Tạp chí Tin học và Điều khiển học, T.14, S.4 (1998) (41-49)

NÂNG CAO HIỆU QUẢ SUY DIỄN TỰ ĐỘNG TRONG BÀI TOÁN CHỨNG MINH

NGUYỄN THANH THỦY, NGUYỄN HỮU ĐỨC, NGUYỄN VŨ QUỐC HƯNG

Abstract. In this paper we shall investigate some techniques and strategies for improving the inference efficiency in automatic proof. We propose three types of techniques. Restriction and filtering, heuristics - based technique and learning. The two typical knowledge representations (propositional logics and predicate logics) and corresponding proposed heuristics are carefully considered. We applied already these techniques in an Intelligent Tutoring System for geometrical proof problem Geometry.

1. NHẬP ĐỀ

Quá trình suy diễn dựa vào 2 thành phần chính:

- Tri thức chuyên gia: Chứa đựng các tri thức liên quan, bao gồm các sự kiện và suy luận.
- Cơ chế suy diễn trên các tri thức.

Có thể hiểu một cách đơn giản cơ chế suy luận này như sau: Cho một tập các sự kiện ban đầu (facts), dựa trên một số quy tắc suy luận (rules) ta cần phải dẫn tới một (hay nhiều) kết luận.

Để mô phỏng cơ chế suy luận trên máy tính ta phải xây dựng được cơ sở các sự kiện, cơ sở các quy tắc suy luận và mô tơ suy diễn.

Có nhiều cách thức biểu diễn tri thức khác nhau như luật sản xuất, mạng ngữ nghĩa, logic mệnh đề, logic vị từ, frame, bộ ba liên hợp O-A-V, biểu diễn hướng đối tượng, cơ sở tri thức mờ... [1].

Mô tơ suy diễn (công cụ để tiến hành quá trình suy luận) bao hàm một số phương pháp phổ biến sau:

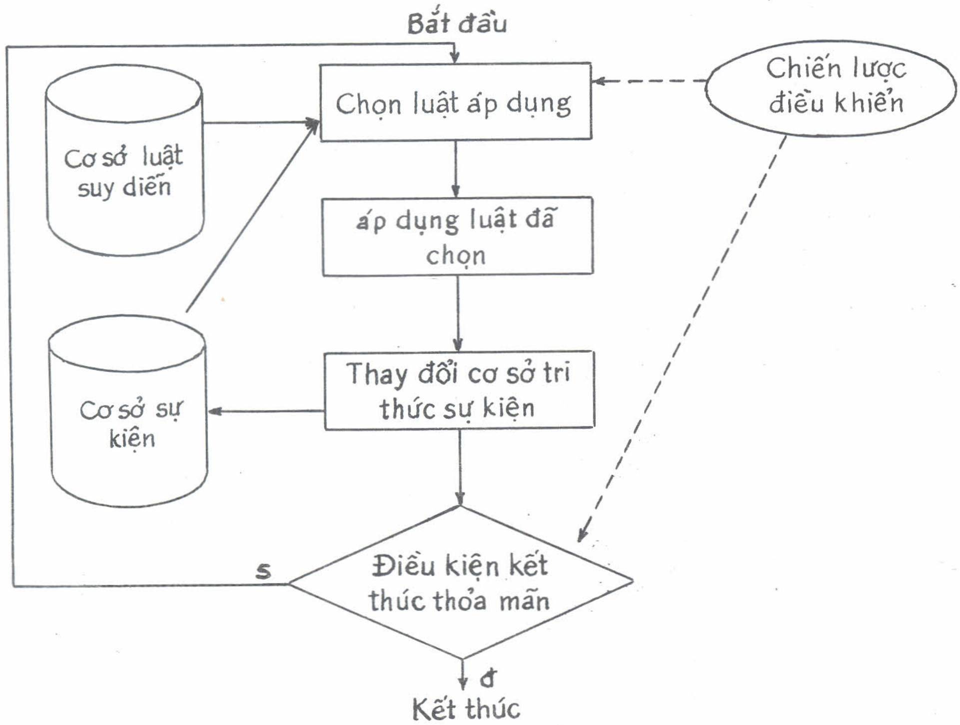
- Cơ chế suy diễn tiến: Xuất phát từ giả thiết GT người ta áp sụng các quy tắc suy diễn để tìm ra các sự kiện trung gian cho đến khi đạt tới kết luận KL
$$\mathrm{GT} = \mathrm{TG}_1 \Rightarrow \mathrm{TG}_2 \Rightarrow \cdots \Rightarrow \mathrm{TG}_n \supseteq \mathrm{KL}$$
- Cơ chế suy diễn lùi: Xuất phát từ kết luận, áp dụng các quy tắc suy diễn để thay thế kết luận bằng các kết luận trung gian dễ hơn cho đến khi các kết luận trung gian này được chứng minh (thuộc giả thiết)
$$\mathrm{KL} = \mathrm{KL}_1 \Leftarrow \mathrm{KL}_2 \Leftarrow \cdots \Leftarrow \mathrm{KL}_n \subseteq \mathrm{GT}$$
- Cơ chế suy diễn hỗn hợp: Có thể biểu diễn như sơ đồ 1.

Vấn đề đặt ra ở đây (cũng chính là trọng tâm của bài báo) là làm thế nào để tăng tốc độ suy diễn của mô tơ suy diễn.

Yếu tố ảnh hưởng đến tốc độ của quá trình suy diễn đối với một cơ sở luật - sự kiện nhất định chính là phương án lựa chọn luật áp dụng phù hợp. Nói một cách nôm na: nếu ta chọn đúng hướng, quá trình suy diễn sẽ giảm thiểu rất nhiều, còn ngược lại, nếu ta chọn lệch hướng, ta vẫn có thể đạt đến kết quả nhưng sẽ tốn rất nhiều thời gian xử lý cho những hướng đi sai.

