Tạp chí Tin học và Điều khiển học, T. 19, S. 1 (2003), 5–16

PHƯƠNG PHÁP LẮC BA LÔ VÀ THUẬT TOÁN TÌM KIẾM XẤP XỈ DÃY CON CHUNG DÀI NHẤT

NGUYỄN QUÝ KHANG

Khoa Toán, trường Đại Học Sư phạm Hà nội 2

Abstract. In a paper entitled: "*Fuzzy Automata and it's applications for finding the longest common subsequence*" (P.T.Huy and N.Q.Khang - The 6-th Vietnam Conference of Mathematics, Hue-7-10/9/2002), using a weakly ordered structure we have introduced Knapsack Shaking Method as fundamentals of application of fuzzy automata for finding the exactly longest common subsequence of two text sequences. In this paper received results will be extended to solve approximate LCS problems using the Knapsack Shaking method. Some new algorithms with their complexities are presented.

Tóm tắt. Trong báo cáo: "*Ôtômát mờ và ứng dụng trong bài toán tìm dãy con chung dài nhất*" (P.T.Huy và N.Q.Khang, Hội nghị Toán học Toàn quốc lần 6, Huế 7-10/9/2002) chúng tôi đã đề xuất phương pháp lắc ba lô nhờ việc xây dựng một cấu trúc thứ tự yếu, làm cơ sở cho việc áp dụng ôtômát mờ để giải bài toán tìm chính xác dãy con chung dài nhất của hai xâu text. Bài báo này mở rộng các kết quả toán học đó và trình bày một số thuật toán ứng dụng phương pháp lắc ba lô vào việc giải bài toán tìm kiếm xấp xỉ dãy con chung dài nhất của hai xâu và xem xét độ phức tạp tính toán của chúng.

1. MỞ ĐẦU

Bài toán so mẫu kiểu xâu ký tự xuất hiện trong nhiều lĩnh vực nghiên cứu, chẳng hạn việc tìm kiếm thông tin tĩnh hoặc động trên mạng. Mọi hệ soạn thảo văn bản hay đa số bộ soạn thảo của ngôn ngữ lập trình không thể thiếu lệnh tìm kiếm; trong sinh học, nghiên cứu các bản đồ gene để phát hiện những thuộc tính cần quan tâm đối với cây trồng hoặc cơ thể sống. Tìm kiếm, so mẫu cũng cần thiết trong viễn thám... Hiện nay, việc xây dựng thuật toán mới hoặc cải tiến các thuật toán so mẫu (chính xác hay xấp xỉ) đang là một vấn đề sôi động.

Các thuật toán so đơn và đa mẫu chính xác hay xấp xỉ đã được trình bày trong các công trình [1,3,4,5,9].

Trong các dạng của bài toán so mẫu, bài toán tìm dãy con chung dài nhất, chính xác hay xấp xỉ, của hai xâu hay nhiều xâu có vai trò quan trọng cả về mặt lý thuyết lẫn ứng dụng (xem [1,9]). Một số cách tiếp cận chủ yếu có thể nêu ra là: phương pháp ôtômát hữu hạn [1,4]; phương pháp quy hoạch động [5] và sử dụng khái niệm *khoảng cách Levenshtein* hay còn gọi là *edit distance* ([1]).

Trong [6] đã đề xuất một hình thức ôtômát mờ, cho phép xem xét một cách thống nhất các dạng so mẫu do Knuth-Morris-Pratt hoặc Boyer-Moore đưa ra. Thuật toán so đơn mẫu là có hiệu quả và dễ cài đặt. Về lĩnh vực này, độc giả quan tâm tới ôtômát mờ, ngôn ngữ và cú pháp mờ... có thể tham khảo [2,8].

Trong [7], chúng tôi đã trình bày một cấu trúc ôtômát mờ giải bài toán tìm chính xác

dãy con chung dài nhất của hai xâu. Cơ sở toán học của nó là nghiên cứu một cấu trúc với các phần tử gọi là cấu hình hay một tập hữu hạn các dãy con của mẫu. Từ đó nghiên cứu nửa nhóm tự do các tác động trên tập cấu hình để nhận được ôtômát mờ giải bài toán tìm dãy con chung dài nhất của hai xâu và đặt tên cho phương pháp đó là *lắc ba lô*. Bằng việc khai thác thông tin về mẫu tương tự như trong [6], phương pháp này tránh được các bước so mẫu không cần thiết. Nó cũng không đòi hỏi bộ nhớ lưu trữ xâu cần duyệt, kể cả dòng dữ liệu trên môi trường mạng.

Trong nhiều ứng dụng, người ta không đòi hỏi xác định chính xác dãy con chung dài nhất mà chỉ cần tìm kiếm xấp xỉ với một độ chính xác nhất định. Mục đích của bài báo này là trình bày các thuật toán tìm kiếm xấp xỉ dãy con chung dài nhất của hai xâu.

Sau phần mở đầu, trong Mục 2 sẽ phát biểu bài toán tìm chính xác dãy con chung dài nhất của hai xâu cũng như bài toán cho trường hợp tìm kiếm xấp xỉ. Mục 3 nhắc lại các kết quả trong [7], làm cơ sở cho Mục 4, trình bày cơ sở toán học của các thuật toán giải quyết bài toán so mẫu xấp xỉ và đánh giá độ phức tạp của chúng.

2. PHÁT BIỂU BÀI TOÁN

Bài toán xác định dãy con chung dài nhất của hai xâu text được xét trong [7] (Longest Common Subsequence Problem-LCS Problem) là cho trước xâu mẫu P và xâu dữ liệu S, ta xây dựng thuật toán:

- Xác định chính xác dãy con chung dài nhất của P và S.
- Xác định độ dài của dãy con chung dài nhất của P và S, không đòi hỏi chỉ ra dãy con cụ thể.

