Tạp chí Tin học và Điều khiển học, T.27, S.1 (2011), 1 - 8

THUẬT TOÁN XÁC ĐỊNH TÍNH CHẤT MÃ CỦA NGÔN NGỮ CHÍNH QUY

NGUYỄN ĐÌNH HÂN[1] , HỒ NGỌC VINH[2] , PHAN TRUNG HUY[3] , ĐỖ LONG VÂN[4]

[1]*Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Hưng Yên*
[2]*Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Vinh*
[3]*Khoa Toán - Tin ứng dụng, Trường Đại học Bách khoa Hà Nội*
[4]*Viện Toán học, Viện Khoa học và Công nghệ Việt Nam*

Abstract. We present an extension of the test proposed by Sardinas and Patterson (1953) for deciding whether a set of words is a code. As a consequence, we obtain an effective algorithm that decides in $\mathcal{O}(k)$ time whether a given regular language is a code, where k is the finite index of the syntactic congruence of this language.

Tóm tắt. Trong bài báo này, chúng tôi trình bày một thuật toán mới mở rộng thuật toán Sardinas-Patterson xác định tính chất mã của một ngôn ngữ. Từ đó nhận được một thuật toán với độ phức tạp cỡ $\mathcal{O}(k)$ để nhận biết một ngôn ngữ chính quy cho trước là mã hay không, với k là chỉ số hữu hạn của tương đẳng cú pháp thỏa ngôn ngữ đó.

1. MỞ ĐẦU

Trong lý thuyết mã, bài toán xác định tính chất mã của ngôn ngữ là một bài toán cơ bản được đề cập trong nhiều công trình nghiên cứu. Năm 1953, Sardinas và Patterson đề xuất một phương pháp tính toán tổ hợp kiểm tra tính chất mã của một tập các từ cho trước. Cho đến nay, phương pháp này vẫn được sử dụng rộng rãi như một tiêu chuẩn kiểm tra tính chất mã của ngôn ngữ. Một số công trình nghiên cứu theo hướng mở rộng phương pháp đó cho các lớp mã mới như mã zigzag (M. Anselmo [1], và D.L. Van, B. Lesaëc, I. Litovsky [2]), Pcodes (F. Blanchet-Sadri, M. Margaret [3]), T-V codes (F.L. Ţiplea và các tác giả khác [4]), mã với từ định biên (H.N. Vinh, P.T. Huy, Đ.L. Vân [5]) và mã nhị phân (D. Macro [6]), hoặc để tính toán một số độ đo của mã như độ không nhập nhằng (P.T. Huy, N.Đ. Hân, P.M. Chuẩn [7]) và độ trễ giải mã (H.N. Vinh, N.Đ. Hân, P.T. Huy [8]). Một số tác giả đề xuất thuật toán kiểm tra mã kiểu Sardinas-Patterson sử dụng kỹ thuật otomat như M. Robert [9], W. Andreas, H. Tom [10] và J. Falucskai [11]. Khi ngôn ngữ được cho bởi một otomat hữu hạn, thuật toán kiểm tra mã trong [9] có độ phức tạp cỡ $\mathcal{O}(n^2)$, với n là tổng số trạng thái và số phép chuyển của otomat hay cỡ $\mathcal{O}(l^2.k^4)$, với l là số chữ của bảng chữ cái và k là số trạng thái của otomat đó.

Trong bài này, chúng tôi cải biên phương pháp tổ hợp của Sardinas và Patterson, đề xuất một thuật toán kiểm tra mã. Khi cho ngôn ngữ chính quy X được đoán nhận bởi một đồng cấu đại số $\alpha : A^* \to M$, với M là một vị nhóm hữu hạn, ta nhận được một thuật toán kiểm tra X có là mã không. Nhờ sử dụng phương pháp đại số, cho phép đánh giá tính hiệu quả rõ rệt của thuật toán này, với độ phức tạp cỡ $\mathcal{O}(k)$, với k là chỉ số tương đẳng cú pháp của X, hơn hẳn độ phức tạp của thuật toán Sardinas-Patterson, được biết có cỡ $\mathcal{O}(2^{2.k})$.

Trước hết, chúng tôi nhắc lại một số ký hiệu và khái niệm được sử dụng trong bài báo, chi tiết xem trong [12, 13]. Cho A là bảng hữu hạn các chữ cái. A^* là vị nhóm tự do của tất cả các từ hữu hạn sinh bởi A. Từ rỗng được ký hiệu là ε và $A^+ = A^* - \{\varepsilon\}$. Mỗi tập con của A^* được gọi là một ngôn ngữ trên A. Một tập $X \subseteq A^+$ được gọi là *mã* trên A nếu mọi từ w thuộc A^+ có nhiều nhất một cách phân tích thành tích của các từ trong X. Giả sử $X, Y \subseteq A^*$, ta gọi *thương trái* (*thương phải*) của X với Y là ngôn ngữ $Y^{-1}X$ (tương ứng XY^{-1}) được xác định bởi $Y^{-1}X = \{w \in A^* \mid yw \in X, y \in Y\}$ và $XY^{-1} = \{w \in A^* \mid wy \in X, y \in Y\}$. Ký hiệu $u^{-1}X$, Xu^{-1} được sử dụng khi tập $Y = \{u\}$ chỉ có một phần tử.