Vậy làm thế nào để có được lựa chọn đúng đắn? Thực tế để có được phương pháp lựa chọn hoàn thiện là rất khó khăn. Thậm chí đối với một số bài toán, không thể có phương pháp hoàn thiện. Tuy nhiên, ở đây có thể xây dựng những phương pháp chọn heuristic và nếu thực hiện được thì nó cũng làm giảm đi khá nhiều thời gian thực hiện suy diễn.

Sơ đồ 1. Cơ chế suy diễn hỗn hợp



Ở đây, ký hiệu – – – → được dùng để chỉ những chỗ có thể áp dụng các cơ chế điều khiển.

Trong các phần tiếp theo đây chúng tôi muốn đưa ra một số kỹ thuật làm giảm thiểu thời gian thực hiện.

a. Kỹ thuật hạn chế và lọc

Tư tưởng chính ở đây là loại bỏ những luật, sự kiện không áp dụng được hoặc không liên quan đến bài toán đang thực hiện hoặc giới hạn các luật, sự kiện đó ở một phạm vi dễ áp dụng nhất. Ví dụ như trong bài toán chứng minh hình học phẳng: nếu như bài toán chỉ liên quan đến khía cạnh về đường thẳng, đoạn thẳng thì các định lý về hình tròn có thể là không cần thiết (cần nhấn mạnh "có thể" là vì nhiều khi ta vẫn có thể áp dụng được những định lý này) do đó ta sẽ loại bớt những định lý này. Điều này sẽ giúp ta giảm được một số lượng lớn các định lý áp dụng.

Bước tiếp theo sau khi đã loại bớt những luật, sự kiện không cần thiết, ta tiến hành việc lọc. Quá trình này sẽ xác định được những luật nào thật sự áp dụng được.

Mô tơ suy diễn có hạn chế và lọc được biểu thị trên sơ đồ 2.

b. Kỹ thuật lựa chọn heuristic

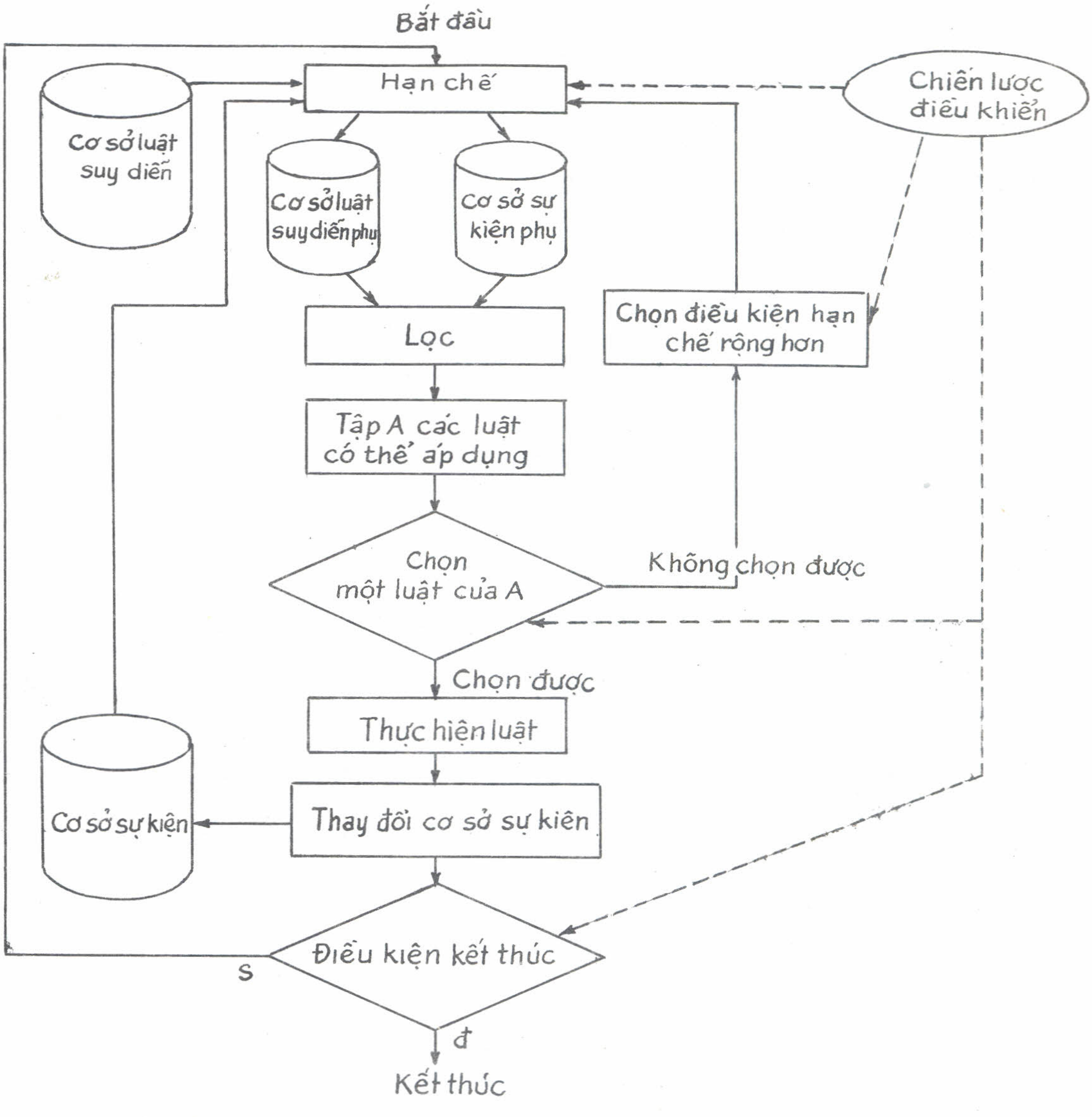
Sau khi đã được tập A các luật có thể áp dụng được, lại nảy sinh một vấn đề mới: chọn luật áp dụng nào trong tập A. Nếu lựa chọn sai lầm cũng sẽ dẫn đến việc tiêu phí thời gian suy diễn cho những hướng đi sai. Để giải quyết, ta phải áp dụng cách thức lựa chọn phù hợp, phương pháp lựa chọn thông minh dựa trên heuristic. Thông thường, cách thức áp dụng này phụ thuộc vào từng bài toán cụ thể, tuy nhiên ta có thể vạch ra một số hướng chính để xây dựng các hàm lựa chọn heuristic như sau:

