MỤC LỤC

[TÓM TẮT NỘI DUNG 2](#_Toc324514604)

[CHƯƠNG I – MỞ ĐẦU 3](#_Toc324514605)

[CHƯƠNG II - BÀI TOÁN TÌM KIẾM KHÚC CON CHUNG CỦA HAI XÂU 5](#_Toc324514606)

[2.1. Cơ sở lý thuyết otomat hữu hạn 6](#_Toc324514607)

[2.2. Các thuật toán 8](#_Toc324514608)

[CHƯƠNG III - THIẾT KẾ CHƯƠNG TRÌNH 19](#_Toc324514609)

[KẾT LUẬN 24](#_Toc324514610)

[HƯỚNG PHÁT TRIỂN 24](#_Toc324514611)

[TÀI LIỆU THAM KHẢO 25](#_Toc324514612)

# TÓM TẮT NỘI DUNG

So sánh mẫu và tìm kiếm mẫu là bài toán quan trọng trong lĩnh vực tìm kiếm database, xử lý văn bản nói riêng và ứng dụng trong CNTT truyền thông nói chung.

Bài toán được quan tâm giải quyết trong báo cáo khoa học của chúng tôi được phát biểu như sau: Cho 1 tập các xâu mẫu Pi được sắp xếp theo một biểu thức động (biểu thức logic hoặc chính quy)…Với mỗi một xâu S đã cho, hãy xác định tập các xâu có cấu trúc như trên có thỏa mãn với S hay không? Nhu cầu tìm kiếm động xuất hiện nhiều trong các công cụ tìm kiếm nâng cao, tuy nhiên đến nay còn phức tạp và cần được nâng cao chất lượng tìm kiếm.

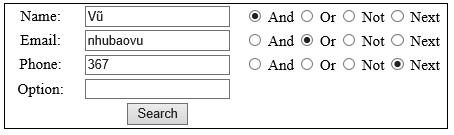
Trong báo cáo này chúng tôi xin trình bày một phương pháp tìm kiếm mềm dẻo (có chấp nhận lỗi) và xây dựng cấu trúc tìm kiếm động (logic hoặc chính quy) để giải quyết bài toán trên dựa vào kĩ thuật Otomat và kĩ thuật xử lý biểu thức động nhằm đáp ứng các tìm kiếm phức tạp, tích hợp vào các hệ thống quản lý dữ liệu.

# CHƯƠNG I – MỞ ĐẦU

Trong những năm gần đây, các bài toán liên quan đến tìm kiếm luôn nhận được sự quan tâm đặc biệt của giới khoa học. Các thuật toán trong tìm kiếm như : Brute – Force , KMP , BM … được phát triển , tuy nhiên các thuật toán nguyên thủy trên không đáp ứng được về việc tìm kiếm xấp xỉ và hơn nữa thời gian tìm kiếm dựa trên các thuật toán kể trên chưa tốt . Việc phát sinh tìm kiếm xấp xỉ được đưa ra và nảy sinh ra các giải thuật mới . Để tìm kiếm xấp xỉ thành công , chúng ta phải xây dựng được được một khái niệm “tương tự ” giữa hai xâu . Tương tự ở đây có thể hiểu là giữa hai xâu kí tự có một vài sai khác ở một số vị trí , chẳng hạn “dân trí” và “dan trí” … Trên thế giới có đề xuất các giải thuật khác nhau như : xâu con chung lớn nhất , dãy con chung dài nhất …. Và nhiều phiên bản biến thể từ hai thuật toán trên.

Một ví dụ tiêu biểu về việc tìm kiếm xấp xỉ là từ khóa “like “ trong SQL giúp tìm kiếm xấp xỉ những xâu có được coi là tương tự với xâu mẫu .

Hiện tại dạng form tìm kiếm của các trang web thường như sau:



Với thiết kế như trên thì số option và liên kết thường là cố định, dẫn đến việc xây dựng source code cũng khá là “cứng”. Khi có nhu cầu cần thay đổi thì thường phải xây dựng lại .

