

به نام خدا

گزارش کار پروژه ی پیاده سازی الگوریتم CYK

علی بابارهایی 992164022

محمد مهدی عباسی 992164034

نام درس : نظریه زبانها و ماشین ها

بهمن سال 1401

نام استاد : وحید حقیقت دوست

### فهرست

2	شرح کلی الگوریتم CYK
4	Pseudocode
4	مثال :
6	انجام مثال ها با زبان برنامه نویسی و کار با نرم افزار
8	فیلم کار با برنامه لینک کامل کدها :
8	منابع:

## شرح کلی الگوریتم CYK

(الگوریتم عضویت برای گرامرهای مستقل از متن)

C:j. Cocke



Y:D.H. Younger,



K:T. Kasami



روش کار این الگوریتم به صورتی هست که مسئله را به **بخش های کوچکتری** تقسیم میکند.

در علوم رایانه یک الگوریتم **برنامهریزی پویا** برای بدست آوردن تجزیه گرامری جملات با استفاده از گرامر مستقل از متن است. نکته ی قابل توجه این هست که این الگوریتم تنها در صورتی کار میکند که گرامر در **فرم نرمال چامسکی** باشد.

این الگوریتم از برنامه نویسی پویا به این شکل استفاده می کند که ابتدا برای هر حرف یک جمله ورودی، تمام متغیرهایی که منجر به تولید آن حرف می شوند را در مجموعه ای مجزا قرار می دهد. بعد همین کار را برای زیرجمله های دو حرفی تکرار می کند و بعد زیرجمله های سه حرفی تا به کل جمله برسد. در نهایت اگر متغیر ابتدایی را در مجموعه متغیرهای جمله پایانی یافت نتیجه می گیرد که جمله توسط گرامر تولید شده است.

$$A \in V_{ij} \text{ iff } \exists k : i \leq k < j \quad A \rightarrow BC \ \& \ B \Rightarrow^* w_{ik} \ \& \ C \Rightarrow^* w_{k+1j}$$

$$V_{ij} = \{A \in V : A \Rightarrow^* w_{ij}\}$$

$$V_{ij} = \bigcup_{k \in \{i, i+1, \dots, j-1\}} \{A : A \rightarrow BC, \text{ with } B \in V_{ik}, \ C \in V_{k+1j}\}$$

## Pseudocode

```
BUILD_CKY_CHART( $[w_1, \dots, w_n]$ , grammar) returns table  
for  $j \leftarrow 1$  to  $n$  do  
     $\text{table}(j-1, j) \leftarrow \{A \mid A \xrightarrow{+} w_j \in \text{grammar}\}$   
for  $j \leftarrow 2$  to  $n$   
    for  $i \leftarrow j-2$  downto  $0$   
        for  $k \leftarrow i+1$  to  $j-1$  do  
             $\text{table}(i, j) \leftarrow \text{table}(i, j) \cup$   
                 $\{A \mid A \rightarrow BC \in \text{grammar},$   
                     $B \in \text{table}(i, k),$   
                     $C \in \text{table}(k, j)\}$ 
```

پیچیدگی الگوریتم عضویت دارای پیچیدگی  $O(n^3)$  میباشد.

## مثال :

می خواهیم وجود یا عدم وجود رشته ی  $w = aabbb$  در زبان تولید شده بوسیله گرامر زیر را تحقیق کنیم.

$$S \rightarrow AB$$

$$A \rightarrow BB \mid a$$

$$B \rightarrow AB \mid b$$

ابتدا یادآوری میکنیم که  $w_{11} = a$  بنابراین  $V_{11}$  مجموعه تمام متغیرهایی است که فوراً  $a$  را اشتقاق میکنند. یعنی  $V_{11} = \{A\}$

از آنجاییکه  $w_{22} = a$ ، داریم که  $V_{22} = \{A\}$  و به همین ترتیب،

$$V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$$

اینک از

$$\{V_{ij} = \bigcup_{k \in \{i, i+1, \dots, j-1\}} \{A : A \rightarrow BC, \text{ with } B \in V_{ik}, C \in V_{kj}\}\}$$

میتوان نتیجه گرفت.

$$V_{12} = \{A : A \rightarrow BC, B \in V_{11}, C \in V_{22}\}$$

این مجموعه شامل تمام متغیرهای واقع در طرف چپ قوانینی است که در طرف راست آنها  $AA$  قرار دارد.