- Hàm đánh giá mức độ cần thiết của một sự kiện để chứng minh một sự kiện khác.
- Hàm đánh giá mức độ cần thiết của một luật để chứng minh một sự kiện.
- Hàm đánh giá mức độ cần thiết của một sự kiện để áp dụng một luật.

- Hàm đánh giá mức độ cần thiết của một luật để áp dụng một luật khác.

Các chi tiết cụ thể áp dụng cho bài toán suy diễn với logic mệnh đề, vị từ sẽ được trình bày trong các phần 2, 3 của bài báo này.

Sơ đồ 2. Mô tơ suy diễn có hạn chế và lọc



c. Kỹ thuật học

Ta biết rằng đối với con người, quá trình suy luận không phải dựa hoàn toàn trên cơ sở các sự kiện và các quy tắc suy luận mà còn dựa trên kinh nghiệm. Một bài toán lớn khi đã phân rã thành nhiều bài toán nhỏ, nếu như có những bài toán nhỏ tương tự đã được giải trước đây thì trong quá trình suy luận, họ sẽ không phải giải quyết lại những bài toán này nữa. Do quá trình suy luận sẽ rút gọn rất nhiều. Để áp dụng tư tưởng này, trên máy tính, ta xây dựng một số cơ chế học. Đối với mỗi bài toán đã giải, ta sẽ phân tích những kỹ thuật "mới" áp dụng cho bài toán này. Nếu kỹ thuật này được áp dụng nhiều lần trong các bài toán thì ta sẽ tổng quát hóa nó lên thành một dạng quy tắc suy diễn mới nhờ một cơ chế điều khiển học. Khi đó, với một bài toán mới có chứa đựng các bài toán con này ta sẽ áp dụng các luật mới học này để giải quyết. Với phương pháp học này, quá trình suy luận sẽ không chỉ dựa vào các tri thức tổng quát của lĩnh vực và của tình huống cụ thể mà còn dựa vào tri thức thu thập được trong các tình huống đã giải quyết. Để xây dựng một cơ chế học như vậy ta phải hướng đến xây dựng một hệ chuyên gia mà mục tiêu của nó là tạo ra các tri thức chuyên gia về quy tắc áp dụng.

Sau khi đã liệt kê một số kỹ thuật như trên, chúng tôi sẽ trình bày tiếp cách áp dụng các kỹ thuật đó trong một số biểu diễn cụ thể hoặc bài toán cụ thể.

Xem xét các bài toán mà tri thức được biểu diễn dưới dạng logic mệnh đề hoặc logic vị từ với hướng suy diễn tiến.

2. NÂNG CAO HIỆU QUẢ SUY DIỄN VỚI LOGIC MỆNH ĐỀ

2.1. Bài toán suy diễn trong logic mệnh đề [1]

Các luật có dạng $p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q$ với ý nghĩa là nếu $p_1, p_2, \ldots, p_n$ là đúng thì q là đúng.

Bài toán chứng minh dựa trên logic mệnh đề có thể phát biểu:

Vào: Tập GT = $\{h_1, h_2, \ldots, h_n\}$ các giả thiết đã cho,
Tập KL = $\{c_1, c_2, \ldots, c_m\}$ các kết luận cần chứng minh,
Tập R = $\{r_1, r_2, \ldots, r_l\}$ các luật suy diễn cho phép.

Ra: Vết suy diễn Vet = $r_{i1}, r_{i2}, \ldots, r_{ik}$ để dẫn từ GT đến KL.

Tại một thời điểm nào đó, ta có tập sự kiện trung gian (các sự kiện đã chứng minh) được ký hiệu bởi TG = $\{f_1, f_2, \ldots, f_t\}$. Các luật có thể áp dụng được cho bởi tập Sat = $\{r : \text{Left} \Rightarrow q \mid \text{Left} \subseteq \text{TG}\}$.

2.2. Kỹ thuật hạn chế và lọc

Dựa trên tập R = $\{r_1, r_2, \ldots, r_l\}$ với r_i có dạng $p_{i1} \wedge p_{i2} \wedge \ldots \wedge p_{ik} \Rightarrow q$ ta có thể hình thành được một đồ thị có hướng mà các đỉnh là các sự kiện p_i, một cạnh đi từ p_i đến p_j có nhãn là luật r_k nếu như $p_i \in \text{Left}(r_k)$ và $p_j = \text{Right}(r_k)$. Với đồ thị như vậy ta có thể hình thành một chiến lược hạn chế tương đối như sau:

Đồ thị tạo ra sẽ phân chia thành nhiều phần liên thông, mỗi phần này sẽ đại diện cho một phạm vi áp dụng của bài toán. Như vậy, để chứng minh một sự kiện nào đó ta chỉ cần quan tâm đến phần đồ thị liên thông có chứa sự kiện cần chứng minh. Từ đó ta có thể chọn được tập các luật và sự kiện liên quan, và hạn chế được các luật và sự kiện trong phần đồ thị khác. Đó là một cách thức đơn giản để hạn chế luật. Mở rộng đối với phương pháp này ta có thể lựa chọn một phần đồ thị con tương đối độc lập theo một tiêu chuẩn nào đó.