Bài toán xác định xấp xỉ dãy con chung dài nhất của hai xâu text P và S cũng là bài toán LCS cho trường hợp tìm kiếm xấp xỉ, nghĩa là xây dựng thuật toán để:

- Xác định dãy con chung của P và S với độ dài lớn hơn hoặc bằng $c.l(P)$ nếu có, trong đó c là một hằng số cho trước, $l(P)$ là độ dài của xâu P, thoả mãn điều kiện $0 < c \leq 1$.
- Xác định sự tồn tại của dãy con chung của P và S có độ dài lớn hơn hoặc bằng $c.l(P)$ với $0 < c \leq 1$ nếu có, không cần chỉ ra dãy con cụ thể.

3. PHƯƠNG PHÁP LẮC BA LÔ

Để trình bày phương pháp này, trước hết ta nhắc lại một số khái niệm và kết quả trong [7].

Với một xâu X tuỳ ý cho trước, ta gọi số lần xuất hiện các ký tự trong X là *độ dài của* X, ký hiệu là $l(X)$. Ký hiệu $X[i]$ hay x_i và A_X, theo thứ tự là *chữ cái thứ* i $(i \leq l(X))$ và *bảng chữ của* X. Các bảng chữ được nói tới đều là hữu hạn, khác rỗng. Lực lượng của bảng chữ A ký hiệu là $|A|$.

Cho xâu mẫu $P = a_1a_2...a_m$ và xâu text $S = b_1b_2...b_n$ trên bảng chữ $A, m, n > 0$. Ta có $l(P) = m$, $l(S) = n$. Một *dãy con* U *của mẫu* P được hiểu là xâu nhận được từ P bằng cách xoá trong P từ 0 đến m ký tự. Nếu U là một dãy con của P và S thì U được gọi là *dãy con chung của* P *và* S. Khi không gây nên nhầm lẫn, ta cũng ký hiệu $l(U)$ là độ dài của dãy con U. Dãy chỉ số tạo bởi các chữ còn lại trong P sau khi xoá được gọi là một *vị trí* của U.

Từ cách xây dựng dãy con, suy ra có thể có nhiều vị trí khác nhau ứng với một dãy con U. Nếu sắp xếp tất cả các vị trí khác nhau của một dãy con U theo thứ tự so sánh trội đồng

thời trên các chỉ số thì phần tử bé nhất được gọi là *vị trí trái nhất của U*, ký hiệu là $LI(U)$. Thành phần cuối cùng bên phải trong $LI(U)$ ký hiệu là $Rm(U)$.

Xét một ví dụ minh hoạ cho các khái niệm nêu trên. Giả sử $P = acbdabad$. Khi đó P có dãy con là $U = abd$ với các vị trí là các dãy chỉ số (1,3,4); (1,3,8); (1,6,8); (5,6,8). Ta có $LI(U)$ = (1,3,4), $Rm(U) = 4 =$ *chỉ số cuối cùng bên phải của LI(abd).*

Để xét tác động trên tập các cấu hình của P sinh bởi các ký tự xuất hiện trong xâu S, ta cần xây dựng một bảng chữ chung cho P và S. Không mất tính tổng quát, có thể thay thế tất cả những ký tự có mặt trong S nhưng không xuất hiện trong P bằng một ký hiệu nào đó không thuộc A_P, chẳng hạn #. Khi đó $A = A_P \cup \{\#\}$ là *bảng chữ chung* của P và S.

Ký hiệu khúc đầu độ dài t của S, $t \leq n$ là S_t, dãy con chung dài nhất của P và S_t là $LCS(P, S_t)$, còn độ dài của $LCS(P, S_t)$ là $L(P, S_t)$.

3.1. Tập các cấu hình

Định nghĩa 1. Một cấu hình của xâu P độ dài $m > 0$ là một tập hợp các dãy con của P, thoả mãn các điều kiện:

1. Tập rỗng $\varnothing$ là một cấu hình của P, ký hiệu là C_0 và gọi là cấu hình khởi đầu.

2. Tập $\varnothing \neq C = \{x_1, x_2, ..., x_k\}$, với $1 \leq k \leq m$, gồm các dãy con đôi một phân biệt và thoả mãn các điều kiện sau:

i) $l(x_i) = i$ (do đó $l(x_i) \neq l(x_j)$ với mọi $0 \leq i \neq j \leq k$ với $k \leq m$),

ii) Với hai dãy con tuỳ ý x_i, x_j thoả mãn $l(x_i) > l(x_j)$ thì $Rm(x_i) > Rm(x_j)$ là một cấu hình của P.

Tập các cấu hình của P được ký hiệu là $CF(P)$.

3.2. Quan hệ bắc cầu và hình chiếu trên tập các cấu hình

Định nghĩa 2. Cho xâu mẫu P độ dài $m > 0$. Trên tập $CF(P)$ ta xác định một quan hệ " $<''$ như sau:

1. Cấu hình rỗng C_0 là phần tử nhỏ nhất (min), nghĩa là với mọi $C \in CF(P)$, hoặc $C_0 < C$ hoặc $C_0 = C$ (ta viết $C_0 \leq C$).

2. Giả sử C_1, C_2 là hai cấu hình tuỳ ý khác C_0.$C_2 < C_1$ (hay nói C_1 trội hơn C_2) nếu thoả mãn các điều kiện:

i) Với mỗi $x \in C_1$, nếu có $y \in C_2$ sao cho $l(x) = l(y)$ thì $Rm(y) \geq Rm(x)$.

ii) Có $x \in C_1$ thoả ít nhất một trong hai điều kiện trội thực sự sau:

+ Nếu có $y \in C_2, l(x) = l(y)$ thì $Rm(y) > Rm(x)$.

+ Không tồn tại $y \in C_2$ sao cho $l(x) = l(y)$.

Ví dụ 1. Cho $P = abcabd$. Ta có $l(P) = 6$, khi đó $C_1 = \{a, ac, bcb, abad\}$ là một cấu hình với $x_1 = a$, $x_2 = ac$, $x_3 = bcb, x_4 = abad$. $C_2 = \{a, ac, bad, abcd\}$ là cấu hình mà $C_2 < C_1$ vì thoả điều kiện 2.i) và có $y_3 = bad \in C_2$ và $x_3 = bcb \in C_1$ thoả 2.ii). Cấu hình C_1 trội hơn cấu hình $C_3 = \{b, bc, bab\}$ theo Điều kiện trong 2.ii).