Trong bài báo cáo này, chúng tôi xin đề xuất giải pháp dùng kỹ thuật automat và giải thuật KMPM để thực hiện việc tìm kiếm xấp xỉ.

Việc xây dựng giải thuật của chúng tôi là khá mềm dẻo và cho phép người dùng tự điều chỉnh option để tăng việc lựa chọn tìm kiếm chính xác.

Thêm nữa, về thời gian tìm kiếm, do giải thuật chúng tôi lựa chọn là automat nên thời gian tìm kiếm là tuyến tính, đáp ứng được nhu cầu về thời gian .

# CHƯƠNG II - BÀI TOÁN TÌM KIẾM KHÚC CON CHUNG CỦA HAI XÂU

**2.1. Phát biểu bài toán**

**\*** Bài toán 1: Cho xâu mẫu P và xâu đích S trên bảng chữ A, bài toán yêu cầu ta tìm ra một hoặc nhiều vị trí xuất hiện của mẫu trên một văn bản. Trong đó mẫu và văn bản là các chuỗi có độ dài N và M (M ≤ N), tập các ký tự được dùng gọi là bảng chữ cái A.

Cho xâu mẫu P = **abc** và xâu đích S = bac**abc**de. Câu hỏi đặt ra là: với xâu mẫu P và xâu đích S như trên trong S có xuất hiện (chính xác) mẫu P hay không ?

Kết quả: **abc** có xuất hiện trong S tại vị trí 3.

**\*** Bài toán 2: Là bài toán mở rộng của bài toán 1: Cho một xâu đích S và tập các xâu mẫu Pi( i = 1..n), các Pi có quan hệ với nhau theo một biểu thức logic hay biểu thức chính quy nào đó.

Tại đây với các yêu cầu từ người dùng là động như vậy ta phải xây dựng được một biểu thức mẫu tìm kiếm là động cho phép người dùng được lựa chọn cho từng yêu cầu cụ thể. Vậy yêu cầu bài toán là tìm các xuất hiện của tập xâu mẫu Pi trong S thỏa mãn biểu thức động (logic hay chính quy) đã cho.

VD1 : Giả sử ta mong muốn tìm 1 văn bản K có nội dung liên quan đến “văn hóa” và “giáo dục”. Nếu đặt P1=“văn hóa” và P2=“giáo dục” như vậy biểu thức chính quy là:P1 Λ P2. Ở đây, ta sẽ thực hiện tìm tất cả các văn bản (xâu đích) có cả hai mẫu “văn hóa” và “giáo dục”.

VD2 : Giả sử chúng ta muốn tìm 1 văn bản về “giá vàng” và không phải về “màu vàng”. Nếu đặtP1=“giá vàng” và P2=“màu vàng” thì ta có biểu thức logic là: P1 Λ(­­­¬P3).

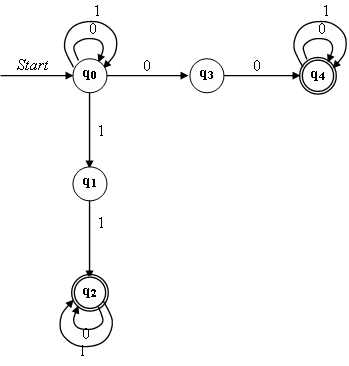
**\*** Bài toán 3: Là bài toán thực tế: tương tự bài toán 2, nhưng cho phép tìm kiếm xấp xỉ các Pi.

Bài toán yêu cầu tìm một hoặc nhiều vị trí xuất hiện của tập xâu mẫu Pi trong S thỏa mãn biểu thức chính quy đã cho.

VD1 : Khi muốn tìm chữ “hà nội” trong 1 file văn bản S. Ta sẽ tìm kiếm xấp xỉ “hà nội ” trong file văn bản S , chẳng hạn “La noi ”…

## 2.1. Cơ sở lý thuyết otomat hữu hạn

Cho bảng chữ cái hữu hạn . Otomat hữu hạn không tất định trênlà bộ , được tạo thành từ tập hữu hạn các *trạng thái*, *trạng thái khởi đầu*, tập con là các *trạng* thái *kết thúc* và tập gọi là tập các cung, xác định *hàm chuyển trạng thái* của Otomat.



