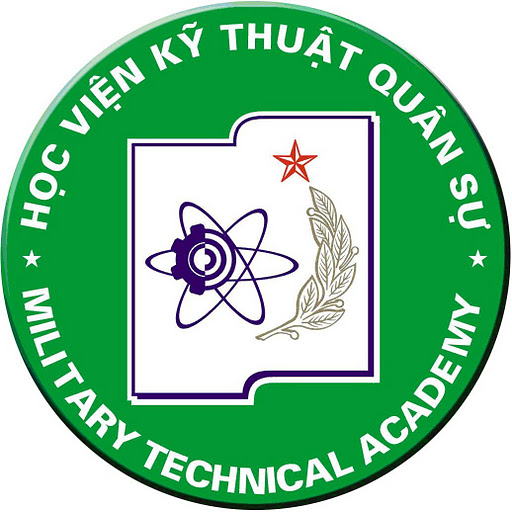
**HỌC VIỆN KỸ THUẬT QUÂN SỰ**



**Khoa Công Nghệ Thông Tin**

****

**Báo cáo môn**

**Lý thuyết Ngôn Ngữ Lập Trình**

**Người thực hiện:** Nguyễn Văn Lộc

**Lớp:** Kỹ thuật phần mềm 14

**Giảng viên hướng dẫn:** HÀ CHÍ TRUNG

**Đề tài:** Mô phỏng thuật toán phân tích cú pháp CYK(Cocke-Younger-Kasami)

**Ngày hoàn thành:** 08/06/2018

Mục Lục

[**1.** **Đặt vấn đề** 3](#_Toc516227947)

[**1.1** **Mục đích của việc giải quyết bài tập được giao** 3](#_Toc516227948)

[**1.2** **Nêu các vấn đề cần giải quyết** 3](#_Toc516227949)

[**1.2.1** **Phân tích cú pháp là gì?** 3](#_Toc516227950)

[**1.2.2** **Các phương pháp phân tích cú pháp** 3](#_Toc516227951)

[**2.** **Kiến thức lý thuyết** 4](#_Toc516227952)

[**2.1** **Văn phạm** 4](#_Toc516227953)

[**2.2** **Thuật toán CYK** 4](#_Toc516227954)

[**3.** **Cấu trúc chương trình** 5](#_Toc516227955)

[**4.** **Sơ đồ khối các bước giải quyết công việc** 6](#_Toc516227956)

[**5.** **Giao diện chương trình và các kết quả đánh giá** 6](#_Toc516227957)

[**6.** **Phụ lục** 8](#_Toc516227958)

[7. **Tài liệu tham khảo**: 8](#_Toc516227959)

[Xem tại đây 8](#_Toc516227960)

1. **Đặt vấn đề**
   1. **Mục đích của việc giải quyết bài tập được giao**

* Tăng cường hiểu biết về lý thuyết ngôn ngữ lập trình,  
  chương trình dịch, tăng cường kỹ năng lập trình.
* Nắm được kiến thức về thuật toán CYK và kiến thức quan trọng, rất có ích trong việc phân tích cú pháp.
* Làm bài tập lớp để kết thúc học phần Lý thuyết ngôn ngữ lập trình.
  1. **Nêu các vấn đề cần giải quyết**
     1. **Phân tích cú pháp là gì?**
* **Phân tích cú pháp(parsing)** là quá trình xác định xem liệu chuỗi ký hiệu kết thúc(token) có thể được sinh ra bởi một văn phạm hay không?
  + 1. **Các phương pháp phân tích cú pháp**
* Phân tích từ trên xuống(Top – Down Parsing): Phương pháp này bắt đầu từ trên xuống(bắt đầu bởi nút gốc), nhãn là ký hiệu chưa kết thúc bắt đầu và lặp lại việc thực hiện hai bước sau:

B1: Tại nút n, nhãn là ký hiệu chưa kết thúc A, chọn một trong những luật sinh của A và xây dựng các con của n cho các ký hiệu trong vế phải của luật sinh

B2: Tìm nút kế tiếp mà tại đó một cây con sẽ được xây dựng. Đối với một số văn phạm, các bước trên được cài đặt bằng một phép quyets(scan) dòng nhập từ trái qua phải.