Giả sử $C \in CF(P)$ với $|P| = m > 0$.

Định nghĩa 3. Hình chiếu của C, ký hiệu là $h(C)$, là một véc tơ m chiều $q \in Z^m$ được xác định theo các quy tắc sau:

1. $h(C_0) = q_0 = (0, 0, ..., 0)$

2. Với $C_0 \neq C = \{x_1, x_2, ..., x_k\}$, giả sử $h(C) = (u_1, u_2, ..., u_m)$ thì $u_i = l(x_t)$ nếu có t $(1 \leq t \leq m)$, sao cho $i = Rm(x_t)$ và $u_i = 0$ nếu trái lại.

Ví dụ 2. Với cấu hình $C_1 = \{a, ac, bcb, abad\}$ ở Ví dụ 1, ta có $q = h(C_1) = (1, 0, 2, 0, 3, 4)$.

3.3. Các tính chất

Từ các Định nghĩa 2 và 3 suy ra một số tính chất sơ cấp sau đây.

1. Quan hệ "<" trên $CF(P)$ có tính chất bắc cầu, không phản xạ và phản đối xứng.

Tuy quan hệ "<" không là quan hệ thứ tự, nhưng được sử dụng để so sánh các dãy con có cùng độ dài nên ta gọi nó là *thứ tự yếu.*

2. Phần tử max là cấu hình có hình chiếu $q = (1, 2, .., m)$.

3. Trong số các cấu hình gồm k dãy con, $k \geq 1$, cấu hình có hình chiếu $q = (1, 2, 3, .., k, 0, 0, .., 0)$ nếu tồn tại, sẽ là cấu hình cực đại (trội nhất).

Tính chất sau đây suy trực tiếp từ định nghĩa, nêu lên tính chất của những thành phần khác 0 của các véc tơ hình chiếu với lưu ý rằng, thứ tự yếu "<" trên các cấu hình liên quan tới các phép so sánh độ dài, toạ độ $Rm()$ của các dãy con có cùng độ dài, và đặc điểm tăng dần của các thành phần khác không của mỗi véc tơ hình chiếu q từ trái sang phải.

4. Cho $q_1 = h(C_1),\ q_2 = h(C_2)$. Giả sử $q_1 = (u_1, u_2, ..., u_m)$ và $q_2 = (v_1, v_2, ..., v_m)$. Khi đó $C_1 < C_2$ khi và chỉ khi thoả các điều kiện sau:

i. Với mọi $1 \leq i,\ k \leq m$, nếu $u_i = v_k$ thì $k \leq i$.

ii. Nếu $u_i,\ v_k$ là hai toạ độ chỉ số cao nhất khác 0 của q_1 và q_2 thì $u_i \leq v_k$.

Hơn nữa, nếu $u_i = v_k$ thì tồn tại $t \leq k$ sao cho $u_t < v_t$. Từ tính chất thứ 4 trên, nếu $q_1 = q_2$ ta có thể gọi C_1 và C_2 là tương đương (đây thực sự là một quan hệ tương đương).

Ví dụ 3. Cho $q_1 = h(C_1),\ q_2 = h(C_2)$, xét một số trường hợp sau:

a) $q_1 = (0, 1, 0, 0, 2, 3)$ và $q_2 = (0, 1, 2, 0, 0, 3)$ thì $C_1 < C_2$.

b) $q_1 = (0, 1, 2, 0, 3, 0)$ và $q_2 = (0, 1, 2, 0, 3, 4)$ thì $C_1 < C_2$.

c) $q_1 = (0, 1, 2, 0, 3, 0)$ và $q_2 = (1, 0, 2, 0, 0, 3)$ thì C_1 và C_2 là không sánh được.

3.4. Tác động trên cấu trúc

Trong phần này, ta xét tác động của các xâu xem như các phần tử của vị nhóm tự do A^* sinh bởi bảng chữ A trên tập các cấu hình.

Định nghĩa 4. Cho mẫu P độ dài m và bảng chữ $A = A_P \cup \{\#\}$ với $\# \notin A_P$. Tác động bởi A lên tập $CF(P)$ là ánh xạ $\varphi : CF(P) \times A \longrightarrow CF(P)$, được xác định như sau:

1. $\varphi(C, \#) = C$ với mọi $C \in CF(P)$.

2. $\varphi(C_0, a) = C' = \{a\}$ với mọi $a \in A_P,\ C_0$ là cấu hình khởi đầu.

3. Với mọi $a \in A_P$, mọi cấu hình $C \neq C_0$, giả sử $C' = \varphi(C, a)$. Khi đó C' được xác định theo các bước sau:

i) Đặt C' bằng C,

ii) Biến đổi C' bởi một vòng lặp theo biến j giảm từ k đến 0 như sau:

a. Với $j = k$, trên P từ trái sang phải nếu có xuất hiện của chữ a tại vị trí có chỉ số lớn hơn $Rm(x_k)$ thì bổ sung vào C' dãy con mới $x_k a$.

b. Với j từ $k-1$ tới 1, trên P từ trái sang phải nếu có xuất hiện của chữ a trong khoảng

$(Rm(x_j),\ Rm(x_{j+1}))$ thì thay x_{j+1} bằng $x_j a$, trái lại ta giữ nguyên x_{j+1}.

c. Với $j = 0$, nếu có xuất hiện của a ở vị trí bên trái $Rm(x_1)$ thì thay x_1 bằng a.

Có thể mở rộng φ trên vị nhóm tự do A^* sinh bởi A.

Từ Định nghĩa 4, ta có các nhận xét sau đây:

1. Điều kiện (1) cho thấy vì sao có thể thay thế tất cả các ký tự xuất hiện trong S nhưng không có mặt trong P bởi ký hiệu # đã nêu trên.

2. Các điều kiện (3.b) và (3.c) cho thấy: với mọi $i,\ i \le k,\ x_i \in C$ và $y_i \in C'$ thoả mãn điều kiện $Rm(y_i) \le Rm(x_i)$.

3. Từ các điều kiện trong Định nghĩa 1, Định nghĩa 4 (3a, 3b, 3c) suy ra tính đúng đắn của tác động φ, nghĩa là C' là một cấu hình.