Cho $X \subseteq A^*$, ta nói rằng X *thỏa bởi đồng cấu vị nhóm* $\alpha : A^* \to M$ nếu tồn tại $B \subseteq M$ sao cho $X = \alpha^{-1}(B)$. Trong trường hợp X là ngôn ngữ chính quy, ta luôn xây dựng được một vị nhóm cú pháp hữu hạn M_X và đồng cấu cú pháp $\alpha_X : A^* \to M_X$ thỏa X. Khi đó, ta gọi $k = |M_X|$ *là chỉ số tương đẳng cú pháp của* X.

2. THỦ TỤC SARDINAS-PATTERSON XÁC ĐỊNH TÍNH CHẤT MÃ CỦA NGÔN NGỮ

Cho tập X của các từ trên bảng chữ A, dựa vào định nghĩa của mã, thủ tục Sardinas-Patterson [12] đưa ra cách tính toán các tập thương để tìm hai phân tích khác nhau của một xâu w bất kỳ trong X, $w \in A^*$. Thủ tục Sardinas-Patterson xác định các tập thương U_i, $i = 0, 1, ...$ được xác định một cách đệ quy như sau:

$$\begin{aligned} U_0 &= X^{-1}X - \{\varepsilon\} \\ U_{i+1} &= U_i^{-1}X \cup X^{-1}U_i, \quad i \geq 0 \end{aligned} \tag{2.1}$$

Tính đúng đắn của thủ tục Sardinas-Patterson dựa trên Định lý 2.1 sau đây:

Định lý 2.1. ([12]). *Tập $X \subseteq A^+$ là mã khi và chỉ khi các tập U_i được xác định theo công thức* (2.1) *không chứa từ rỗng.*

Chứng minh của Định lý 2.1 dựa trên Bổ đề 2.1 sau đây:

Bổ đề 2.1. ([12]). *Cho $X \subseteq A^+$ và $(U_i)_{i\geq 0}$ được định nghĩa theo công thức* (2.1). *Với $\forall i > 0$ và $k \in \{0, ..., i\}$, chúng ta có $\varepsilon \in U_i$ khi và chỉ khi tồn tại một từ $w \in U_i$ và $m, n \geq 0$ sao cho $wX^m \cap X^n \neq \emptyset$, $m + n + k = i$.*

Mệnh đề sau đây khẳng định sự tồn tại của thuật toán Sardinas-Patterson khi X là tập đoán nhận được.

Mệnh đề 2.1. ([12]). *Nếu X là một tập đoán nhận được, thì tập tất cả các U_i ($i \geq 0$) là hữu hạn.*

Nhận xét 2.1. Mệnh đề 2.1 khẳng định Định lý 2.1 cung cấp một thuật toán kiểm tra một ngôn ngữ chính quy X bất kỳ có là mã không.

Ví dụ 2.1. Cho $A = \{a, b\}$ và $X = \{b, abb, abbba, bbba, baabb\}$. Tập X không là mã vì tồn tại từ $w = abbbabbbaabb$, w có hai phân tích khác nhau trong X:

$$w = (abbba)(bbba)(abb) = (abb)(b)(abb)(baabb).$$

Sử dụng thuật toán kiểm tra mã, ta có:

$$U_0 = \{ba, aabb, bba\}\ , \qquad U_1 = \{a, ba, abb\}\ , \qquad U_2 = \{\varepsilon, a, ba, bb, bbba, abb\}.$$

Vì $\varepsilon \in U_2$, suy ra X không là mã.

Nhận xét 2.2. Từ thuật toán xác định tính chất mã của một ngôn ngữ ở trên, ta có nhận xét: Các tập U_i sẽ nhận được sau mỗi bước tính toán nhờ áp dụng hữu hạn các phép $\cup$, $\cap$, $-$, các phép cắt trái, phải. Nếu X là ngôn ngữ chính quy thì theo kết quả kinh điển, tồn tại một vị nhóm hữu hạn P và một đồng cấu vị nhóm $\varphi : A^* \to P$ thỏa X và các tập U_i, $i = 0, 1, ...$ Nghĩa là số các tập U_i không vượt quá số tập con của P (bằng $2^{|P|}$) (xem [12]). Mệnh đề và hệ quả sau đây khẳng định điều này.