چون هیچ قانونی به این فرم وجود ندارد لذا  $V_{12} = \emptyset$ .

$$V_{23} = \{A : A \rightarrow BC, B \in V_{22}, C \in V_{33}\}$$

بنابر این  $AB$  باید در طرف راست قرار داشته باشد و از این رو  $V_{23} = \{S, B\}$

پس خواهیم داشت :

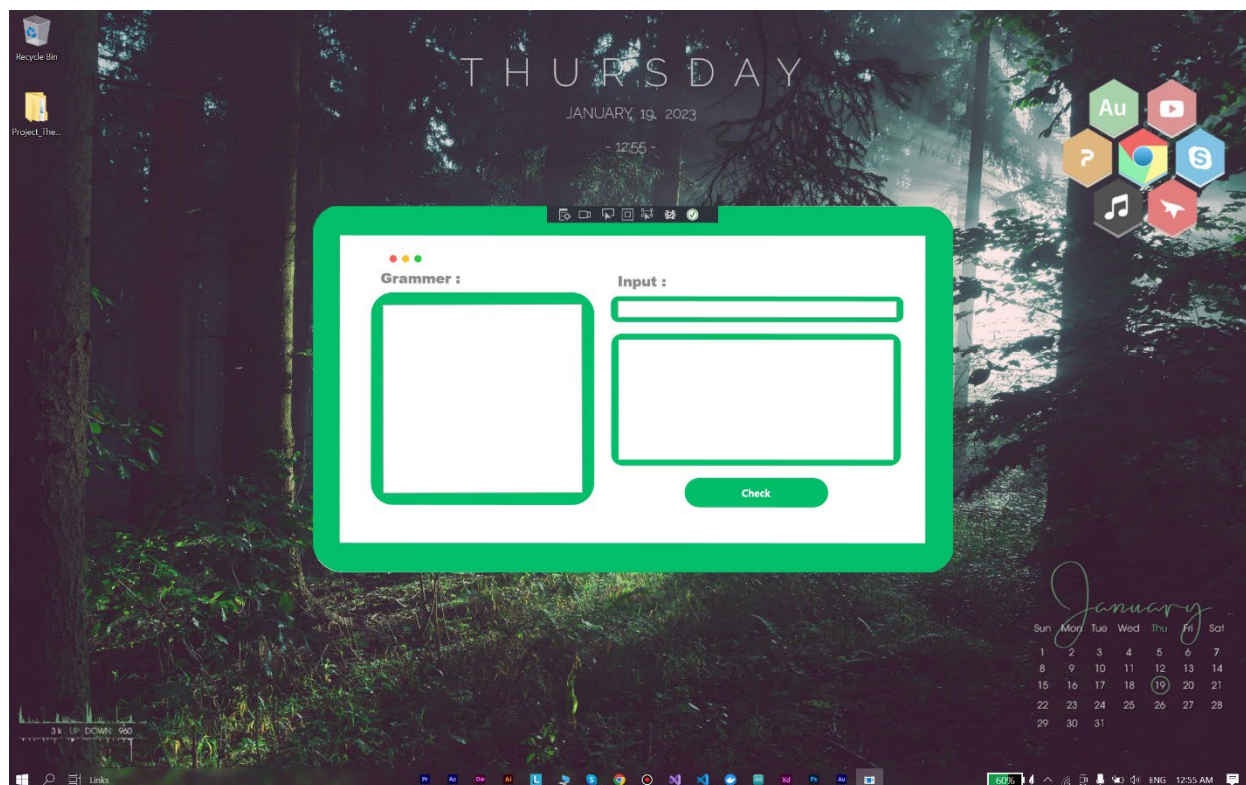
$$V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$$

$$V_{12} = \emptyset, V_{23} = \{S, B\}, V_{34} = \{A\}, V_{45} = \{A\}, V_{13} = \{S, B\}$$

$$V_{24} = \{A\}, V_{35} = \{S, B\}, V_{14} = \{A\}, V_{25} = \{S, B\}, V_{15} = \{S, B\}$$

$w$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$V_{ii}$	$V_{11}$	$V_{22}$	$V_{33}$	$V_{44}$	$V_{55}$
$V_{ii+1}$	$V_{12}$	$V_{23}$	$V_{34}$	$V_{45}$	
$V_{ii+2}$	$V_{13}$	$V_{24}$	$V_{35}$		
$V_{ii+3}$	$V_{14}$	$V_{25}$			
$V_{ii+4}$	$V_{15}$				

