho lớp các kiến thức loại nầy với đầy đủ các chức năng như hiệu chỉnh, truy cập, trao đổi tri thức phân tán và giải toán tự động.
2. Áp dụng các mô hình để phát triển các sản phẩm phần mềm giải toán hoàn chỉnh như giải các bài toán về đại số, giải tích, hình học, vật lý, hóa học v.v... trong đó cung cấp nhiều chức năng đa dạng được phân lớp để có thể dùng trong tra cứu kiến thức, học kiến thức, học giải toán với nhiều mức độ khác nhau.
3. Phát triển các kỹ thuật học và khám phá tri thức dựa trên các mô hình biểu diễn tri thức để tăng cường khả năng học và phát triển tri thức của các hệ giải bài toán thông minh.