Mệnh đề 2.2. *Cho $X, Y \subseteq A^*$ là hai ngôn ngữ chính quy, và hai đồng cấu vị nhóm $\alpha : A^* \to M$ và $\beta : A^* \to N$ lần lượt thỏa X và Y, từ đó ta luôn xây dựng được một đồng cấu (toàn cấu) vị nhóm $\varphi : A^* \to P$, với P là một vị nhóm hữu hạn, thỏa đồng thời cả X và Y.*

Ví dụ 2.2. Cho vị nhóm $U_1 = \{0, 1\}$, với phần tử đơn vị là 1 và phần tử zero là 0, A là bảng chữ hữu hạn khác rỗng và cho ngôn ngữ X thỏa bởi một đồng cấu vị nhóm $\alpha : A^* \to M$, ngôn ngữ $Y = \{\varepsilon\}$ thỏa bởi đồng cấu vị nhóm $\beta : A^* \to U_1$, được xác định bởi: $\beta(\varepsilon) = 1$, $\forall u \in A^*$, $\beta(u) = 0$. Dễ thấy rằng, cả X và Y cùng thỏa bởi toàn cấu $\varphi : A^* \to P$, với $P \subseteq M \times U_1$, được xác định bởi: $\forall u \in A^*$, $\varphi(u) = (\alpha(u), \beta(u))$. Khi đó, P là vị nhóm được sinh bởi $\varphi(u)$, $\forall u \in A^*$. Nếu $|M|$ là hữu hạn thì $|P| \leq 2.k$, với $k = |M|$ là chỉ số tương đẳng cú pháp của X.

Từ Mệnh đề 2.2, ta có Hệ quả 2.1 sau đây.

Hệ quả 2.1. *Cho $X, Y \subseteq A^*$ là hai ngôn ngữ chính quy. Nếu X và Y cùng thỏa bởi toàn cấu vị nhóm $\varphi : A^* \to P$, với P là một vị nhóm hữu hạn, thì φ cũng thỏa mọi $L \in \mathcal{A}(X, Y)$ trong đó $\mathcal{A}$ là lớp ngôn ngữ sinh bởi X, Y nhờ sử dụng hữu hạn các phép $\cup$, $\cap$, $-$, các phép cắt trái, cắt phải.*

Chứng minh. Theo giả thiết $X = \varphi^{-1}(B)$, $Y = \varphi^{-1}(C)$, với $B, C \subseteq P$. Với mọi $x \in X \cup Y$, $\varphi(x) \in B$ hoặc $\varphi(x) \in C$, suy ra

$$X \cup Y = \varphi^{-1}(B \cup C) = \varphi^{-1}(B) \cup \varphi^{-1}(C).$$

Do đó φ thỏa ngôn ngữ $X \cup Y$. Lập luận tương tự với phép $\cap$ và phép $-$.

Theo định nghĩa: $B^{-1}C = \{m \in P \mid \exists b \in B : bm = c \in C\} \Rightarrow X^{-1}Y = \varphi^{-1}(B^{-1}C)$. Do đó φ thỏa ngôn ngữ $X^{-1}Y$. Lập luận tương tự với phép XY^{-1}. ■

Nhận xét 2.3. Nếu $X \subseteq A^+$ là ngôn ngữ chính quy và với mọi tập V_i, $i = 0, 1, ...$ thỏa bởi toàn cấu vị nhóm $\varphi : A^* \to P$, với P là một vị nhóm hữu hạn, và các tập V_i có tính chất $V_0 \subseteq V_1 \subseteq \ldots \subseteq A^*$ thì tồn tại $K_i \subseteq P$ sao cho $V_i = \varphi^{-1}(K_i)$, $i = 0, 1, ...$ Khi đó $K_0 \subseteq K_1 \subseteq \ldots \subseteq P$. Nghĩa là số tập V_i không vượt quá số phần tử của P. Nhận xét này là cơ sở để ta xây dựng thuật toán kiểm tra mã trong phần tiếp theo.

3. THUẬT TOÁN XÁC ĐỊNH TÍNH CHẤT MÃ CỦA NGÔN NGỮ CHÍNH QUY

Dựa trên các nhận xét trên, chúng tôi đề xuất một thủ tục mới để kiểm tra một ngôn ngữ chính quy $X \subseteq A^+$ cho trước có là mã không bằng phương pháp tổ hợp mới để tính toán các tập thương V_i, $i = 0, 1, ...$ một cách đệ quy như sau:

$$\begin{aligned} V_0 &= X^{-1}X - \{\varepsilon\} \\ V_{i+1} &= (V_i^{-1}X \cup X^{-1}V_i) \cup V_i, \quad i \geq 0 \end{aligned} \tag{3.1}$$

Tính đúng đắn của thủ tục dựa trên Định lý 3.1 và các Bổ đề sau đây.