Ví dụ 4. Xét cấu hình $C_3 = \{b, bc, bab\}$ với $P = abcabd$ như trên. Ta có: $\varphi(C_3, a) = C' = \{a, bc, bca\}$ và $Rm(a) = 1 < Rm(b) = 2,\ Rm(bca) = 4 < Rm(bab) = 5$. $\varphi(C_3, d) = C'' = \{b, bc, bab, babd\}$.

Ý nghĩa của các tác động nêu trong Định nghĩa 4 có thể được giải thích như sau:

Thao tác (a) chỉ ra rằng, tại bước $j = k$ cần chọn chỉ số trái nhất của các xuất hiện trong P của $a \in A_P$ kể từ $Rm(x_k) + 1$ đến m.

Thao tác (b) cho biết tại bước thứ j $(k - 1 \le j \le 1)$ cần chọn chỉ số trái nhất trong các xuất hiện của $a \in A_P$ trong P, kể từ $Rm(x_i) + 1$ đến $Rm(x_{i+1}) - 1$.

Thao tác (c) đòi hỏi tại bước $j = 0$ cần chọn chỉ số trái nhất trong các xuất hiện của $a \in A_P$ trong P, kể từ 1 đến $Rm(x_1) - 1$, nếu có.

Nếu viết các dãy con của cấu hình C từ dưới lên trên, thì dưới tác động của chữ a, vị trí của nó được thay đổi gợi lên hình ảnh như khi *xếp vật a vào ba lô C*, ta *lắc C* để *vật a* rơi dần xuống vị trí sâu nhất có thể. Điều này giải thích cho thuật ngữ *phương pháp lắc ba lô* (Knapsack Shaking Method).

Tính chất sau đây của tác động sẽ được sử dụng khi giải bài toán tìm kiếm xấp xỉ.

Mệnh đề 1. *Cho xâu P và φ là tác động trên các cấu hình của P.*

1. *Nếu $\varphi(C, a) = C'$ thì hoặc $C = C'$ hoặc $C < C'$(C' trội hơn C).*
2. *Nếu $\varphi(C_1, a_1) = C_2,\ \varphi(C_2, a_2) = C_3, ...$ thì với mọi $i < k$, hoặc $C_i = C_k$ hoặc $C_i < C_k$.*

Chứng minh. Trực tiếp suy ra từ Định nghĩa 4. ∎

3.5. Ôtômát liên kết với cấu trúc $CF(P)$ và ôtômát mờ

Về ôtômát mờ chúng tôi sử dụng các khái niệm và định nghĩa như trong [8].

Mỗi trạng thái mờ được xem như một tập mờ trên tập rõ hữu hạn phổ dụng $Q = \{1, 2, .., n\}$, cụ thể như một hàm $f : Q \longrightarrow R$ và được biểu diễn dưới dạng véc tơ rõ $f = (f(1), f(2), ..., f(n))$. Điều này cho phép coi mỗi trạng thái của ôtômát mờ như một véc tơ toạ độ thực, còn phép chuyển trạng thái là phép chuyển véc tơ thành véc tơ. Ở đây, ta coi $f(i) \in N,\ i = 1, 2, ..., n$. Với các ứng dụng khác, có thể xem $f(i)$ là các giá trị thực thuộc đoạn [0, 1].

Định nghĩa 5. Cho mẫu P độ dài m và bảng chữ $A = A_P \cup \{\#\},\ \# \notin A_P$.$(CF(P), <)$ là cấu trúc xác định theo Định nghĩa 4. Ta xây dựng ôtômát mờ khởi đầu $A = (A, Q, q_0, \delta)$, trong đó: A là bảng chữ; Tập trạng thái Q gồm các véc tơ $q = h(C)$ với C là cấu hình đạt được

từ C_0, nghĩa là hoặc $C = C_0$, hoặc có $w = a_1a_2...a_t \in A^*$ sao cho $\delta(C_0, a_1) = C_1,\ \delta(C_1, a_2) = C_2, ..., \delta(C_t, a_t) = C_{t+1} = C;\ q_0 = (0, 0, ..., 0)$ là trạng thái khởi đầu; Hàm chuyển $\delta : Q \times A \longrightarrow Q$ được xác định bởi $\delta(h(C), a) = h(\varphi(C, a))$ với mọi $C \in CF(P)$ và $a \in A$.

Có thể mở rộng δ trên A^* để nhận được tác động của một dãy từ thuộc A^* trên tập trạng thái của ôtômát A.

Về tính đúng đắn của Định nghĩa 5 ta có:

Mệnh đề 2. *Định nghĩa 5 là đúng đắn, xác định một ôtômát với chuyển đơn định chỉ phụ thuộc vào trạng thái q, không phụ thuộc việc chọn cấu hình C mà $h(C) = q$.*

Chứng minh. Phép chứng minh mệnh đề dựa trên bổ đề sau đây. Bản thân bổ đề cũng cho ta những tính chất cần áp dụng đối với phép tác động.

Bổ đề 1. *Cho C_1, C_2 là các cấu hình đạt được từ C_0. Khi đó:*

1. *Nếu $h(C_1) = h(C_2) = q$ thì $C_1 = C_2$.*
2. *Nếu $h(C_1) = h(C_2) = q$ thì $h(\varphi(C_1, a)) = h(\varphi(C_2, a))$.*

3.6. Bài toán tìm chính xác dãy con chung dài nhất

Việc tìm chính xác dãy con chung dài nhất của hai xâu P và S bao gồm:

Bài toán 1. Xác định $LCS(P, S)$.

Bài toán 2. Xác định $L(P, S)$.

Phần này nhắc lại các kết quả dùng làm cơ sở toán học cho hai bài toán trên đã trình bày trong [7].

Bài toán 1 được giải quyết bằng việc xây dựng ôtômát mờ mà mỗi trạng thái là một cấu hình C và áp dụng kết quả sau đây.