Mỗi cung là một bộ , ta nói  có nhãn , đi từ trạng thái  đến trạng thái , và thường được kí hiệu là 

Một đường đi hữu hạn  trong otomat  là chuỗi hữu hạn các cung liên tiếp , trong đó,  là độ dài của đường đi, từ tạo thành bởi các nhãn  trên các cung được gọi là nhãn của đường đi. Trạng thái  là điểm đầu và trạng thái  là điểm cuối của đường đi. Ta thường kí hiệu 

Đường đi  với  được gọi là đường đi thành công. Tập hợp tất cả các nhãn của các đường đi thành công của otomat  được kí hiệu là  và thường được gọi là *ngôn ngữ chính qui đoán nhận bởi*. Một tập được gọi là *tập đoán nhận được* nếu có otomat  sao cho .

Otomat  được gọi là tất định nếu với các cung  tùy ý thuộc , từ  và  suy ra . Điều này có nghĩa là với mỗi trạng thái  và chữ cái , có đúng một trạng thái q sao cho . Khi  là tất định, ta định nghĩa một hàm , xác định bởi với , để đơn giản ta kí hiệu 

Hàm  được mở rộng thành hàm: , được gọi là hàm chuyển mở rộng của otomat  xác định trên mỗi từ , với định nghĩa qui nạp theo độ dài của  và , qui ước  được sử dụng theo nghĩa mở rộng.

## 2.2. Các thuật toán

**Thuật toán KMP**

Thuật toán so khớp chuỗi Knuth-Morris-Pratt (KMP) là một trong những thuật toán tìm sự xuất hiện của một xâu P có trong một xâu văn bản S. Sẽ tiếp tục quá trình tìm kiếm khi không phù hợp, chính mẫu P cho ta các thông tin để xác định vị trí bắt đầu của kí tự so sánh tiếp theo, do đó bỏ qua quá trình kiểm tra lại các kí tự đã khớp với mẫu trước đó.

#### *Ý tưởng thuật toán*

Sau mỗi lần dịch mẫu, đối với thuật toán chưa có tinh chỉnh (thô) đã quên toàn bộ thông tin về các kí tự đã khớp trước đó. Vì vậy, cho một lần so khớp tiếp theo nó sẽ so sánh lại chuỗi văn bản với mẫu một lần nữa và cứ như vậy. Điều này dẫn đến độ phức tạp của thuật toán là O(nm) với n – độ dài của chuỗi văn bản S; m – độ dài của mẫu P.

Thuật toán Knuth, Morris and Pratt [KMP 77] sử dụng lại các thông tin các so sánh khớp đã thu được trước đó. Nó không thực hiện so sánh lại các kí tự đã xuất hiện trong mẫu cho một lần so khớp mới tiếp theo. Kết quả là độ phức tạp tìm kiếm của thuật toán KMP là O(n).

Tuy nhiên, phương pháp cần một giai đoạn tiền xử lý phân tích cấu trúc của mẫu P. Pha xử lý này có độ phức tạp là O(m). Với , như vậy độ phức tạp của thuật toán vẫn là O(n).

**2.4. Giải thuật tìm kiếm xấp xỉ**

**2.4.1. Otomat so khớp mẫu chính xác**

Bài toán phát biểu: Cho xâu mẫu P và xâu đích S. Độ mờ xuất hiện mẫu P trên S tại vị trí j nào đó là một giá trị nguyên a thỏa mãn :

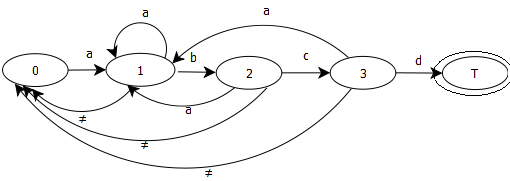
* a = 0 nếu Sj≠ P1
* a là số lớn nhất sao cho =