Bổ đề 3.1. *Cho $X \subseteq A^+$ và $(V_i)_{i\geq 0}$ được xác định theo công thức* (3.1)*. Với mọi $i \geq 0$, $z \in A^*$ nếu $z \in V_i$ thì tồn tại $n, m \geq 1, x_1, x_2, ..., x_n, y_1, y_2, ..., y_m \in X$ sao cho*

$$x_1x_2 ... x_nz = y_1y_2 ... y_m,\ x_1 \neq y_1.$$

Chứng minh. Ta chứng minh quy nạp theo i điều khẳng định trên.
+ Với $i = 0$: Từ $z \in V_0$ ta phải chứng minh tồn tại $n, m \geq 1, x_1, x_2, ..., x_n, y_1, y_2, ..., y_m \in X$ sao cho $x_1x_2 ... x_nz = y_1y_2 ... y_m$, $x_1 \neq y_1$.

Từ định nghĩa $V_0 = X^{-1}X - \{\varepsilon\}$, $z \in V_0$, suy ra $z \in X^{-1}X - \{\varepsilon\}$, nghĩa là tồn tại $x_1, y_1 \in X$ sao cho $x_1z = y_1$. Vì $\varepsilon \notin V_0$ suy ra $x_1 \neq y_1$.

Vậy, điều khẳng định đúng với $i = 0$.

+ Bước quy nạp, giả sử điều khẳng định đã đúng với $i = k$, ta chứng minh nó cũng đúng với trường hợp $i = k + 1$.

Với $i = k+1$, từ $z \in V_i$ ta phải chứng minh tồn tại $n, m \geq 1, x_1, x_2, ..., x_n, y_1, y_2, ..., y_m \in X$ sao cho $x_1x_2 ... x_nz = y_1y_2 ... y_m$, $x_1 \neq y_1$.

Theo định nghĩa: $V_{k+1} = (V_k^{-1}X \cup X^{-1}V_k) \cup V_k$. Khi đó, với $z \in V_{k+1}$, suy ra $z \in V_k$ (trường hợp 1) hoặc $z \in V_k^{-1}X$ (trường hợp 2) hoặc $z \in X^{-1}V_k$ (trường hợp 3).

Ta xét các trường hợp như sau:
- Trường hợp 1: $z \in V_k$, theo giả thiết quy nạp, điều khẳng định đúng.
- Trường hợp 2: $z \in V_k^{-1}X$, suy ra tồn tại $z' \in V_k, x \in X$ sao cho $z'z = x$. Vì $z' \in V_k$, theo giả thiết quy nạp, tồn tại $l, s \geq 1, x_1, x_2, ..., x_l, y_1, y_2, ..., y_s \in X$ sao cho

$$x_1x_2 ... x_lz' = y_1y_2 ... y_s,\ x_1 \neq y_1.$$

Nhân z vào hai vế của biểu thức trên, ta có:

$$x_1x_2 ... x_lz'z = y_1y_2 ... y_sz,\ x_1 \neq y_1.$$

Thay $z'z = x$ vào trên ta có:

$$x_1x_2 ... x_lx = y_1y_2 ... y_sz,\ x_1 \neq y_1.$$

- Trường hợp 3: $z \in X^{-1}V_k$, suy ra tồn tại $x \in X, z' \in V_k$ sao cho $xz = z'$. Vì $z' \in V_k$, theo giả thiết quy nạp, tồn tại $l, s \geq 1, x_1, x_2, ..., x_l, y_1, y_2, ..., y_s \in X$ sao cho

$$x_1x_2 ... x_lz' = y_1y_2 ... y_s,\ x_1 \neq y_1.$$

Thay $z' = xz$ vào biểu thức trên, ta có:

$$x_1x_2 ... x_lxz = y_1y_2 ... y_s,\ x_1 \neq y_1.$$

Vậy, điều khẳng định đúng với mọi $i \geq 0$. ■

Bổ đề 3.2. *Cho $X \subseteq A^+$ và $(V_i)_{i\geq 0}$ được xác định theo công thức* (3.1)*. Với mọi $x_1, x_2, ..., x_n$, $y_1, y_2, ..., y_m \in X$, $n, m \geq 1$ và $\forall z \in A^*$: $|z| < |y_m|$, nếu $x_1x_2 ... x_nz = y_1y_2 ... y_m$, $x_1 \neq y_1$ thì tồn tại $i \geq 0$ sao cho $z \in V_i$.*