Với các luật đạt được, kết hợp với bộ sự kiện trung gian TG ta có thể tìm những luật nào có thể áp dụng.

2.3. Kỹ thuật lựa chọn heuristic

Trước khi đề cập đến những kỹ thuật cụ thể, chúng tôi muốn đưa ra một số khái niệm:

- Một sự kiện f_1 được gọi là đứng trước một sự kiện f_2 nếu như tồn tại luật r có dạng như sau: $r : \text{Left} \Rightarrow f_2$ và $f_1 \in \text{Left}$.
- Một luật r được gọi là đứng trước một sự kiện f nếu như f là vế phải của r
  $r : \text{Left} \Rightarrow f \in \text{Left}$.
- Một sự kiện f được gọi là đứng trước một luật r nếu như f thuộc vế trái của r
  $r : \text{Left} \Rightarrow q,\ f \in \text{Left}$.
- Một luật r_1 được gọi là đứng trước một luật r_2 nếu như luật r_1 đứng trước một sự kiện f nào đó và f lại đứng trước r_2
  $r_1 : \text{Left}_1 \Rightarrow f;\quad r_2 : \text{Left}_2 \Rightarrow q;\quad f \in \text{Left}_2$.

Dựa trên các khái niệm này, ta có thể xây dựng các hàm heuristic như sau:

1. Hàm đánh giá mức độ cần thiết của sự kiện h để chứng minh một sự kiện c

Định nghĩa 1.

MDCT1 $(h, c) = 1$ nếu như h đứng trước c.
MDCT1 $(h, c) = 1 + \min(\text{MDCT1}(f, c))$ với mọi f sao cho h đứng trước f.

Có thể xây dựng một ma trận MDCT1 kích thước $n \times n$ với n là số sự kiện đang được xét.

Thuật toán 1. Xây dựng ma trận DMCT1 theo các bước sau:

1. Với tất cả các sự kiện f_1 đứng trước f_2, MDCT1 $(f_1, f_2) = 1$, các giá trị còn lại đặt bằng 0.

2. Để tính một giá trị MDCT1 (f_1, f_2) nào đó chưa được tính (MDCT1 $(f_1, f_2) = 0$ ta sẽ xét tất cả các sự kiện thuộc TG $= \{f \mid f_1$ đứng trước $f\}$.

3. Chọn sự kiện f thuộc TG sao cho MDCT1 (f, f_2) nhỏ nhất và khác 0.

4. Nếu chọn được ta sẽ đặt giá trị MDCT1 $(f_1, f_2) =$ MDCT1 $(f, f_2) + 1$. Ngược lại MDCT1 (f_1, f_2) chưa được tính.

5. Lặp lại bước 2 cho đến khi không thể tính thêm được giá trị nào khác nữa.

Quy tắc lựa chọn dựa trên ma trận MDCT1 như sau: Lựa chọn luật r : Left $\Rightarrow q$ trong các luật đã lọc bằng cách chọn r sao cho MDCT1 (q, c) là nhỏ nhất. Dễ thấy thuật toán 1 có độ phức tạp $O(n^2)$.

2. Hàm đánh giá mức cần thiết của một luật r để chứng minh một sự kiện c

Định nghĩa 2

MDCT2 $(r, c) = 1$ nếu r đứng trước c.
MDCT2 $(r, c) = 1 + \min$ (MDCT2 (r_1, c) với mọi r_1 sao cho r đứng trước r_1.

Giống như hàm MDCT1, ở đây ta cũng xây dựng một ma trận MDCT2 kích thước $m \times n$ với m là số luật áp dụng và n là số sự kiện.

Thuật toán 2. Xây dựng ma trận MDCT2 theo các bước sau:

1. Với tất cả các luật r đứng trước f, MDCT2 $(r, f) = 1$, các giá trị còn lại đặt bằng 0.

2. Để tính một giá trị MDCT2 (r, f) nào đó chưa được tính (MDCT2 $(r, f) = 0$) ta sẽ xét tất cả các luật thuộc TG $= \{r_i \mid r_i$ đứng sau $r\}$.

3. Chọn luật r' thuộc TG sao cho MDCT2 (r', f) nhỏ nhất và khác 0.

4. Nếu chọn được thì ta sẽ đặt giá trị MDCT2 $(r, f) =$ MDCT2 $(r', f) + 1$. Ngược lại MDCT2 (r, f) chưa được tính.

5. Lặp lại bước 2 cho đến khi không thể tính thêm được giá trị nào khác nữa.

Quy tắc chọn luật dựa trên MDCT2: Tại một thời điểm dựa trên ma rận MDCT2, ta có thể lựa chọn luật r : Left $\Rightarrow q$ trong các luật đã lọc bằng cách chọn r sao cho MDCT2 (r, c) là nhỏ nhất. Thuật toán 2 có độ phức tạp $O(m \times n)$.

3. NÂNG CAO HIỆU QUẢ SUY DIỄN ĐỐI VỚI LOGIC VỊ TỪ

3.1. Bài toán suy diễn trong logic vị từ [1]

Các luật đều ở dạng chuẩn Horn:

$p_1(X_{11}, X_{12}, ..., X_{1n_1}) \wedge p_2(X_{21}, X_{22}, ..., X_{2n_2}) \wedge \ldots \wedge p_s(X_{s1}, X_{s2}, ..., X_{sn_s}) \Rightarrow q(Y_1, Y_2, ..., Y_m)$

Khác với logic mệnh đề, ở đây một luật có thể áp dụng trên một lớp các vị từ cùng loại.

Ví dụ:

Nếu ta có luật: songsong$(A, B, C, D) \wedge$ songsong$(A, D, B, C) \Rightarrow$ hbh (A, B, C, D).