Khi đó khi đọc mỗi kí tự Sj nào đó của S ta sẽ có được ngay độ mờ xuất hiện mẫu. Do ý nghĩa của độ mờ là độ dài khúc đầu dài nhất của mẫu P đã xuất hiện trên S nên otomat sẽ có tập trạng thái là tập số nguyên {0, 1,..., m}. Hoạt động của otomat mờ so mẫu chính xác sẽ như sau:

* Khởi đầu con trỏ trên S là j = 0. Tại đó chưa xuất hiện khúc đầu nào của mẫu nên trạng thái khởi đầu của otomat là q0 = 0.
* Duyệt S, mỗi lần một kí tự, bắt đầu từ S1
* Giả sử trạng thái của otomat là q thì khi đọc được kí tự Sj, trạng thái mới (ứng với vị trí j trên S) sẽ là q” = Ɍ(q,Sj) (Ɍ là hàm chuyển của otomat).
* Tại vị trí j trên S, nếu trạng thái của otomat là q, có nghĩa khúc đầu dài nhất xuất hiện trên S của P có độ dài q. Nếu q = m, báo hiệu một lần xuất hiện mẫu, bắt đầu từ vị trí j -m+1.Mô hình otomat mờ cần được xây dựng một cách thích hợp để đápứng được yêu cầu sánh mẫu như trên.

Ví dụ: Cho một xâu mẫu P: **abcd** và xâu đích S: **gaabcd**

Ta có otomat biểu diễn trạng thái tìm kiếm xâu mẫu P trong S như hình dưới đây:

**

**Hình 1. Otomat tìm kiếm mẫu chính xác**

Định nghĩa: Otomat mờ so mẫu là bộ A(P) = (A, Q, q0, σ, F) trong đó:

+ Bảng chữ vào A = AP U {#}

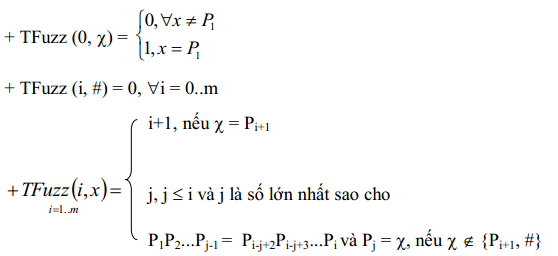
+ Tập trạng thái Q = {0, 1, ..., m}

+ Trạng thái khởi đầu q0 = 0

+ Trạng thái kết thúc F = m.

+ Hàm chuyển σ: 

σ (q,a) = TFuzz (q,a), với hàm TFuzz được xác định như sau:

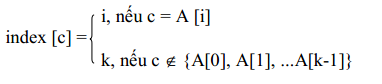


Việc xác định j dựa vào bảng next và một vòng lặp.

**Thuật toán KMP mờ:**

Khi cài đặt thuật toán cần lưu ý lựa chọn cấu trúc dữ liệu phù hợp để có thể truy nhập nhanh chóng trong bảng TFuzz.

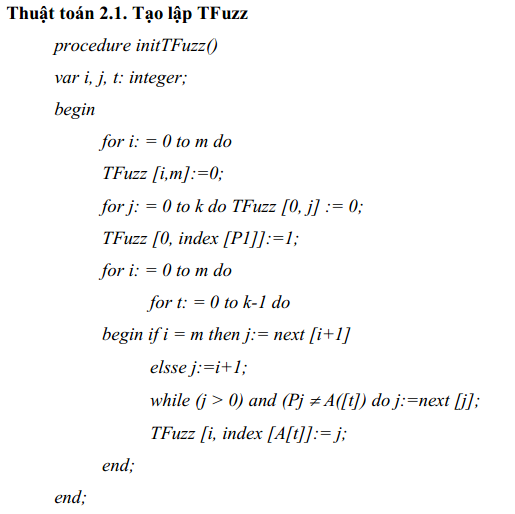
Gọi A[0..k] là mảng lưu giữ bảng chữ A của otomat, trong đó k là số kí tự phân biệt trong mẫu P. Màng được sắp theo chiều tăng của các kí tự và A[k] = ‘#’. Để thuận tiện khi truy nhập đến các chữ cái trong A, có thể sử dụng mảng index xác định vị trí của chữ trong bảng.