Chứng minh. Ta chứng minh quy nạp theo n điều khẳng định trên.
+ Với $n = 0$, từ dãy $x_1z = y_1y_2 ... y_m$, $|z| < |y_m|$, $x_1 \neq y_1$ ta phải chứng minh tồn tại $i \geq 0$ sao cho $z \in V_i$. Theo giả thiết $x_1 \neq y_1$, ta xét 2 trường hợp sau:
- Trường hợp 1: $|x_1| < |y_1|$. Theo giả thiết $x_1 \neq \varepsilon$, $|z| < |y_m|$, suy ra $m = 1$. Khi đó, ta có thể viết $x_1z = y_1$, $x_1 \neq y_1$. Vậy, $z \in (X^{-1}X - \{\varepsilon\}) = V_0$.
- Trường hợp 2: $|x_1| > |y_1|$. Vì $|z| < |y_m|$, suy ra tồn tại $u \in A^+$ sao cho $y_m = uz$. Từ đẳng thức $x_1z = y_1y_2 ... y_m$, $x_1 \neq y_1$, suy ra $x_1z = y_1y_2 ... y_{m-1}uz$, $x_1 \neq y_1$. Khi đó:

$$y_1^{-1}x_1 = y_2y_3 ... y_{m-1}u \in V_0$$

$$\Rightarrow\ y_2^{-1}V_0 \ =\ y_3 \dots y_{m-1}u \in V_1$$
$$\dots$$
$$\Rightarrow\ y_{m-1}^{-1}V_{m-2} \ =\ u \in V_{m-1},\ \Rightarrow\ u^{-1}y_m \ =\ z \in V_m$$

Vậy điều khẳng định đúng với $n = 0$.

+ Giả sử điều khẳng định đúng với $n = k$, ta chứng minh nó cũng đúng với $n = k + 1$. Từ đẳng thức $x_1x_2 \dots x_kx_{k+1}z = y_1y_2 \dots y_m$, $x_1 \neq y_1$, ta chứng minh tồn tại $i \geq 0$ sao cho $z \in V_i$.

Xét xâu $x_{k+1}z$, ta có ba trường hợp sau xảy ra:

- Trường hợp 1: $|x_{k+1}z| < |y_m|$, suy ra $x_{k+1}z \in V_i$ (theo quy nạp). Do đó, $x_{k+1}^{-1}V_i = z \in V_{i+1}$

- Trường hợp 2: $x_{k+1}z = y_m$, suy ra $x_{k+1}^{-1}y_m = z$. Nếu $z \neq \varepsilon$ thì $z \in V_0 = (X^{-1}X - \{\varepsilon\})$. Ngược lại, nếu $z = \varepsilon$ thì $\varepsilon \in V_i$ nào đó.

Thật vậy, theo giả thiết ta có $x_1x_2 \dots x_{k+1}z = y_1y_2 \dots y_m$, $x_1 \neq y_1$. Vì $x_{k+1}z = y_m$, suy ra $x_1x_2 \dots x_k = y_1y_2 \dots y_{m-1}$, $x_1 \neq y_1$ hay $x_1x_2 \dots x_k\varepsilon = y_1y_2 \dots y_{m-1}$, $x_1 \neq y_1$. Theo giả thiết quy nạp, suy ra $\varepsilon \in V_i$ nào đó.

- Trường hợp 3: $|x_{k+1}z| > |y_m|$. Theo giả thiết $|z| < |y_m|$, suy ra tồn tại $u \in A^+$ sao cho $y_m = uz$. Từ $|x_{k+1}z| > |y_m|$ và $y_m = uz$, suy ra tồn tại $u' \in A^+, y_s \in X$, $1 \leq s \leq m-1$, $|u'| < |y_s|$ sao cho $x_{k+1} = u'y_{s+1}y_{s+2} \dots y_{m-1}u$.

Hơn nữa theo giả thiết quy nạp $x_1x_2 \dots x_kx_{k+1}z = y_1y_2 \dots y_m$, $x_1 \neq y_1$. Thay $y_m = uz$ và $x_{k+1} = u'y_{s+1}y_{s+2} \dots y_{m-1}u$ vào trên ta có: $x_1x_2 \dots x_ku'y_{s+1}y_{s+2} \dots y_{m-1}uz = y_1y_2 \dots y_{m-1}uz$, $x_1 \neq y_1$. Suy ra $x_1x_2 \dots x_ku' = y_1y_2 \dots y_s, x_1 \neq y_1$.

Theo quy nạp, suy ra $u' \in V_i$ với i nào đó. Do đó, ta có :

$$u'^{-1}x_{k+1} \ =\ y_{s+1}y_{s+2} \dots y_{m-1}u \in V_{i+1} = V_i^{-1}X$$
$$\Rightarrow\ y_{s+1}^{-1}V_{i+1} \ =\ y_{s+2} \dots y_{m-1}u \in V_{i+2}$$
$$\dots$$
$$\Rightarrow\ y_{m-1}^{-1}V_{i+m-1} \ =\ u \in V_{i+m} \quad \Rightarrow \quad u^{-1}y_m \ =\ z \in V_{i+m+1}.$$

Vậy điều khẳng định đúng với mọi $n \geq 0$. ■

Định lý 3.1. *Cho $X \subseteq A^+$ và $(V_i)_{i\geq 0}$ được xác định theo công thức* (3.1)*. Khi đó, X là mã khi và chỉ khi $\varepsilon \notin V_i$, với mọi $i \geq 0$.*

Chứng minh. ($\Rightarrow$) X là mã thì $\varepsilon \notin V_i$ với mọi $i \geq 0$.

