

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



BÁO CÁO BÀI TẬP LẬP TRÌNH LÝ THUYẾT TÍNH TOÁN

Bài tập lập trình 4:

Dạng chuẩn Chomsky và thuật toán CYK

Giảng viên hướng dẫn:

Trần Vĩnh Đức

Lớp:

108529

Sinh viên thực hiện:

Vũ Đình Hoàng - 20161728

Hà Nội 2019

I. Dạng chuẩn Chomsky

1. Đề bài

- Đưa văn phạm về dạng chuẩn Chomsky.
- Input: File **G.jff** ở định dạng JFLAP biểu diễn văn phạm G.
- Output: File **Chomsky.jff** ở định dạng JFLAP biểu diễn văn phạm G' ở dạng chuẩn Chomsky tương đương với văn phạm G.

2. Thuật toán

2.1. Xử lý biến bắt đầu

- Thêm biến bắt đầu mới để biến bắt đầu không xuất hiện ở vế phải của bất cứ quy tắc nào.
- Ta thêm biến mới S_0 và quy tắc $S_0 \rightarrow S$, với S là biến khởi đầu.

2.2. Loại bỏ quy tắc ϵ

- Với mỗi quy tắc có dạng $A \rightarrow \epsilon$:
 - o Loại bỏ quy tắc $A \rightarrow \epsilon$
 - o Nếu có quy tắc dạng $R \rightarrow uAv$, ta thêm quy tắc $R \rightarrow uv$. Thực hiện bước này với mọi sự xuất hiện của A. Ví dụ: Nếu có quy tắc dạng $R \rightarrow uAvAw$ thì ta phải thêm quy tắc
$$S \rightarrow uAvw \mid uvAw \mid uvw$$
 - o Nếu có quy tắc $R \rightarrow A$, ta thêm quy tắc $R \rightarrow \epsilon$ trừ khi ở các bước trước ta đã loại bỏ quy tắc $R \rightarrow \epsilon$.

2.3. Loại bỏ quy tắc đơn

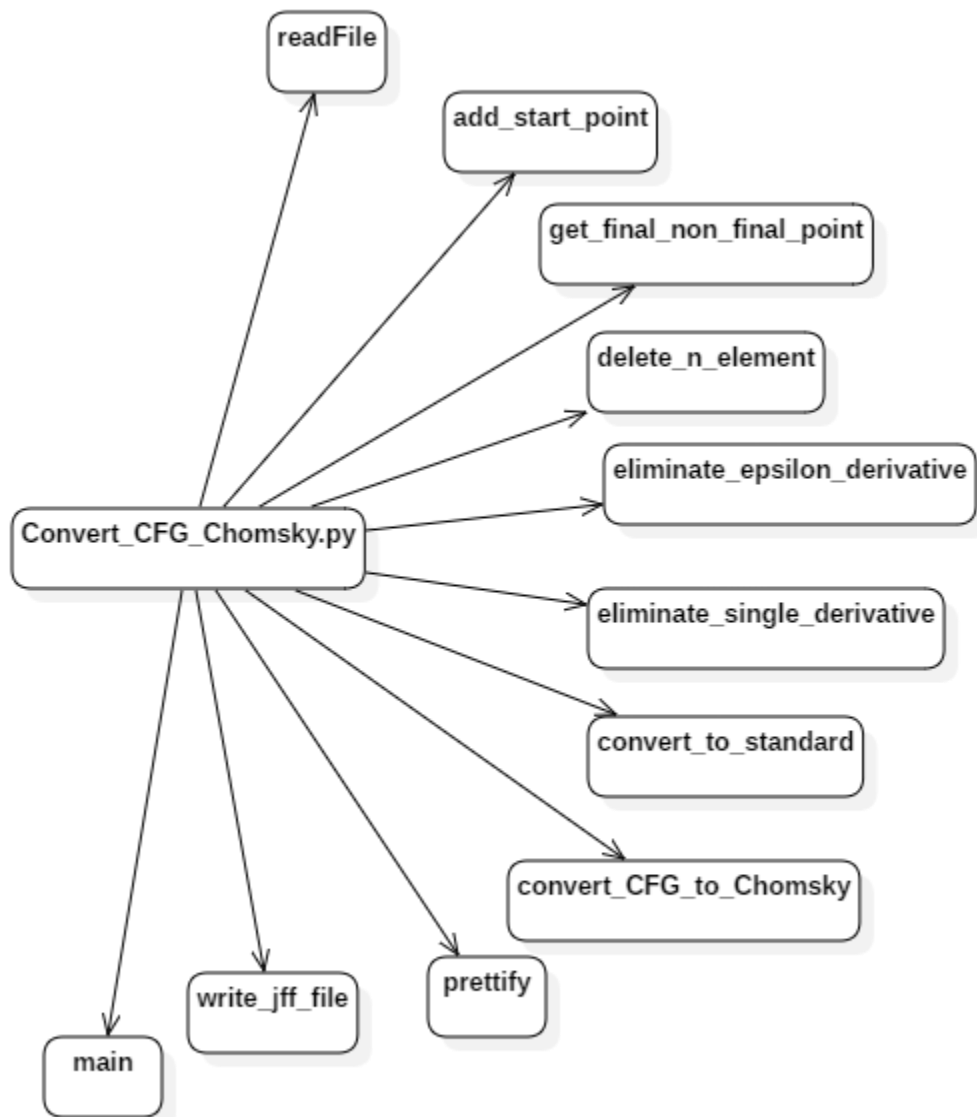
- Với mỗi quy tắc dạng $A \rightarrow B$, ta thực hiện:
 - o Loại bỏ quy tắc $A \rightarrow B$
 - o Với mỗi quy tắc dạng $B \rightarrow u$, ta thêm quy tắc $A \rightarrow u$, trừ khi đây là quy tắc đã bị loại bỏ ở các bước trước đó.