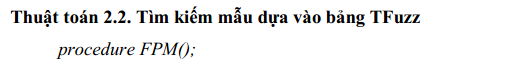
TFuzz là mảng [0..m, 0..k], trong đó TFuzz [i, j] là độ mờ mới khi độ mờ i gặp kí tự x có index [] = j.

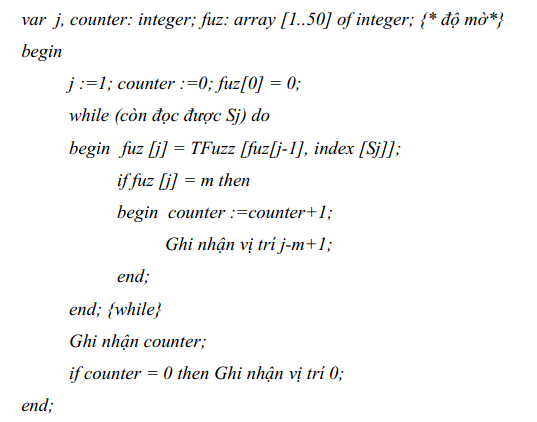
Khi đó chi tiết thuật toán tạo lập bảng TFuzz và tìm kiếm dựa vào bảng TFuzz sẽ như sau:

**Thuật toán: Tạo lập TFuzz**



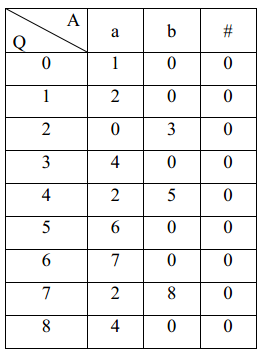
**Thuật toán: Tìm kiếm mẫu dữa vào bảng TFuzz**



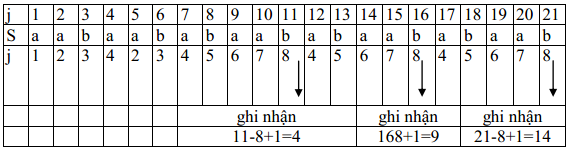


**Ví dụ**: Cho mẫu P = aababaab, A = {a, b, #}, Ap = {a,b}.

Bảng TFuzz được tính toán dựa trên mảng next cho kết quả như sau:



Quá trình so mẫu trên dòng dữ liệu S = aabaababaababaababaab sẽ như sau:

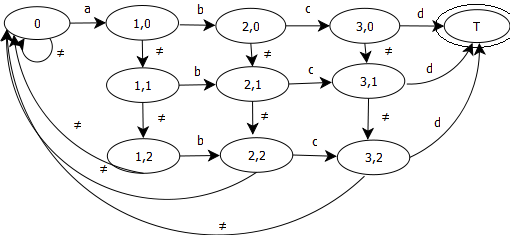


**2.4.2. Otomat nâng cao tìm kiếm mẫu xấp xỉ**

1. *Otomat cải tiến dành cho tìm kiếm xấp xỉ*

Kỹ thuật Otomat này có thêm cải tiến là lưu thêm *trạng thái tích lũy lỗi*. Do đó, kỹ thuật này cho phép ta có thể bỏ qua lỗi để tìm kiếm xấp xỉ.Tuy nhiên để đảm bảo không ra quá nhiều kết quả tìm kiếm không như mong muốn, ta có thêm *mức tích lũy* hay *ngưỡng phạm sai* cho phép giới hạn lỗi phạm sai.

Ví dụ: ta có mức tích lũy là 2 và tổng các lỗi phạm sai không quá 4.