Ta chứng minh bằng phản chứng. Giả sử tồn tại $i \geq 0$ sao cho $\varepsilon \in V_i$. Khi đó, theo Bổ đề 3.1, tồn tại $n, m \geq 1, x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m \in X$ sao cho $x_1x_2 \dots x_n\varepsilon = y_1y_2 \dots y_m$, $x_1 \neq y_1$ hay $x_1x_2 \dots x_n = y_1y_2 \dots y_m$, $x_1 \neq y_1$, đây là hai phân tích khác nhau của một từ trong X. Suy ra X không là mã, mâu thuẫn.

($\Leftarrow$) Ta chứng minh $\varepsilon \notin V_i$ với mọi $i \geq 0$ thì X là mã.

Ta chứng minh bằng phản chứng. Giả sử X không là mã. Khi đó, tồn tại $w \in A^*$ có hai phân tích: $w = x_1x_2 \dots x_n = y_1y_2 \dots y_m$, $x_1 \neq y_1$, $x_i, y_j \in X$, $i = 1, .., n$, $j = 1, .., m$. Hay ta có thể viết $x_1x_2 \dots x_n\varepsilon = y_1y_2 \dots y_m$, $x_1 \neq y_1$, $x_i, y_j \in X$. Vì $|y_j| > |\varepsilon|$, $j = 1, .., m$, suy ra tồn tại $i \geq 0$ sao cho $\varepsilon \in V_i$ (theo Bổ đề 3.2), mâu thuẫn. Vậy, X là mã. ■

Hệ quả 3.1. *Cho $X \subseteq A^+$ và $(V_i)_{i\geq 0}$ được xác định theo công thức* (3.1)*. Nếu $V_i = \emptyset$ thì X là mã.*

Chứng minh. Ta chứng minh nếu $V_i = \emptyset$ thì X là mã. Thật vậy, theo cách xác định V_i như trên, nếu có $V_i = \emptyset \ \Rightarrow\ V_{i-1} = \emptyset \ \Rightarrow\ \dots \ \Rightarrow\ V_0 = \emptyset$ hay $V_i = \emptyset$, với mọi $i \geq 0$. Do đó, ta có $\varepsilon \notin V_i$, với mọi $i \geq 0$. Theo Định lý 3.1, suy ra X là mã. ■

Hệ quả 3.2. *Cho $X \subseteq A^+$ và $(V_i)_{i\geq 0}$ được xác định theo công thức* (3.1)*. Nếu tồn tại $i \geq 0$ sao cho $V_{i+1} = V_i$ và $\varepsilon \notin V_s$, với $i \geq s \geq 0$ thì X là mã.*

Chứng minh. Ta phải chứng minh nếu $\varepsilon \notin V_s$, với mọi $s \leq i, V_{i+1} = V_i$ thì X là mã. Thật vậy, theo định nghĩa:

$$V_{i+2} = (V_{i+1}^{-1}X \cup X^{-1}V_{i+1}) \cup V_{i+1}$$

Nếu có $V_{i+1} = V_i$, thì thay V_{i+1} bởi V_i ta có:

$$V_{i+2} = (V_i^{-1}X \cup X^{-1}V_i) \cup V_i = V_{i+1} = V_i$$

Tương tự, ta có $V_n = V_i$, với mọi $n \geq 0$. Vì $\varepsilon \notin V_s$, suy ra $\varepsilon \notin V_n = V_s$, với mọi $i \geq s \geq 0$. Do đó, ta có $\varepsilon \notin V_s$, với mọi $i \geq s \geq 0$. Theo Định lý 3.1, suy ra X là mã. ■

Với một ngôn ngữ bất kỳ thuộc lớp ngôn ngữ chính quy, số các tập V_i, $i = 0, 1, ...$ xác định như trên là hữu hạn. Từ thủ tục, ta nhận được một thuật toán mới để kiểm tra một ngôn ngữ chính quy có là mã hay không:

*** Thuật toán ESP (the Extension of Sardinas and Patterson Algorithm)**

Input: Cho $X \subseteq A^+$, X là ngôn ngữ chính quy.

Output: Kết luận X là mã hoặc không.