Định lý 1. *Cho xâu mẫu P và một xâu text S. Xét ôtômát $\boldsymbol{B} = (A, Q, C_0, \varphi)$ với tập trạng thái là các cấu hình của P với hàm chuyển φ, trạng thái khởi đầu là C_0, còn trạng thái kết thúc là $C_n = \varphi(C_0, S)$. Giả sử $C_n = \{x_1, x_2, .., x_k\}$, trong đó x_k là dãy con với $l(x_k) = \max\{l(x_i),\ i = 1...k\}$ (hay x_k dài nhất trong C_n). Khi đó:*

1. *Với dãy con chung tuỳ ý U của P và S, tồn tại $x_i \in C_n,\ i \geq 1$, sao cho:*

i. $l(U) = i = l(x_i)$.

ii. $Rm(x_i) \leq Rm(U)$.

2. $LCS(P, S) = x_k$.

Bài toán 2 là hệ quả của Bài toán 1, do đó ta chỉ cần quan tâm đến hàm trạng thái δ thông qua định lý sau.

Định lý 2. *Cho xâu mẫu P, $|P| = m$ và xâu text S; $\boldsymbol{A} = (A, Q, q_0, \delta)$ là ôtômát xác định theo Định nghĩa 5, q_0 là trạng thái khởi đầu và trạng thái kết thúc là $q_n = \delta(q_0, S)$. Giả sử $q_n = (n_1, n_2, .., n_k, 0, .., 0)$ với k là chỉ số cao nhất mà $n_k \neq 0$. Khi đó ta có $L(P, S) = n_k$.*

Ví dụ 5. Tìm xâu con chung dài nhất của mẫu $P = acdac$ và xâu $S = cdacda$.

Ta có $l(P) = 5,\ l(S) = 6$. Quá trình tính toán đối với ôtômát có là tập trạng thái là cấu hình và hàm chuyển φ, với dãy tác động S để như sau.

$C_0 = \varnothing;\ \ C_1 = \varphi(C_0, S[1]) = \varphi(C_0, c) = \{c\};$

$C_2 = \varphi(C_1, S[2]) = \varphi(C_1, d) = \{c, cd\};$

$C_3 = \varphi(C_2, S[3]) = \varphi(C_2, a) = \{a, cd, cda\}$;
$C_4 = \varphi(C_3, S[4]) = \varphi(C_3, c) = \{a, ac, cda, cdac\}$;
$C_5 = \varphi(C_4, S[5]) = \varphi(C_4, d) = \{a, ac, acd, cdac\}$;
$C_6 = \varphi(C_5, S[6]) = \varphi(C_5, a) = \{a, ac, acd, acda\}$;
Kết quả là $LCS(P, S) = x_4 = acda$.

Ví dụ 6. Cho $P = uabvcaba$ và $S = abvacbauabvc$. Ta có $l(P) = 8$; $l(S) = 12$; $A_P = \{a, b, c, u, v\} = A$. Ta có:

$q_0 = (0, 0, 0, 0, 0, 0, 0, 0)$;
$q_1 = \delta(q_0, S[1]) = \delta(q_0, a) = (0, 1, 0, 0, 0, 0, 0, 0)$;
$q_2 = \delta(q_1, S[2]) = \delta(q_1, b) = (0, 1, 2, 0, 0, 0, 0, 0)$;
$q_3 = \delta(q_2, S[3]) = \delta(q_2, v) = (0, 1, 2, 3, 0, 0, 0, 0)$;
$q_4 = \delta(q_3, S[4]) = \delta(q_3, a) = (0, 1, 2, 3, 0, 4, 0, 0)$;
$q_5 = \delta(q_4, S[5]) = \delta(q_4, c) = (0, 1, 2, 3, 4, 0, 0, 0)$;
$q_6 = \delta(q_5, S[6]) = \delta(q_5, b) = (0, 1, 2, 3, 4, 0, 5, 0)$;
$q_7 = \delta(q_6, S[7]) = \delta(q_6, a) = (0, 1, 2, 3, 4, 5, 0, 6)$;
$q_8 = \delta(q_7, S[8]) = \delta(q_7, u) = (1, 0, 2, 3, 4, 5, 0, 6)$;
$q_9 = \delta(q_8, S[9]) = \delta(q_8, a) = (1, 2, 0, 3, 4, 5, 0, 6)$;
$q_{10} = \delta(q_9, S[10]) = \delta(q_9, b) = (1, 2, 3, 0, 4, 5, 6, 0)$;
$q_{11} = \delta(q_{10}, S[10]) = \delta(q_{10}, v) = (1, 2, 3, 4, 0, 5, 6, 0)$;
$q_{12} = \delta(q_{11}, S[12]) = \delta(q_{11}, c) = (1, 2, 3, 4, 5, 0, 6, 0)$;
Vậy $L(P, S) = n_6 = 6$.

4. BÀI TOÁN TÌM KIẾM XẤP XỈ

Trong phần này, chúng ta mở rộng các kết quả trên, xây dựng cơ sở toán học cho hai bài toán tìm kiếm xấp xỉ dãy con chung dài nhất nêu ở Mục 2.

4.1. Cơ sở toán học

Như đã đề cập, Định lý 1 là cơ sở toán học cho phương pháp lắc ba lô. Đối với bài toán so mẫu xấp xỉ, chúng ta cần mở rộng như sau.

Định lý 3. *Cho xâu mẫu P và xâu text S. Xét ôtômát mà tập trạng thái là tập cấu hình của P cùng với hàm chuyển φ, trạng thái đầu là C_0. Xét $C_t = \varphi(C_0, S_t)$ với t tuỳ ý, $1 \le t \le n$. Giả sử $C_t = \{x_1, x_2, ..., x_k\}$ trong đó x_k là dãy con x_k dài nhất trong C_t. Khi đó:*

1. Với dãy con chung tuỳ ý U của P và S_t, tồn tại $x_i \in C_n$, $i \ge 1$, sao cho

i. $l(U) = i = l(x_i)$.

ii. $Rm(x_i) \le Rm(U)$.

2. $x_k = LCS(P, S_t)$.

3. Với mỗi hằng số nguyên k cho trước, $1 \le k \le l(P)$, S_t là khúc đầu ngắn nhất của S thoả mãn $L(P, S_t) = k$ khi và chỉ khi t là số nguyên bé nhất sao cho C_t có k dãy con, nghĩa là $C_t = \{x_1, x_2, ..., x_k\}$ với $x_k = LCS(P, S_t)$.