## انجام مثال ها با زبان برنامه نویسی و کار با نرم افزار (c#)



در اینجا نمای کلی نرم افزار را مشاهده میکنیم که با استفاده از زبان **c#** و همچنین با **WPF** نمای کلی آن را طراحی کرده ایم.

در سمت چپ گرامر را دریافت میکنیم و باید توجه داشت که در این قسمت باید برای جداسازی هر قانون از علامت **" / "** این استفاده کنیم.

علامت **" | "** تعریف نشده است برای مثال

$S \rightarrow A|B$  غلط

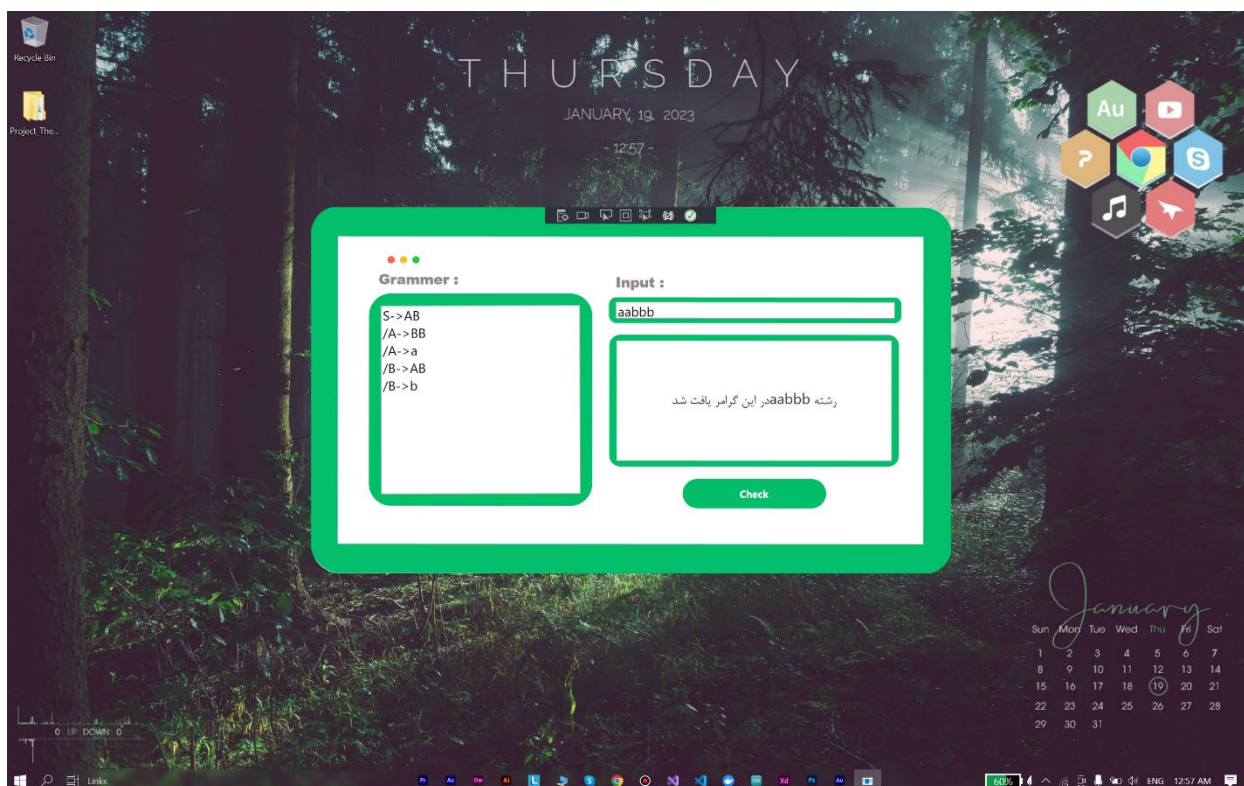
$S \rightarrow A$  درست

$/S \rightarrow b$

در قسمت بالای سمت راست ما ورودی (رشته) را از کاربر خواهیم گرفت که این ورودی اگر با زبان ساخته شده توسط گرامر سمت چپ

پذیرفته شد، خروجی این عملیات در پنجره ی سمت راست پایین نمایش داده میشود.