Bước_ 1. $V_0 = X^{-1}X - \{\varepsilon\}$, $n = 0$
If $V_0 = \emptyset$ *Then goto Bước_ 4*

Bước_ 2. (*Loop*)
$V_{n+1} = (V_n^{-1}X \cup X^{-1}V_n) \cup V_n$

Bước_ 3. *If* $\varepsilon \in V_{n+1}$ *Then goto Bước_ 5*
If $V_n = V_{n+1}$ *Then goto Bước_ 4*
Else $n = n + 1$, *Loop*

Bước_ 4. *Thông báo "X là mã" và Kết thúc.*

Bước_ 5. *Thông báo "X không là mã" và Kết thúc.*

Nhận xét 3.1. Cho X là ngôn ngữ chính quy thỏa bởi đồng cấu đại số $\alpha : A^* \to M$, với M là một vị nhóm hữu hạn, và ngôn ngữ $Y = \{\varepsilon\}$. Theo Hệ quả 2.1, $|\mathcal{A}(X, Y)|$ có cỡ $2^{2.k}$ và $|P|$ có cỡ $2.k$, với k là chỉ số tương đẳng cú pháp của X. Khi đó, thuật toán Sardinas-Patterson sẽ cho câu trả lời khẳng định sau tối đa $2^{2.k}$ bước, với mỗi bước là một bước tính U_{i+1} từ U_i theo công thức (2.1) (là số tối đa các tập $U_i \subseteq \mathcal{A}(X, Y)$ khác nhau, $\forall i \geq 0$). Thuật toán mở rộng ESP được đề xuất trên đây đưa ra câu trả lời với số bước tối đa chỉ là $2.k$, với mỗi bước là một bước tính V_{i+1} từ V_i theo công thức (3.1).

Ví dụ 3.1. Cho $A = \{a, b, c, d\}$, $X = \{a, ab, bc, cb, abd\}$. Theo thuật toán Sardinas-Patterson, ta có: $U_0 = \{b, bd, d\}$, $U_1 = \{c\}$, $U_2 = \{b\}$, $U_3 = \{c\}$. Vì $U_3 = U_1$, suy ra X là mã. Theo thuật toán ESP, ta có: $V_0 = \{b, bd, d\}$, $V_1 = V_2 = \{b, c, bd, d\}$. Vì $V_2 = V_1$, suy ra X là mã.

Ví dụ 3.1 minh họa một trường hợp đặc biệt, do tính chất bao hàm của các tập V_i, thuật toán ESP dừng sau khi tính V_2, còn thuật toán Sardinas-Patterson phải tiếp tục tính U_3 vì các tập U_i khác nhau, $2 \geq i \geq 0$.

Tổng quát, theo thuật toán Sardinas-Patterson, các tập U_i sẽ được xác định nhờ áp dụng hữu hạn các phép $\cup$, $\cap$, $-$, các phép cắt trái, cắt phải. Theo Hệ quả 2.1, φ thỏa tất cả các tập U_i, nghĩa là số các tập U_i không vượt quá số tập con của P (bằng $2^{2.k}$). Trong trường hợp xấu nhất, thuật toán kiểm tra tính chất mã theo Sardinas-Patterson có độ phức tạp cỡ hàm mũ $\mathcal{O}(2^{2.k})$ theo số bước đã nêu.

Bây giờ ta sẽ xem xét thuật toán ESP với các tập V_i, $i = 0, 1, ...$ được định nghĩa theo công thức (3.1). Vì V_i thỏa bởi φ, đặt $V_0 = \varphi^{-1}(K_0)$, $V_1 = \varphi^{-1}(K_1)$, . . . , $V_i = \varphi^{-1}(K_i)$, trong đó $K_i \subseteq P$. Vì $V_0 \subseteq V_1 \subseteq \ldots \subseteq V_i \subseteq A^*$ nên ta có khẳng định sau đây:

Nếu các tập V_i khác nhau với mọi i, khi đó: $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_i \subseteq P$. Số tập K_i là hữu hạn và không vượt quá $|P|$ tập, suy ra số các tập V_i không vượt quá $|P|$ tập. Ngược lại, nếu có sự lặp lại của các tập V_i thì số tập K_i khác nhau như trên và không vượt quá $|P|$ tập. Do đó, số các bước thực hiện để tính toán các tập V_i là hữu hạn và không vượt quá $|P|$ bước.

Như vậy, ta có thể kết luận: nếu ta chỉ quan tâm đến số các tập V_i khác nhau như trong thuật toán Sardinas-Patterson thì thuật toán ESP có độ phức tạp cỡ $\mathcal{O}(|P|) \approx \mathcal{O}(k)$.

Mặt khác, nếu độ phức tạp của mỗi bước tính V_{i+1} từ V_i kể đến từng phép tính trên vị nhóm P, có cỡ $(|P|.|P| + |P|.|P|)$ và xây dựng vị nhóm P đòi hỏi tối đa $|P|^3$ phép tính trên vị nhóm M nhờ thuật toán loang dần, thì độ phức tạp của thuật toán kiểm định mã ESP theo số bước và các phép tính cơ sở trên vị nhóm P trong trường hợp xấu nhất là $\mathcal{O}(|P|^3) \approx \mathcal{O}(k^3)$. Lưu ý rằng nếu X cho bởi một otomat hữu hạn có k trạng thái, thuật toán kiểm định mã theo [9] có độ phức tạp cỡ $\mathcal{O}(l^2.k^4)$ như đề cập trong phần mở đầu. Điều này cho thấy vai trò cải tiến của thuật toán ESP.