Ta có thể áp dụng luật này trên hình bình hành $ABCD$ nhưng ta cũng có thể áp dụng nó trên hình bình hành $IJKL$ theo công thức

1. Hàm đánh giá mức độ cần thiết của sự kiện h để chứng minh một sự kiện c

Định nghĩa 1.

MDCT1 $(h, c) = 1$ nếu như h đứng trước c.
MDCT1 $(h, c) = 1 + \min(\text{MDCT1}(f, c))$ với mọi f sao cho h đứng trước f.

Có thể xây dựng một ma trận MDCT1 kích thước $n \times n$ với n là số sự kiện đang được xét.

Thuật toán 1. Xây dựng ma trận DMCT1 theo các bước sau:

1. Với tất cả các sự kiện f_1 đứng trước f_2, MDCT1 $(f_1, f_2) = 1$, các giá trị còn lại đặt bằng 0.

2. Để tính một giá trị MDCT1 (f_1, f_2) nào đó chưa được tính (MDCT1 $(f_1, f_2) = 0$ ta sẽ xét tất cả các sự kiện thuộc TG $= \{f \mid f_1$ đứng trước $f\}$.

3. Chọn sự kiện f thuộc TG sao cho MDCT1 (f, f_2) nhỏ nhất và khác 0.

4. Nếu chọn được ta sẽ đặt giá trị MDCT1 $(f_1, f_2) =$ MDCT1 $(f, f_2) + 1$. Ngược lại MDCT1 (f_1, f_2) chưa được tính.

5. Lặp lại bước 2 cho đến khi không thể tính thêm được giá trị nào khác nữa.

<u>Quy tắc lựa chọn dựa trên ma trận MDCT1 như sau</u>: Lựa chọn luật r : Left $\Rightarrow q$ trong các luật đã lọc bằng cách chọn r sao cho MDCT1 (q, c) là nhỏ nhất. Dễ thấy thuật toán 1 có độ phức tạp $O(n^2)$.

2. Hàm đánh giá mức cần thiết của một luật r để chứng minh một sự kiện c

Định nghĩa 2

MDCT2 $(r, c) = 1$ nếu r đứng trước c.
MDCT2 $(r, c) = 1 + \min(\text{MDCT2}(r_1, c)$ với mọi r_1 sao cho r đứng trước r_1.

Giống như hàm MDCT1, ở đây ta cũng xây dựng một ma trận MDCT2 kích thước $m \times n$ với m là số luật áp dụng và n là số sự kiện.

Thuật toán 2. Xây dựng ma trận MDCT2 theo các bước sau:

1. Với tất cả các luật r đứng trước f, MDCT2 $(r, f) = 1$, các giá trị còn lại đặt bằng 0.

2. Để tính một giá trị MDCT2 (r, f) nào đó chưa được tính (MDCT2 $(r, f) = 0$) ta sẽ xét tất cả các luật thuộc TG $= \{r_i \mid r_i$ đứng sau $r\}$.

3. Chọn luật r' thuộc TG sao cho MDCT2 (r', f) nhỏ nhất và khác 0.

4. Nếu chọn được thì ta sẽ đặt giá trị MDCT2 $(r, f) =$ MDCT2 $(r', f) + 1$. Ngược lại MDCT2 (r, f) chưa được tính.

5. Lặp lại bước 2 cho đến khi không thể tính thêm được giá trị nào khác nữa.

<u>Quy tắc chọn luật dựa trên MDCT2</u>: Tại một thời điểm dựa trên ma rận MDCT2, ta có thể lựa chọn luật r : Left $\Rightarrow q$ trong các luật đã lọc bằng cách chọn r sao cho MDCT2 (r, c) là nhỏ nhất.

Thuật toán 2 có độ phức tạp $O(m \times n)$.

3. NÂNG CAO HIỆU QUẢ SUY DIỄN ĐỐI VỚI LOGIC VỊ TỪ

3.1. Bài toán suy diễn trong logic vị từ [1]

Các luật đều ở dạng chuẩn Horn:

$p_1(X_{11}, X_{12}, ..., X_{1n_1}) \wedge p_2(X_{21}, X_{22}, ..., X_{2n_2}) \wedge \ldots \wedge p_s(X_{s1}, X_{s2}, ..., X_{sn_s}) \Rightarrow q(Y_1, Y_2, ..., Y_m)$

Khác với logic mệnh đề, ở đây một luật có thể áp dụng trên một lớp các vị từ cùng loại.

Ví dụ:

Nếu ta có luật: songsong$(A, B, C, D) \wedge$ songsong$(A, D, B, C) \Rightarrow$ hbh (A, B, C, D).

Ta có thể áp dụng luật này trên hình bình hành $ABCD$ nhưng ta cũng có thể áp dụng nó trên hình bình hành $IJKL$ theo công thức songsong$(I, J, K, L) \wedge$ songsong$(I, L, J, K) \Rightarrow$ hbh (I, J, K, L).

Để thực hiện các suy diễn trên logic vị từ người ta đã đưa ra khái niệm hàm gán trị θ: Nếu cho $\theta = \{X/A,\ Y/B,\ Z/C,\ T/D\}$ thì phép gán trị

songsong$(X, Y, Z, T)\,\theta =$ songsong(A, B, D, D).

Một luật r có thể áp dụng được nếu như tồn tại một hàm gán trị θ sao cho với mọi vị từ $p(X_1, X_2, ..., X_n) \in$ Left(r) thì $p\theta$ là một vị từ mang giá trị đúng. Kết quả của phép áp dụng r là ta có một vị từ đúng mới: $q(Y_1, Y_2, ..., Y_m)\theta$.