* Phân tích từ dưới lên(bottom-up): quá trình này ngược lên với phân tích top-down
* Có nhiều ngôn ngữ lập trình hỗ trợ phân tích cú pháp như Java, C#....
* Trong bài này em sử dụng ngôn ngữ c# để thực hiện làm về phân tích cú pháp.

1. **Kiến thức lý thuyết**
   1. **Văn phạm**

* Ngữ pháp hay văn phạm là quy tắc chủ yếu trong cấu trúc ngôn ngữ. Việc tạo ra các quy tắc chính cho một ngôn ngữ riêng biệt là ngữ pháp của ngôn ngữ đó, vì vậy mỗi ngôn ngữ có một ngữ pháp riêng biệt của nó. Ngữ pháp là một phần trong nghiên cứu ngôn ngữ hay còn gọi là ngôn ngữ học.
  1. **Thuật toán CYK**
* Nhằm khắc phục hạn chế của các phương pháp top-down(văn phạm không có đệ quy trái… ) và bottom-up(văn phạm không có suy dẫn rỗng và không có ký hiệu đệ quy… ). Vì vậy người ta sử dụng thuật toán CYK để khắc phục những hạn chế đó, thuật toán CYK chấp nhận văn phạm có suy dẫn rỗng.
* Ý tưởng thuật toán:
  + Giả thiết chuỗi w = w1w2…wn
  + Ta định nghĩa tập Xij là tập tất cả các ký hiệu có thẻ suy diễn ra chuỗi con wiwi+1…wi+j-1 (chuỗi con bắt đầu từ wi và có độ dài j)
  + Bài toán đoán nhận S ⇒\* w tương đương với việc trả lời S có thuộc tập X1n hay không?
  + Vấn đề bây giờ là tính Xij như thế nào?

//tính X của các chuỗi độ dài 1

for (int i = 1; i <= n; i++) X[i,1] = { A | A → wi }

// tính X của các chuỗi độ dài 2,3,…,n

for (int j = 2; j <= n; j++)

for (int i = 1; i <= n-j+1; i++) {

X[i,j] = {}

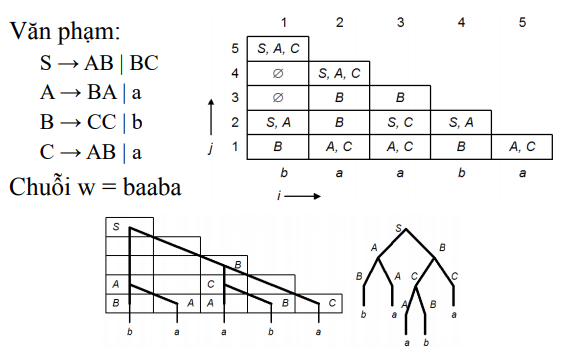
for (int k = 1; k <= j-1; k++) X[i,j] += {

A | nếu A → BC | và B thuộc X[i,k] | và C thuộc X[i+k,j-k]

}

}

* Ví dụ:



* Đánh giá thuật toán:
  + Hạn Chế:
    - Thuật toán không làm việc với suy dẫn rỗng
    - Sô lượng ký hiệu trung gian( non-terminal) nhiều, do việc chuyển đổi từ CFG sang chuẩn Chomsky
  + Độ phức tạp thuật toán(xấu nhất) là O(x|G|)
    - Số n là độ dài chuỗi w
    - |G| là kích thước của văn phạm dạng CNF
  + Bản chất là ý tưởng bottom-up nhưng áp dụng các kỹ thuật quy hoạch động
  + Dễ dàng liệt kê mọi cây phân tích khác nhau và lại bỏ các suy dẫn trùng lặp