Ví dụ 3.2. Cho $A = \{a, b\}$, $X = \{aa, baa, ba\}$. Theo thuật toán trên ta có:

$V_0 = \{a\}$, $V_1 = \{a\}$. Ta có $V_0 = V_1$, suy ra X là mã.

Ví dụ 3.3. Cho $A = \{a, b, c\}$, $X = \{ba, bac, cb\}$. Theo thuật toán trên ta có:

$V_0 = \{c\}$, $V_1 = \{c, b\}$, $V_2 = \{c, b, a, ac\}$,

$V_3 = \{c, b, a, ac\}$,

Ta có $V_2 = V_3$, suy ra X là mã.

Ví dụ 3.4. Cho $A = \{a, b\}$, $X = \{aa, aab, baa, baab\}$. Theo thuật toán trên ta có:

$V_0 = \{b\}$, $V_1 = \{b, aa, aab\}$, $V_2 = \{\varepsilon, b, aa, aab\}$

Vì $\varepsilon \in V_2$, suy ra X không là mã.

Ví dụ 3.5. Cho $A = \{a, b\}$, $X = \{a^*b\}$. Ta có: $V_0 = \emptyset$, suy ra X là mã.

4. KẾT LUẬN

Trong bài báo, chúng tôi giới thiệu chi tiết thuật toán xác định tính chất mã của một ngôn ngữ bất kỳ theo hướng mở rộng thuật toán Sardinas-Patterson truyền thống. Kết quả áp dụng đối với lớp ngôn ngữ chính quy cho thấy sự cải tiến khi đưa ra được thuật toán chỉ có độ phức tạp tuyến tính thay vì độ phức tạp cỡ hàm mũ. Kết quả này sẽ được phát triển trong nghiên cứu lý thuyết mã.

TÀI LIỆU THAM KHẢO

[1] M. Anselmo, On zigzag codes and their decidability, *Theory Computer Science* **74** (1990) 341-354.

[2] D.L. Van, B. Lesaëc and I. Litovsky, On coding morphisms for zigzag codes, *Informatique Théorique et Applications* **26** (6) (1992) 565-580.

[3] F. Blanchet-Sadri and M. Margaret, *Pcodes of Partial Words*, Retrieved Jan 2011, from University of North Carolina, Department of Mathematical Sciences, 2005.

[4] F.L. Ţiplea, E. Mäkinen, D. Trincă, and C. Enea, Characterization Results for Time-Varying Codes, *Fundamenta Informaticae* **53** (2) (2002) 185-198.

[5] Hồ Ngọc Vinh, Phan Trung Huy, Đỗ Long Vân, Mở rộng mã và thuật toán kiểm định mã luân phiên và mã của các từ định biên *Tạp chí Tin học và Điều khiển học* **26** (4) (2010) 301 - 311.

[6] D. Macro, "New techniques for Signal Representation and Coding", Doctoral dissertation No. ING-INF/03, Universitas Studiorum Brixiæ, 2006.

[7] Phan Trung Huy, Nguyễn Đình Hân và Phạm Minh Chuẩn, Mã luân phiên chẵn - Phân bậc độ nhập nhằng và tiêu chuẩn kiểm tra, *Kỷ yếu Hội thảo quốc gia lần thứ XII, Một số vấn đề chọn lọc của công nghệ thông tin và truyền thông*, Biên Hòa, 5-6/8, 2009 (171-185).

[8] Hồ Ngọc Vinh, Nguyễn Đình Hân, Phan Trung Huy, Mã với tích biên và độ trễ giải mã *Tạp chí Công nghệ Thông tin và Truyền thông* **V-1** (4) (24) (2010) 46-56.

[9] M. Robert, An O(n^2) Time Algorithm for Deciding Whether a Regular Language is a Code, *Journal of Computing and Information* **2** (1) (1996) 79-89.

[10] W. Andreas and H. Tom, The finest homophonic partition and related code concepts. *Lecture Notes in Computer Science* **841** (1994) 618-628.

[11] J. Falucskai, On equivalence of two tests for codes, *Acta Mathematica Academiae Paedagogicae Nyíregyháziensis* **24** (2008) 249-256.

[12] J. Berstel, D. Perrin and C. Reutenauer, *Theory of Codes*, Academic Press Inc., NewYork, 1985.

[13] G. Lallement, *Semigroups and Combinational Applications*, John Wiley and Sons, Inc., 1979.

Ngày nhận bài 24 - 2 - 2011