Độ phức tạp trong suy diễn trên cơ sở logic vị từ lớn hơn rất nhiều so với logic mệnh đề chính vì tác động của các phép gán trị này. Đối với suy diễn vị từ, các kỹ thuật nâng cao tốc độ suy diễn rất quan trọng vì nó cho phép ta giảm bớt nhiều thao tác suy diễn thừa. Xem xét các kỹ thuật này trên một mô tơ suy diễn tiến:

Thuật toán 3. Suy diễn tiến trên cơ sở logic vị từ:

Đầu vào:

Các giả thiết: GT

Các kết luận: KL

Các luật suy diễn LUAT

Đầu ra:

Câu trả lời đúng nếu từ giả thiết ta có thể suy diễn đến kết luận và sai trong trường hợp ngược lại

```
TG := GT;
Sat := Loc(TG,LUAT); /* Lọc các bộ (r,θ) có thể áp dụng được */
While (SAT = ∅) or (KL ∈ TG) do
        (r,θ) := Lay(Sat) /* r : Left ⇒ c(Y1, ..., Yn) */
        TG := TG ∪ c(Y1, ..., Yn)θ;
        Sat := Loc(TG,LUAT);
EndWhile
If KL ∈ TG then
        Suy diễn được
else
        Không thể suy diễn được từ GT đến KL
```

Việc tăng tốc độ trong thuật toán suy diễn tiến như trên phụ thuộc rất nhiều vào 2 thao tác Loc và Lay. Thao tác Loc tương ứng với quá trình hạn chế và lọc như đã nói phần trên, còn Lay tương ứng với việc xây dựng các hàm lựa chọn thích hợp.

Chúng tôi muốn đề cập trong phần này một số các phương pháp áp dụng cho 2 thao tác này.

3.2. Thao tác Loc

Thực chất của quá trình Loc là chọn những bộ (r, θ) có khả năng áp dụng trên tập TG. Nếu xem xét như thuật toán đã trình bày ở trên, thao tác Loc được lặp đi lặp lại sau mỗi lần tìm ra được một luật mới. Nếu như số lượng luật của tập LUAT và số lượng sự kiện của TG lớn thì thao tác này sẽ thực sự trở thành nhân tố làm tăng thời gian suy diễn. Hướng giải quyết vấn đề này là xây dựng kỹ thuật lọc sao cho tránh được các luật không thể áp dụng được và các luật đã được áp dụng. Đối với tập Sat, ta có thể thêm dần các luật chứ không tính toán lại hoàn toàn nó. Các luật mới thêm này phải có dạng Left $\Rightarrow q(Y_1, Y_2, ..., Y_m)$ trong đó vị từ mới suy diễn được $(c(Y_1, Y_2, ..., Y_n))$ phải là một thành phần của Left. Từ đó có thể đưa ra kỹ thuật lọc bao gồm 2 bước như sau:

a/ Lọc sơ bộ: Chọn tập luật RLoc $= \{r :$ Left $\Rightarrow q(\ldots) \mid c \in$ Left$\}$.

b/ Xác định bộ gán trị: Trong tập luật RLoc đã tìm thấy, với mỗi luật r ta phải xác định các bộ gán trị θ thỏa mãn. Muốn làm được điều này, ta tiến hành thử áp dụng r trên bộ TG đã có (cần nhớ là vị từ c bắt buộc phải có mặt trong phép thử này). Phân hoạch tập TG thành các tập vị từ với tên vị từ giống nhau (ví dụ các vị từ cùng một loại songsong) ký hiệu là TG(p)
Nếu luật r có dạng: $p_1 \wedge p_2 \wedge \ldots \wedge p_{i-1} \wedge c \wedge p_{i+1} \wedge \ldots \wedge p_n \Rightarrow q$, ta sẽ thử áp dụng r với tất cả các vị từ $p_j(X_{j1}, X_{j2}, ..., X_{jni})$ sao cho:
$p_j(X_{j1}, X_{j2}, ..., X_{jn_i}) \in \mathrm{TG}(p_j)$ với $j \neq i$,
$p_j(X_{j1}, X_{j2}, ..., X_{jn_i}) = c(Y_1, Y_2, ..., Y_n)$ với $i = j$.
Nếu kết quả áp dụng thử này tốt nó sẽ cho ta một bộ gán trị θ tương ứng và khi đó ta có thể thêm cặp (r, θ) vào tập Sat.
Tiến hành quá trình này với tất cả các luật thuộc RLoc ta sẽ có được các luật và phép gán trị mong muốn.
Cách làm này giúp ta tránh được việc lọc lại trên số lượng lớn các luật.

3.3. Thao tác Lay

Chiến lược suy diễn sẽ phụ thuộc rất nhiều vào thao tác Lay vì trên tập Sat có rất nhiều các luật có thể áp dụng. Chọn một trong nó sẽ cho kết quả một vị từ mới, vị từ mới này sẽ lại điều khiển quá trình chọn luật... Chính vì vậy nếu ta đưa ra được một chiến lược chọn tốt, ta có thể đạt đến kết quả tốt. Cũng như đối với logic mệnh đề, thao tác chọn này tương ứng với các hàm heuristic.