2.4. Chuyển các quy tắc còn lại về dạng đúng

- Thay thế các quy tắc dạng $A \rightarrow u_1u_2 \dots u_k$ với $k \geq 3$ và $u_i \in V \cup \Sigma$ bởi các quy tắc $A \rightarrow u_1A_1$, $A_1 \rightarrow u_2A_2$, $A_{k-2} \rightarrow u_{k-1}u_k$
- Thay thế mọi ký hiệu kết thúc u_i trong các quy tắc vừa thay thế ở trên bởi biến mới U_i và thêm quy tắc $U_i \rightarrow u_i$

3. Cấu trúc chương trình

Chương trình gồm các hàm:



+ readFile(): đọc file .jff đầu vào và lưu văn phạm thu được vào biến 'grammar'

Lưu văn phạm dưới dạng list

Mỗi phần tử của list gồm 2 phần tử:

1. Biến không kết thúc
2. Dẫn xuất của nó

+ add_start_point(): thêm $S_0 \rightarrow S$ vào đầu văn phạm

1. Sao chép grammar sang grammar_chomsky
2. Thêm $S_0 \rightarrow S$ vào đầu văn phạm chomsky
3. Thêm S_0 vào tập biến không kết thúc
- 4.

+ get_final_non_final_point(): tìm và lưu các điểm kết thúc và không kết thúc vào final_point và non_final_point

(Ở đây lưu chữ viết hoa là biến không kết thúc, chữ thường là biến kết thúc)

+ delete_n_element(k, n, point, collect, added): xóa các vị trí A mà $A \rightarrow \epsilon$ trong dẫn xuất

Point: A($A \rightarrow \epsilon$)

n: số lần xuất hiện của A trong dẫn xuất collect

k: số A được xóa

1. $k > n$ return
2. xóa 1 A trong dẫn xuất
3. gọi lại delete_n_element($k+1, \dots$)

+ eliminate_epsilon_derivative(): bước xóa dẫn xuất $A \rightarrow \epsilon$ trong thuật toán

+ eliminate_single_derivative(): xóa các dẫn xuất đơn

+ convert_to_standard(): đưa dẫn xuất về dạng chuẩn

+ convert_CFG_to_Chomsky(): tổng hợp các hàm ở trên đưa CFG về dạng chuẩn Chomsky