Chứng minh. Phép chứng minh các kết luận (1) và (2) của định lý đã được trình bày trong [7]. Ta còn phải chứng minh (3).

Thật vậy, từ tính đơn định của ôtômát với hàm chuyển φ ta có $\varphi(C_0, S_t) = C_t$, nếu $t \geq 1$ thì $\varphi(C_0, S_{t-1}) = C_{t-1}$ và $\varphi(C_{t-1}, S[t]) = C_t$. Từ đó, trong trường hợp $k = 1$ hay $t = 1$, có thể kiểm tra trực tiếp dựa vào Định nghĩa 4 về phép tác động φ. Còn lại, chỉ cần kiểm tra trường hợp k, $t > 1$. Khi đó, từ Định nghĩa 4 suy ra nếu C_t có dạng $C_t = \{x_1, x_2, .., x_k\}$ với t là chỉ số bé nhất có thể, thì C_{t-1} phải có dạng $\{y_1, y_2, ..., y_{k-1}\}$. Vậy áp dụng (2) suy ra t là chỉ số bé nhất sao cho $L(P, S_t) = k$. Đảo lại, giả sử t là chỉ số bé nhất, $t > 1$ sao cho $L(P, S_t) = k$. Từ (2) và Định nghĩa 4 lại suy ra t là chỉ số bé nhất sao cho C_t gồm k dãy con $x_1, ..., x_k$. ■

Tiếp theo, để trình bày cơ sở toán học cho thuật toán giải Bài toán 2, ta đưa vào khái niệm *vết* của một cấu hình.

Định nghĩa 6. Cho P là xâu khác xâu rỗng, $l(P) = m$ và C là một cấu hình của P. Vết của C, ký hiệu $Tr(C)$, là một véc tơ dạng $Tr(C) = (b_1, b_2, .., b_m)$, trong đó b_i được xác định bởi một trong hai điều kiện sau:

1. Nếu $C = C_0$ thì $b_j = 0$ với mọi j, $1 \leq j \leq m$.

2. Nếu $C = \{x_1, x_2, ..., x_k\}$, $l(x_i) = i$, $1 \leq i \leq k$ thì với mỗi $j = 1, 2, ..., m$, $b_j = 1$ khi và chỉ khi $j = Rm(x_i)$ với $i \leq k$ nào đó.

Từ các định nghĩa, suy ra quan hệ sau đây giữa hình chiếu và vết của một cấu hình.

Bổ đề 2. *Giả sử $h_C(C) = (u_1, u_2, ..., u_m)$ và $Tr(C) = (b_1, b_2, ..., b_m)$. Khi đó $b_i = 1$ khi và chỉ khi $u_i \neq 0$ với mọi i, $1 \leq i \leq m$.*

Thuật ngữ vết nảy sinh từ chính tính chất này. Kết quả sau đây nêu lên mối liên hệ giữa vết và cấu hình.

Mệnh đề 3. *Cho P là xâu khác xâu rỗng trên bảng chữ A và C, C' là hai cấu hình tuỳ ý của P. Ta có:*

1. *$Tr(C) = Tr(C')$ khi và chỉ khi $C = C'$ hoặc C và C' là tương đương, nghĩa là $h(C) = h(C')$.*
2. *Nếu $\varphi(C, a) = C'$ với $a \in A$ nào đó thì $Tr(C) = Tr(C')$ khi và chỉ khi $C = C'$.*

Chứng minh. (1) được suy ra từ Bổ đề 2 và Mệnh đề 1, còn (2) là hệ quả của (1) và Bổ đề 1. ■

Để cài đặt các thuật toán dựa trên khái niệm vết và các tính chất đặc trưng của vết, ta xây dựng hàm tác động α trên các vết. Một vết T là một véc tơ m chiều với tọa độ chỉ gồm 0 hoặc 1.

Với xâu P cho trước và mỗi chữ a, ta quy ước:

Hàm *Right*(a, l) với $l < m$ trả lại giá trị là vị trí xuất hiện trái nhất của a trong khúc con $P[l+1]P[l+2]...P[m]$ nếu có, trái lại sẽ trả lại giá trị 0.

Hàm *Left(a,l)* với $l > 1$ trả lại giá trị trái nhất của a trong khúc con $P[1]P[2]...P[l-1]$ nếu có, trái lại sẽ trả lại giá trị 0.

Hàm *Lm(a,s,t)* với $1 \leq s < t - 1 < m$ trả lại vị trí trái nhất của a trong khúc con $P[s+1]P[s+2]...P[t-1]$ của P nếu có, trái lại trả lại giá trị 0.

Ta nhận xét rằng $Left(a, l) = Lm(a, 0, l)$, $Right(a, l) = Lm(a, l, m + 1)$.

Định nghĩa 7. Cho xâu P, $l(P) = m \geq 1$, A_P là bảng chữ của P và $A = A_P \cup \{\#\}$. Giả sử cho trước vết $T = (b_1, b_2, ..., b_m)$. Với mỗi chữ $a \in A$, vết mới qua tác động α sinh bởi chữ a

lên T, ký hiệu là $\alpha(T,a) = T' = (d_1, d_2, ..., d_m)$ được xác định như sau:

1. Nếu $a \notin A_P (a = \#)$ thì $T' = T$;

2. Nếu $a \in A_P$ thì T' nhận được từ T bởi các bước lần lượt như sau:

i. Khởi đầu $T' = T$;

ii. Xét một biến j từ m giảm dần tới 1. Mỗi trường hợp của j xét:

Nếu j là chỉ số đầu tiên thoả $Right(a,j) > 0$ và $b_j = 1$ thì đặt $b_{Right(a,j)} = 1$;

Nếu có $Lm(a,s,j) > 0$ với $s < j-1$ và $b_j = 1$ và b_s là vị trí kề phải nhất của b_j mà $b_s = 1$ thì đặt $b_{Lm(a,s,j)} = 1$ và $b_j = 0$; { ta nói rằng đã đẩy bj về bên trái}

Nếu b_j là chỉ số trái nhất khác 0, $j > 1$ và $Left(a,j) > 0$ thì đẩy vị trí 1 từ vị trí j về phải tới vị trí $Left(a,j)$, nghĩa là đặt $b_j = 0$ và $b_{Left(a,j)} = 1$;

Ví dụ 7. $P = aabcabd$, $T = (0,1,0,1,0,0,1)$, $T' = \alpha(T,b)$ thì $T' = (0,1,1,0,0,1.0)$.