**

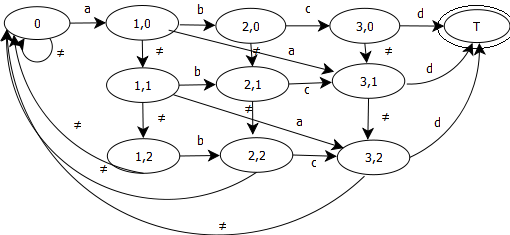
**Hình 2. Otomat tìm kiếm mẫu xấp xỉ**

* Ưu điểm: Tránh được LCS(Longest common subsequence). Thời gian giải thuật là tuyến tính cỡ O(n), với n là độ dài xâu văn bản S.
* Nhược điểm: Số trạng thái bùng nổ, số cung lưu trữ nhiều
* Môi trường tìm kiếm: văn bản, hệ thống, cơ sở dữ liệu.

1. *Trường hợp khác của mẫu P*

Theo a) thì mẫu P được coi là chính xác và có lỗi là cố định, một trường hợp khác là lỗi được bơm vào P một đơn vị , giả sử  là do lỗi người dùng nhập mẫu tìm kiếm bị sai theo lối chèn thêm kí tự khác làm kích thước mẫu tăng không bình thường.

Ví dụ mẫu P = “th.ông tư” có kí tự ‘.’ Chèn vào giữa. Như vậy, lúc này Otomat trạng thái tìm kiếm mẫu P được thiết kế như hình sau:



**Hình 3. Lỗi được bơm thêm vào mẫu P**

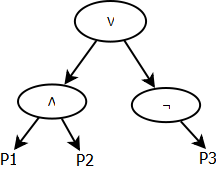
1. *Tìm kiếm mẫu biểu thức logic, chính quy*

Từ những yêu cầu tìm kiếm động của người dùng hoặc theo một bài toán thực tế tìm kiếm động nào đó. Cây biểu thức động tìm kiếm được thiết kế theo các bước sau và phương pháp giải.

**Bước 1** : Từ các mẫu đơn Pi ta xây dựng cây logic T cho biểu thức .

VD : Giả sử ta có 1 biểu thức như sau : P1 Λ P2 v (­­­¬ P3)

Cây biểu thức logic sẽ có dạng như sau :



**Hình 4. Cây biểu thức động T (P1 , P2 ,P3)**

**Bước 2**: Duyệt cây chuyển thành biểu thức hậu tố:

C:\Users\at\Desktop\a1.png

**Bước 3**: Nhập xâu S và sinh ra các otomat tương ứng nhận biết cho Pi

Với cây T => phát sinh 3 Otomat A1, A2, A3

A1 đón nhận P1

A2 đón nhận P2

A3 đón nhận P3.

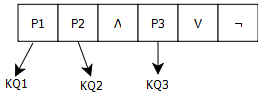
**Bước 4** : Kiểm tra S có phù hợp với cây T hay không?

Giải pháp thô :

Sử dụng A1 so mẫu P1 trên S ta được kết quả True hoặc False lưu trên con trỏ KQ1.

Sử dụng A2so mẫu P2trên S ta được kết quả True hoặc False lưu trên con trỏ KQ2.

Sử dụng A3so mẫu P3trên S ta được kết quả True hoặc False lưu trên con trỏ KQ3.



Giả sử: KQ1=True ,KQ2=False, KQ3=False

C:\Users\at\Desktop\3.png

KQ= KQ1 Λ KQ2 v (­­­¬ KQ3)= True Λ False v True = True

Với bài toán này nếu KQ=True => phù hợp nên S được trả về kết quả

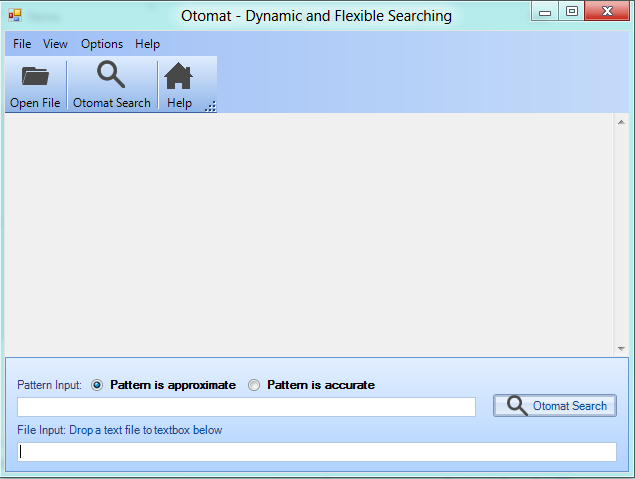
KQ=False=> không phù hợp nên S không được trả về