Nếu ta có vị từ $p(X_1, X_2, ..., X_n)$ thì ta gọi p là tên vị từ hay loại vị từ. Khi đó luật suy diễn trong logic vị từ

$r : p_1(X_{11}, X_{12}, ..., X_{1n1}) \wedge p_2(X_{21}, X_{22}, ..., X_{2n2}) \wedge \ldots \wedge p_s(X_{s1}, X_{s2}, ..., X_{sns}) \Rightarrow q(Y_1, Y_2, ..., Y_m)$
sẽ tương ứng với quan hệ phụ thuộc hình thức các loại vị từ sau:
$r' : p_1 \wedge p_2 \wedge \ldots \wedge p_s \Rightarrow q.$

1/ Mức độ cần thiết của một loại vị từ p_1 để chứng minh một loại vi từ p_2

Ta sẽ xây dựng các hàm MDCT3(p_1, p_2), hàm này có ý nghĩa giống như đối với logic mệnh đề. Tuy nhiên, cách thức xây dựng nó thì hơi khác. Bởi vì với một quan hệ p_1 trước p_2 nhưng các cách thể hiện $p_1(\ldots)$, $p_2(\ldots)$ của chúng chưa chắc đã đứng trước nhau trong một luật cụ thể nào đó. Ý tưởng để xây dựng hàm này là chỉ xem xét xem khả năng có thể áp dụng một loại vị từ p_1 để chứng minh loại vị từ p_2. Cách tính toán cụ thể như sau:

Định nghĩa 3. $\mathrm{MDCT3}^1(p_1, p_2)$ được tính bằng tỷ số giữa số lần xuất hiện của p_1 trong các luật suy diễn trực tiếp ta p_2 và tổng số các vị từ xuất hiện ở vế trái các luật đó. Nếu như xét trong một chuỗi suy diễn

$$\mathrm{GT} \Rightarrow \mathrm{TG}_1 \Rightarrow \mathrm{TG}_2 \Rightarrow \cdots \Rightarrow \mathrm{TG}_n \Rightarrow p_2(\cdots),$$

$\mathrm{MDCT3}^1(p_1, p_2)$ được xem như một con số để tính toán khả năng xuất hiện của p_1 trong TG_{n-1}.

Như vậy làm thế nào để có được con số tính toán khả năng xuất hiện của p_2 trong một bước trung gian TG_k nào đó? Ta tiến hành việc xác định ma trận $\mathrm{MDCT3}^{n-k}$ kích thước $n \times n$.

Thuật toán 4. Giả sử đã tìm được ma trận $\mathrm{MDCT3}^i$ kích thước $n \times n$, để tìm được $\mathrm{MDCT3}^{i+1}$ kích thước $n \times n$ ta dựa vào nguyên tắc sau:

$$\mathrm{MDCT3}^{i+1}(p_1, p_2) = \sum \mathrm{MDCT3}^1(p_1, p) * \mathrm{MDCT}^i(p, p_2),$$

ở đây tổng lấy theo p là vị từ thuộc tập các loại vị từ.

Có thể giải thích công thức này như sau: Con số $\mathrm{MDCT3}^i(p_1, p_2)$ đánh giá mức độ cần thiết cấp i của p_1 với p_2 có quan hệ tuyến tính với các số đánh giá mức độ cần thiết cấp $i-1$ của các vị từ p với p_2. Trong đó p là các vị từ được suy luận trực tiếp từ p_1. Hệ số tuyến tính của p được lựa chọn chính là mức độ cần thiết của p_1 để chứng minh $p = \mathrm{MDCT1}^1(p_1, p)$.

Do đó: $\mathrm{MDCT3}^{i+1} = \mathrm{MDCT3}^1 * [\mathrm{MDCT3}^i]^*$ (trong đó $[\mathrm{MDCT3}^i]^*$ là ma trận chuyển vị của $\mathrm{MDCT3}^i$).

Quy tắc lựa chọn dựa trên MDCT3: Sau khi đã tìm được các ma trận $\mathrm{MDCT3}^i$, để có được quyết định lựa chọn tại thời điểm nào đó của quá trình chứng minh ta phải làm các bước sau:

1) Xác định giá trị t (độ phức tạp của bài toán) dựa trên tập TG, KL và tập luật).

2) Xây dựng ma trận quyết định $\mathrm{MTQD1} = \sum_{i=1}^{t} \mathrm{MDCT3}^i$.

3) Chọn luật r trong Sat có dạng Left $\Rightarrow q$ sao cho $\mathrm{MTQD1}(q, c)$ đạt giá trị lớn nhất.

Sở dĩ ta phải thực hiện phép cộng các ma trận vì cần phải tính cả đến các khả năng chứng minh với độ dài của vết suy diễn nhỏ hơn t.

Thuật toán có 4 độ phức tạp $O(t \times n^2)$.

2/ Mức độ cần thiết của một loại luật r để chứng minh một loại vị từ p

Định nghĩa 4. Tương tự như việc xây dựng ma trận ở phần 1/ ta cũng xây dựng một ma trận $\mathrm{MDCT4}^k(r, p)$ để đánh giá mức độ cần thiết mức k của một luật r với loại vị từ p.

Thuật toán 5. Cách tính ma trận MDCT4:

$\mathrm{MCDT4}^1(r, p) = 1$ nếu r có dạng r : Left $\Rightarrow p$.

$\mathrm{MDCT4}^1(r, p) = 0$ trong trường hợp còn lại.

Giả sử ta đã có $\mathrm{MDCT4}^i$ ta sẽ phải tính $\mathrm{MDCT4}^{i+1}$.

Xét một chuỗi suy diễn:

$$\mathrm{TG}_{i+1} \Rightarrow^r \mathrm{TG}_i \Rightarrow^{r'} \cdots \Rightarrow c$$

Nếu xét độ cần thiết của luật r đối với c ($\mathrm{MDCT4}^{i+1}(r, c)$) thì giá trị này sẽ có quan hệ tuyến tính với giá trị $\mathrm{MDCT4}^i(r', c)$ với r' là các luật đứng sau r (giống như định nghĩa trong phần logic mệnh đề). Hệ số tuyến tính này có thể được xem như giá trị cần thiết để của một luật r để có thể sử dụng luật r'. Ta xây dựng một ma trận RR kích thước $m \times m$ để tính toán giá trị cần thiết này.