1. **Cấu trúc chương trình**

* Ngôn ngữ lập trình sử dụng: C# window form
* Chương trình gồm form main
* Các hàm được sử dụng: [phụ lục 1](Phụ-Lục-I.docx)

1. **Sơ đồ khối các bước giải quyết công việc**

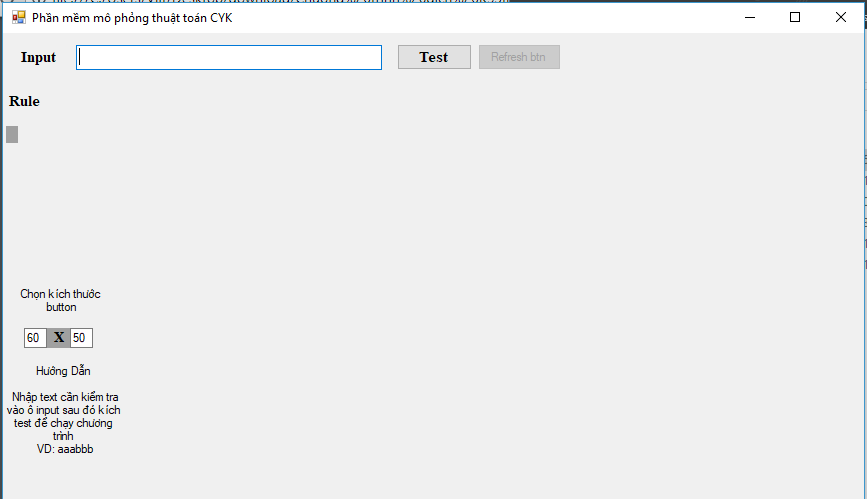
Kiểm tra chuỗi nhập vào với bộ luật đặt ra

Thực thi thuật toán

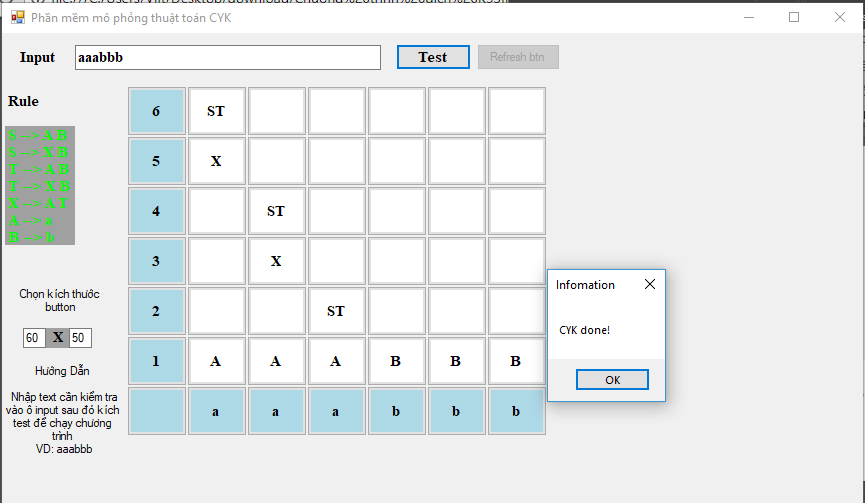
Đưa ra kết quả

1. **Giao diện chương trình và các kết quả đánh giá**

Giao diện chính



Sau khi thực hiện chương trình



Đánh giá chương trình: Chương trình hiện mới chỉ mô phỏng được thuật toán CYK

1. **Phụ lục**

Phụ lục I: Giải thích code và các hàm chi tiết

1. **Tài liệu tham khảo**:

[Xem tại đây](https://txnam.net/wp-content/uploads/txnam/Bai%20Giang/Chuong%20Trinh%20Dich%20K53II/Chuong%20trinh%20dich%20K53II%20-%2012.pdf)