# 

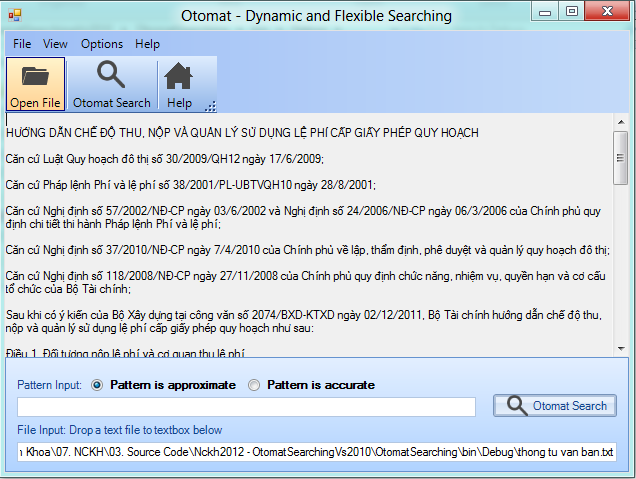
# CHƯƠNG III - THIẾT KẾ CHƯƠNG TRÌNH

Dựa phương pháp tìm kiếm động bằng phương pháp Otomat đề xuất trên, chúng em đã xây dựng thành công chương trình demo tìm kiếm thông tin trên mẫu chính xác và trên mẫu xấp xỉ văn bản tiếng việt có dấu theo phương pháp xây dựng Otomat.Mục đích xây dựng chương trình nhằm kiểm tra thuật toán đề xuất. Từ đó có thể xây dựng, phát triển cho ra sản phẩm có chất lượng mang tính ứng dụng thực tế.

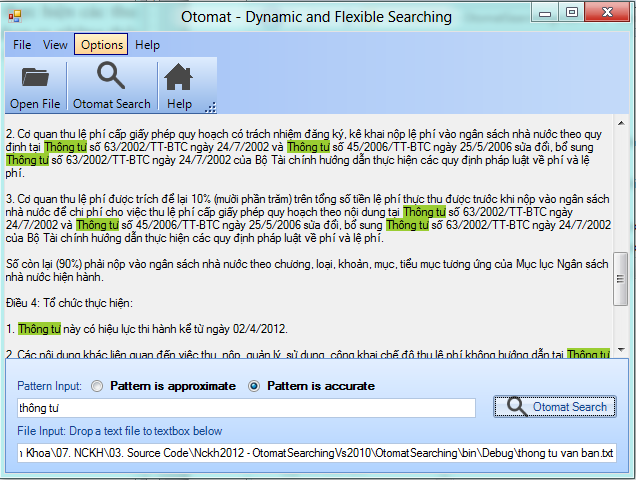
Chương trình được xây dựng trên ngôn ngữ Visual C#, có giao diện đơn giản và thân thiện. Để thực hiện tìm kiếm nội dung trong file văn bản. Người dùng chỉ việc thao tác kéo thả file dữ liệu vào giao diện chương trình hoặc chọn chức năng mở file trên thanh công cụ. Dữ liệu được load và người dùng cần nhập mẫu văn bản tìm kiếm vào ô mẫu tìm kiếm Pattern Input. Theo trên giao diện người dùng sẽ có 2 lựa chọn hoặc là tìm kiếm mẫu xấp xỉ theo phương pháp Otomat động (mặc định) hoặc tìm kiếm mẫu chính xác. Sau khi hoàn tất nhập liệu và tùy chỉnh tham số chương trình, chọn nút Otomat Search để thực hiện việc tìm kiếm.