+ prettify(), writte_jff_file(): đưa dạng chuẩn Chomsky về dạng file .jff

+ main(): thực hiện chương trình

II. Thuật toán CYK

1. Đề bài

- Viết chương trình mô tả thuật toán CYK

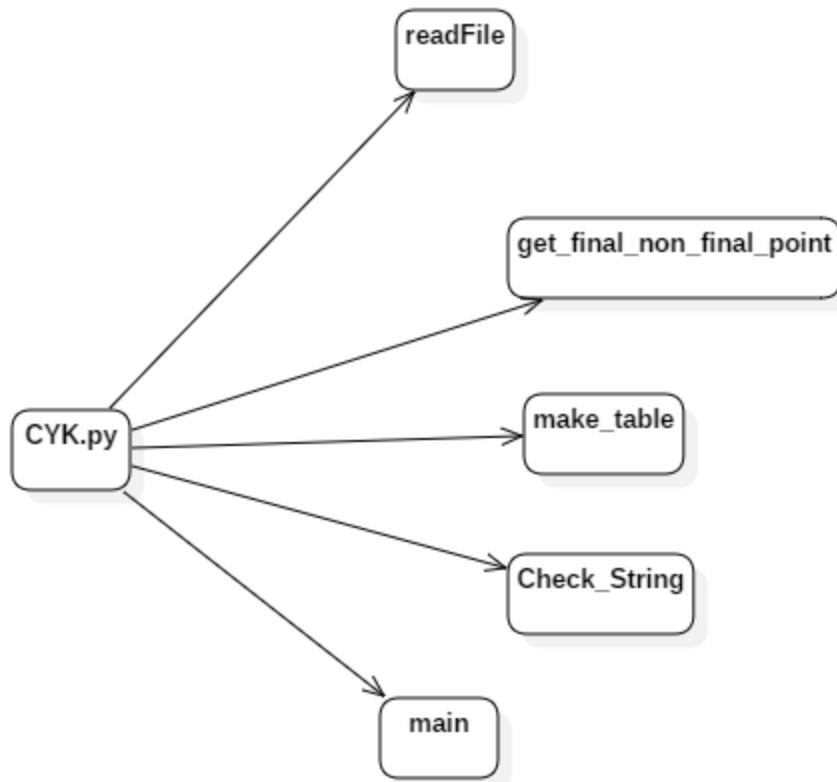
- Input:
 - o File Chomsky.jff mô tả văn phạm G đã ở dạng chuẩn Chomsky
 - o Xâu $w \in \Sigma^*$
- Output: Thông báo “Yes” nếu xâu w được sinh bởi văn phạm G, “No” nếu ngược lại.

2. Thuật toán

- Xét xâu vào $w = w_1w_2 \dots w_n$
- Ta xây dựng bảng $n \times n$: Phần tử (i, j) (với $i \leq j$) của bảng là tập các biến sinh ra xâu $w_iw_{i+1} \dots w_j$.
- Mã giả: Với xâu vào $w = w_1w_2 \dots w_n$
 1. Nếu $w = \varepsilon$ và có luật $S \rightarrow \varepsilon$, chấp nhận. [[xử lý ε]]
 2. For $i = 1$ đến n [[xử lý xâu con độ dài 1]]
 3. For mỗi biến A :
 4. Kiểm tra xem có luật $A \rightarrow b$ với $w_i = b$
 5. Nếu có, đặt A vào $T(i, i)$
 6. For $l = 2$ đến n : [[l là độ dài xâu con]]
 7. For $i = 1$ đến $n - l + 1$:
 8. $j \leftarrow i + l - 1$
 9. for $k = i$ đến $j - 1$:
 10. for mỗi sản xuất $A \rightarrow BC$
 11. Nếu $B \in T(i, k)$ và $C \in T(k + 1, j)$, đặt A vào $T(i, j)$.
 12. Nếu $S \in T(1, n)$, chấp nhận xâu w . Ngược lại, bác bỏ.

3. Cấu trúc chương trình

Các hàm:



- + readFile(): đọc và lưu trữ văn phạm
- + get_final_non_final_point()
- + make_table(): tạo bảng table[n][n]
- + Check_String(): kiểm tra chuỗi
- + main()