Giả sử r' có dạng Left $\Rightarrow q$ thì có thể áp dụng được r' ta phải chứng minh được các vị từ thuộc Left. Mà ta có giá trị cần thiết của một luật r để chứng minh trực tiếp một vị từ c được tính bởi $\mathrm{MDCT4}^1(r, c)$.

Do đó công thức để tính $\mathrm{RR}(r, r')$ có thể được bằng $\sum_{p \in \mathrm{Left}(r')} (\mathrm{MDCT4}^1(r, p))/k$, ở đây k là số vị từ có trong $\mathrm{Left}(r')$.

Xây dựng công thức tính $\mathrm{MDCT4}^{i+1}(r, c)$ như sau:

$$\mathrm{MDCT4}^{i+1}(r, c) = \sum \mathrm{RR}(r, r') * \mathrm{MDCT4}^i(r', c),$$

ở đây, tổng được lấy theo các r' thuộc tập các luật.

Do vậy: $\mathrm{MDCT4}^{i+1} = \mathrm{RR} * \mathrm{MDCT4}^i$.

Quy tắc lựa chọn dựa trên MDCT4: Với ma trận MDCT4 ta tiến hành việc lựa chọn giống như phần 1:

1) Xác định giá trị t (độ phức tạp của bài toán) dựa trên tập TG, KL và tập luật.

2) Xây dựng ma trận quyết định $\mathrm{MTQD2} = \sum_{t=1}^{t} \mathrm{MDCT4}^i$.

3) Chọn luật r trong Sat có dạng Left $\Rightarrow q$ sao cho $\mathrm{MTQD2}(q, c)$ đạt giá trị lớn nhất.

Thuật toán 5 có độ phức tạp $O(t \times m^2)$.

Nhận xét:

Thực ra các hàm heuristic này chỉ có ý nghĩa tương đối chứ không có thể cho kết quả tuyệt đối bởi vì còn nhiều yếu tố ảnh hưởng tới quá trình suy diễn như sự liên quan giữa các ký hiệu trong những vị từ GT và KL. Tuy nhiên, nó cũng có tác dụng với khá nhiều trường hợp suy diễn. Hướng phát triển cho những hàm heuristic là tìm ra những quan hệ giữa các vị từ GT và KL (quan hệ về ký hiệu) và những đánh giá độ phức tạp của bài toán.

4. THỬ NGHIỆM CÁC KỸ THUẬT TRONG PHẦN MỀM CHỨNG MINH HÌNH HỌC TRÊN CƠ SỞ LOGIC VỊ TỪ

Khi thiết kế phần mềm trợ giúp chứng minh các bài toán hình học, một vấn đề quan trọng là thiết kế các ngôn ngữ mô tả các đối tượng hình học và các tính chất, định lý hình học [5]. Ngôn ngữ mô tả này dựa trên logic vị từ. Tuy nhiên có một số điểm cần phải thay đổi cho phù hợp.

- Một mệnh đề hình học có thể biểu diễn dưới dạng nhiều vị từ.

Ví dụ: cùng một mệnh đề "3 điểm A, B, C thẳng hàng" ta có các biểu diễn khác nhau:
ThangHang(A, B, C), ThangHang(A, C, B), ThangHang(B, A, C), ThangHang(B, C, A), ThangHang(C, A, B), ThangHang(C, B, A).

- Vấn đề này nảy sinh việc cùng một định lý có thể cho kết quả nhiều vị từ (tuy chúng cùng là một mệnh đề).

Chính vì vậy trong trường hợp đặc biệt này, ta phải tổ chức một định lý thành nhiều luật khác nhau và phải đưa ra một cơ chế để hạn chế việc áp dụng lại cùng một định lý (mặc dù đó là áp dụng những luật khác nhau).

Việc trợ giúp chứng minh có thể thực hiện theo 2 hướng: tiến, lùi, theo phương thức trọn gói hay từng bước dựa trên các nguồn thông tin [3]

- Tập sự kiện đã được chứng minh
- Tập sự kiện phải chứng minh
- Tập định lý có thể dùng
- Hình vẽ

Hệ GEOMETRY Version 1.0 bước đầu đã được thử nghiệm, bao gồm các phân hệ: vẽ hình, soạn thảo Giả thiết / Kết luận, trợ giúp chứng minh và soạn lời chứng minh [6], [7].

TÀI LIỆU THAM KHẢO

[1] Nguyễn Thanh Thủy, *Trí tuệ nhân tạo - các phương pháp giải quyết vấn đề và xử lý tri thức*, Nhà xuất bản Giáo dục, 1995.

[2] Nguyễn Thanh Thủy, *Systèmes Experts: Conception et Réalisation*, Institut Francophone d'Informatique, 1996, 1997.

[3] Nguyễn Thanh Thủy, Các kỹ thuật trợ giúp chứng minh bài toán hình học phẳng , *Tạp chí Tin học và Điều khiển học* **12** (4) (1996).

[4] Michel Pintado, "Contribution de la représentation de connaissances orientée objet et raisonnement par analogie", The doctorgle de l'Université de Paris 6, 1994.

[5] Nguyễn Thanh Thủy, Thiết kế các ngôn ngữ mô tả và xây dựng cơ sở tri thức cho các phần mềm dạy học hình học, *Tạp chí Tin học và Điều khiển học* **12** (2) (1996).

[6] Nguyễn Thanh Thủy, GEOMETRY Un environnement interatif pour résolution de problèmes de géométrie, *Hội nghị SEACME'7*, 1996.

[7] Nguyễn Thanh Thủy, Nguyễn Hữu Đức, Thiết kế và cài đặt giao diện đồ họa trong phần mềm trợ giúp giải bài toán chứng minh hịnh học, *Tạp chí Tin học và Điều khiển học* **13** (2) (1997).

Nhận bài ngày 15 - 7 - 1998

Khoa Công nghệ thông tin
Trường Đại học Bách Khoa, Hà Nội.