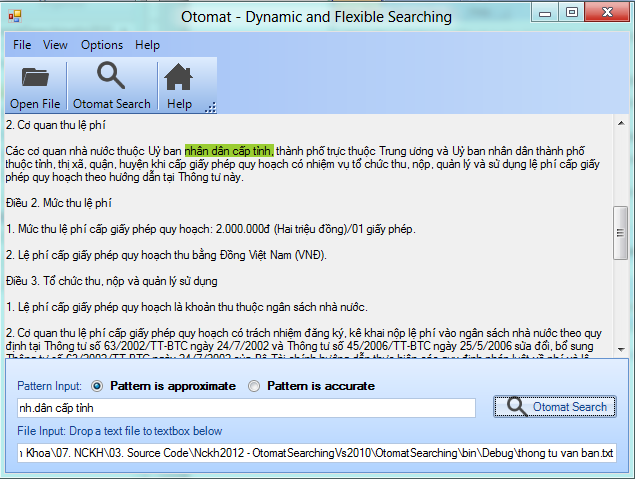
Hình 5. Giao diện chính chương trình



Hình 6. Load dữ liệu từ tệp văn bản



Hình 7. Kết quả tìm kiếm mẫu chính xác



Hình 8. Kết quả tìm kiếm mẫu có xấp xỉ

Kết quả chạy chương trình, ta thấy ngay được khả năng tìm kiếm được văn bản với mẫu đầu vào có thể là mẫu chính xác hoặc có xấp xỉ. Và người sử dụng được khuyến khích nhập các mẫu tìm kiếm được viết tắt để có thể rút ngắn được thời gian tìm kiếm do việc nhập liệu.

Ví dụ sử dụng trong ưng dụng, với mẫu P = “**nh.dân cấp tỉnh**”. Và ta thu được kết quả chính xác với nội dung “**nhân dân cấp tỉnh**”.

# KẾT LUẬN

Báo cáo đã tìm hiểu về otomat mờ, một số thuật toán tìm kiếm và giới thiệu hai bài toán so mẫu xấp xỉ - chính xác bằng phương pháp dùng KMP Mờ và giải thuật automat. Trình bày những thuật toán so mẫu của hai bài toán trên đều dựa vào độ tương tự giữa hai xâu theo một mô hình "lỗi" kinh điển.

- Trình bày được tổng quan về tìm kiếm mẫu trên văn bản, từ đó đưa ra các dạng tìm kiếm mẫu.

Một số hạn chế của luận văn:

Chưa tối ưu hóa được chương trình một cách tốt nhất.

Trình bày báo cáo còn đôi chỗ lủng củng.

Mới tiến hành cài đặt trên môi trường window , chưa cài đặt thử với được với môi trường mạng .

# HƯỚNG PHÁT TRIỂN

* Tối ưu hóa lại chương trình
* Tiến hành xây dựng tìm kiếm trên môi trường mạng và tích hợp phép tìm kiếm vào 1 số hệ quản trị như SQL …..
* Do thời gian và khả năng có hạn nên trong báo cáo còn có nhiều hạn chế , mong thầy cô và các bạn góp ý để nhóm chúng em có thể hoàn thiện hơn được chương trình .

# TÀI LIỆU THAM KHẢO

[1] Alfred V. Aho, *Algorithms for Finding Pattern in Strings*, Chapter V, Handbook of Theoretical Computer Science, Vol. A, Jan Van Leeuwen, Algorithms and Complexity, The MIT Press., 1990, pp 257-299 .

[2] Christian Charras, Thierry Lecroq, *Hanbook of Exact String- matching Algorithms* , http:// www-igm.univ-mlv.fr/~lecroq/string/index.html .

[3] Nguyễn Thị Thanh Huyền, Phan Trung Huy, *Tiếp cận mờ trong một số thuật toán so mẫu*, Tạp chí Tin học và điều khiển học, số 3, tập 18 (2002) .

[4] Nguyễn Thị Thanh Huyền, Bùi Kiên Cường, Phan Trung Huy, *Các thuật toán tìm kiếm xâu con và tìm kiếm tựa ngữ nghĩa có sử dụng Otomat mờ* , Kỷ yếu Hội thảo quốc gia lần thứ VI về “Một số vấn đề chọn lọc của Công nghệ thông tin”, Thái Nguyên, 2003.