$T'' = \alpha(T,a)$ thì $T'' = (1,0,0,1,1,0,0)$.

Ví dụ 8. $P = uabvcaba$, $T = (1,1,1,1,1,0,0,0)$, $T' = \alpha(T,v)$ thì $T' = (1,1,1,1,1,0,0,0)$, $T'' = \alpha(T,c) = (1,1,1,1,1,0,0,0)$.

Có thể đối chiếu kết quả này với kết quả tính toán các trạng thái của ôtômát trong Ví dụ 6 (Mục 3.6).

4.2. Các thuật toán tìm kiếm

Phần còn lại, dựa trên các kết quả trình bày, chúng ta xây dựng các thuật toán cho bài toán tìm kiếm xấp xỉ dãy con chung của hai xâu.

Thuật toán 1. Tìm dãy con chung độ dài $\geq k$, k cho trước.

Vào: Xâu mẫu text P trên bảng chữ A_P, $l(P) = m \geq 1$.

Số nguyên dương k, $k \leq m$.

Xâu text S trên bảng chữ $A = A_P \cup \{\#\}$, $l(S) = n$.

Ra: Cho biết có tồn tại dãy con chung của P và S độ dài $\geq k$ không. Nếu có, đưa ra kết quả tìm kiếm.

Nội dung Thuật toán 1:

0. implementation of $\varphi(C,a)$ using Definition 4;
1. initially assign $C := \varnothing$;
2. $j = 1$; {read string S left to right}
3. result := false;
4. while (result = false) and ($j \leq l(S)$) do

begin

a) $a := S[j]$; {get a character of string S}

b) $C_{next} := \varphi(C,a)$;

c) $x :=$ longest sequence in C_{next};

d) if $l(x) \geq k$ then

begin

result := true;

return(x);
end;
e) $j := j + 1$;
f) $C := C_{next}$;
end;
5. announce: 'There isn't';
6. return (' ');

Thuật toán 2. Xác định sự tồn tại của dãy con chung có độ dài $\geq k$.

Vào: Xâu mẫu text P trên bảng chữ A_P, $l(P) = m \geq 1$.
Số nguyên dương k, $1 \leq k \leq m$.
Xâu text S trên bảng chữ $A = A_P \cup \{\#\}$, $l(S) = n$.

Ra: Cho biết có tồn tại dãy con chung của P và S độ dài $\geq k$ không.

Nội dung Thuật toán 2:

0. implementation of $Right(), Left()$ and $Lm()$ functions;
1. implementation of α using functions listed in the step 0;
2. initial assignment $T_0 := (0, 0, .., 0)$;
3. $j := 1$; {read string S left to right}
4. stop := false;
5. while (stop = false) and $(j \leq l(S))$ do
   begin
   a) $a := S[j]$: {get a character of string S}
   b) $T_{next} := \alpha(T, a)$; {next trace}
   c) l:= location of the last element of T_{next} being 1;
   $\{l = \max\{j | b_j = 1\}\}$
   d) if $l \geq k$ then
   begin
   stop := true;
   return (true);
   end;
   e) $j := j + 1$;
   f) $T := T_{next}$;
   end;
6. return (false);

Từ các kết quả trình bày trên, ta có:

Định lý 4. *Thuật toán 1 và Thuật toán 2 là đúng đắn.*

Ta nhận xét rằng, Thuật toán 1 có thể được cải tiến bằng cách ghi nhận các vị trí xuất hiện các dãy con của cấu hình nhờ sử dụng dãy mặt nạ bit cho các dãy con. Từ đó xác định được giá trị $LI()$ của các dãy này và dãy con dài nhất tìm được.

4.3. Đánh giá thuật toán

4.3.1. Độ phức tạp thời gian

Trường hợp xấu nhất xảy ra khi phải thực hiện số lần lặp bằng $l(S) = n$.

Nếu xem thời gian thực hiện một phép chuyển trạng thái $\varphi(C, a) = C'$ là như nhau đối với mọi chữ a xuất hiện trong S và là một hằng λ nào đó, thì theo Định lý 2, độ phức tạp thời gian của Thuật toán 1 có cỡ $n\lambda$. Để xác định λ, ta nhận xét rằng trong trường hợp tổng quát, với mỗi chữ a xuất hiện trong S và $a \in A_P$, số tác động cần xét có cỡ $n|A_P|/|A|$. Tần suất xuất hiện của mỗi chữ trong A_P là $m/|A_P|$. Vì vậy tổng số phép tác động khi đọc xong xâu S là $n(|A_P|/|A|)m/|A| = nm/|A_P|$, nghĩa là $\lambda = m/|A_P|$. Do đó, thời gian thực hiện Thuật toán 1 có cỡ $nm/|A|$. Điều này cũng đúng với Thuật toán 2. Bỏ qua hằng số $m/|A_P|$, ta có độ phức tạp thời gian của Thuật toán 1 và Thuật toán 2 là $O(n)$, nghĩa là tuyến tính đối với n.