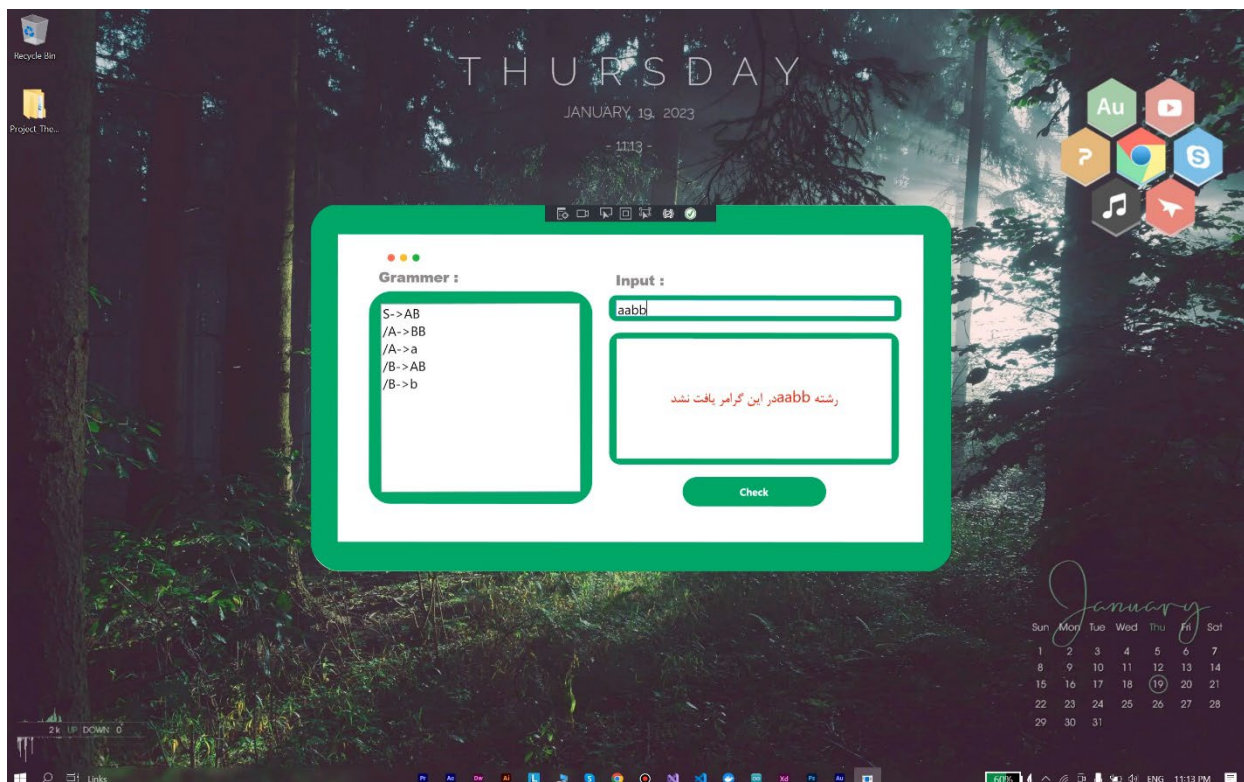
برای مثال :



در این زبان ما میتوانیم رشته ی  $w=aabb$  را پیدا کنیم. که در مثال جزوه هم نیز این رشته پذیرفته شد.

اما اگر رشته ی  $aabb$  را قرار دهیم نمیتوانیم در این گرامر پیدا کنیم





همچنین دکمه قرمز سمت چپ خروج از اپلیکیشن است

فیلم کار با برنامه لینک کامل کدها :

منابع:

[https://en.wikipedia.org/wiki/CYK\\_algorithm](https://en.wikipedia.org/wiki/CYK_algorithm)

Grune, Dick (2008). Parsing techniques

P.Linz-Book- Introduction-Formal-Languages-Automata

Slides of my master in Shahed University



پایان  
موفق باشید