4.3.2. Độ phức tạp không gian

Thuật toán 1

Không gian nhớ cho Thuật toán 1 phụ thuộc vào cấu trúc lưu trữ cấu hình C, giúp cho việc tăng tốc độ tính toán hàm φ, tính nhanh hàm $Rm(U)$ và thao tác tìm chữ a trong khoảng $(Rm(x_{i-1}),\ Rm(x_i))$ theo Định nghĩa 4 (3b). Cụ thể là: để lưu trữ C, cần lưu tối đa m dãy con $x_1, ..., x_m$. Vậy không gian nhớ cần sử dụng có cỡ m^2. Gọi giá trị này là M_1. Để tính nhanh hàm $Rm(U)$, giả sử $U = Wa$ (coi $Rm(\varepsilon) = 0$, ε là từ rỗng). Khi đó $Rm(U) = Lm(a, Rm(W), m+1)$. Để tính $Lm(a)$ trong khoảng $(Rm(x_{i-1}), Rm(x_i))$ ta sử dụng hàm $Lm(a, (Rm(x_{i-1}), Rm(x_i))$. Vì thế, sự cài đặt qui về việc xác định cấu trúc dữ liệu phù hợp để tính nhanh hàm $Lm(a, i, j)$. Giả sử bộ nhớ đòi hỏi là M_2. Một cách chi tiết hơn, có thể thấy M_2 có cỡ $m|P|$. Do đó, bộ nhớ cần thiết để lưu trữ mẫu P, cấu hình C và để tính giá trị hàm $Lm()$ là $m + M_1 + M_2$, và như vậy có cỡ m^2. Cấu trúc lưu trữ đoán nhận chữ a có thuộc A_P không có cỡ $|A|$ (bit), xem như một hằng. Vậy độ phức tạp không gian của Thuật toán 1 là $O(m^2)$.

Thuật toán 2

Tương tự như cách tính cho Thuật toán 1, cấu trúc lưu trữ để xác định chữ a có thuộc A_P hay không có cỡ $|A|$(bit). Ta cần lưu trữ 2 véc tơ vết $Tr(C)$ và $sTr(C)$ tiếp theo cho mỗi phép chuyển trạng thái, do đó cần cỡ $2m$ (bit). Trên thực tế chỉ cần lưu trữ một véc tơ rồi biến đổi nó, nghĩa là chỉ cần cỡ m (bit). Để lưu xâu mẫu P cần không gian nhớ cỡ m. Ngoài ra, để lưu trữ dữ liệu cho việc tính nhanh hàm $Lm()$ chúng ta cần không gian nhớ cỡ $m|A_P|$. Các hàm $Left(), Right()$ có thể xem như các trường hợp riêng của hàm $Lm()$. Tổng hợp lại, không gian nhớ cần cho Thuật toán 2 có cỡ $2m + m|A_P| + |A| = m(2 + |A_P|) + |A|$. Vậy, độ phức tạp không gian của Thuật toán 2 là $O(m)$.

Để kết thúc bài báo này, chúng tôi đưa ra một ví dụ minh hoạ ứng dụng của tìm kiếm xấp xỉ. Giả sử cho hai xâu *P= 'given string patterns', S=' string pattern matching in LCS problem'*. Ta hãy coi mỗi dấu cách giữa các từ là một ký hiệu bỏ qua được (don't care letter), mỗi từ của P và S là một 'chữ mới'. Ví dụ *A=given, B=string, C=patterns, C' =pattern ...* Khi đó $P = ABC$, $|P| = 3.S = BC'DEFG$. Tại mỗi bước so các chữ, thay cho giá trị 0,1 ta chấp nhận cả các giá trị trung gian (gọi là độ mờ sánh cặp): tính tỉ số giữa số chữ cái trùng nhau của mỗi cặp từ được so với độ dài của từ dài nhất, nếu tỉ số đó vượt một ngưỡng ε cho trước (chẳng hạn, 0.5) thì lấy, trái lại trả lại giá trị 0. Ví dụ so 'given' với string chỉ có 2 chữ chung là i và n, vậy tỉ lệ là $2/6 < \varepsilon$ nên trả lại giá trị so sánh là 0. Véc tơ khởi đầu là

$q_0 = (0, 0, 0)$, $q_1 = (0, 6/6, 0) = (0, 1, 0)$. Ghi 1 vào một biến nhớ. $q_2 = (0, 0, 6/7) = (0, 0, 0.86)$... Tiếp tục tính toán, ta thấy tổng độ mờ nhận được trên các toạ độ bằng $1 + 0.86 = 1.86$ là độ xấp xỉ lớn nhất đạt được. Như vậy, bằng cách sắp xếp và phân loại dữ liệu cùng với tính toán mờ, chúng ta có thể xử lý hoặc hiển thị thông tin một cách hiệu quả.

Lời cảm ơn

Tác giả xin trân trọng cảm ơn Ban Biên tập Tạp chí Tin học và Điều khiển học về những nhận xét quý báu cho bản thảo, làm cho nội dung bài báo được hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

[1] Alfred V. Aho, *Algorithms for Finding Patterns in String.* Chapter 5, Vol. A, Handbook of Theoretical Computer Science. Elsevier Sciences Publisher BV. 1990.

[2] P.R.J. Asveld, *The Non-Self-Embedding Property for Generalized Fuzzy Context-Free Grammars,* Publ. Math. Debrecen 54 Suppl. (1999) 553–573.

[3] Crochemore, M. and C. Hancart, *Pattern matching in strings, in Algorithms and Theory of Computation* Handbook, M. Atallah (ed.), CRC Press, Boca Raton, 1999.

[4] Chritsian Charras, Thierry Lecroq, *Handbook of Exact String- matching Algorithms,* Book online 2002.

[5] D. Hirschberg, Algorithm for the longest common subsequence problem. *Journal of the Association for Computing Machinery* **24** (1977) 664–675.

[6] Nguyễn Thị Thanh Huyền, Phan Trung Huy, Tiếp cận mờ trong một số bài toán so mẫu, *Tạp chí Tin học và Điều khiển học* **18** (2002) 201–210.

[7] Phan Trung Huy, Nguyễn Quý Khang, ôtômát mờ và ứng dụng trong bài toán tìm dãy con chung dài nhất, *Hội nghị Toán học Toàn quốc lần thứ 6,* Huế 7-10/9/2002. (Abstracts, tr. 96).

[8] Mordeson John N., Malik Davender S., *Fuzzy Automata and Languages: Theory and Application,* ISBN 1-58488-225-5 Publication date: 3/25/2002 .

[9] Tao Jiang, Ming Li, On the approximation of shortest common supersequences and longest common subsequences. *SIAM Journal of Computing.* 1995. 24(5:1122-1139).

Nhận bài ngày 20 - 1 -2003