بسم الله الرحمن الرحیم

نظریه زبان‌ها و ماشین‌ها

جلسه ۱۷

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

الگوریتم CYK

Assume that we have a grammar $G = (V, T, S, P)$ in Chomsky normal form and a string

$$w = a_1 a_2 \cdots a_n.$$

We define substrings

$$w_{ij} = a_i \cdots a_j,$$

and subsets of V

$$V_{ij} = \left\{ A \in V : A \overset{*}{\Rightarrow} w_{ij} \right\}.$$

Clearly, $w \in L(G)$ if and only if $S \in V_{1n}$.

الگوریتم CYK

IUT-ECE

$$V_{ij} = \left\{ A \in V : A \stackrel{*}{\Rightarrow} w_{ij} \right\}.$$

1. To compute V_{ij}, observe that $A \in V_{ii}$ if and only if G contains a production $A \rightarrow a_i$.

2. for $j > i$, A derives w_{ij} if and only if there is a production $A \rightarrow BC$, with $B \stackrel{*}{\Rightarrow} w_{ik}$ and $C \stackrel{*}{\Rightarrow} w_{k+1j}$ for some k with $i \leq k, k < j$. In other words,

$$V_{ij} = \bigcup_{k \in \{i, i+1, \ldots, j-1\}} \{A : A \rightarrow BC, \text{ with } B \in V_{ik}, C \in V_{k+1,j}\}. \qquad (6.8)$$

الگوریتم CYK

$$V_{ij} = \left\{ A \in V : A \stackrel{*}{\Rightarrow} w_{ij} \right\}.$$

An inspection of the indices in (6.8) shows that it can be used to compute all the V_{ij} if we proceed in the sequence

1. Compute $V_{11}, V_{22}, ..., V_{nn}$,

2. Compute $V_{12}, V_{23}, ..., V_{n-1,n}$,

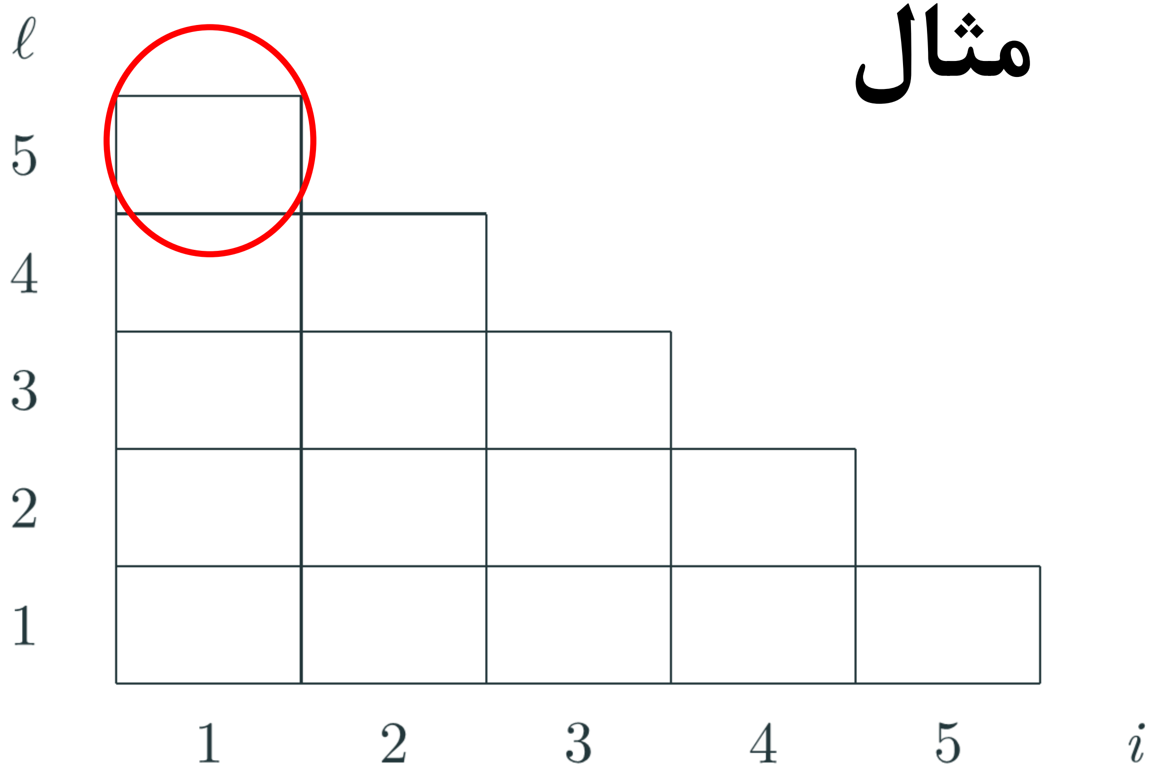
3. Compute $V_{13}, V_{24}, ..., V_{n-2,n}$,

and so on.

Dynamic programming

IUT-ECE

مثال



$$w_{i,l} = w_i w_{i+1} \dots w_{i+l-1}$$

- برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

مثال

IUT-ECE

ℓ					
5					
4					
3					
2					
1	V11	V22	V33	V44	V55
	1	2	3	4	5	i

$$w_{i,l} = w_i w_{i+1} \dots w_{i+l-1}$$

- برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

$$V_{ii} = \{A:\ A \rightarrow w_i\}$$

مثال

ℓ					
5	V15				
4	V14	V25			
3	V13	V24	V35		
2	V12	V23	V34	V45	
1	V11	V22	V33	V44	V55
	1	2	3	4	5	i

$$w_{i,l} = w_i w_{i+1} \ldots w_{i+l-1}$$

- برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

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

مثال

$S \to AB \mid BC$

$A \to BA \mid \text{a}$

$B \to CC \mid \text{b}$

$C \to AB \mid \text{a}$

$x = \text{baaba}$

$$w_{i,l} = w_i w_{i+1} \dots w_{i+l-1}$$

ℓ						
5						
4						
3						
2						
1						
	1	2	3	4	5	i
	b	a	a	b	a	

- برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

$$V_{ii} = \{A\colon\ A \to w_i\}$$

مثال

5	V15				
4	V14	V25			
3	V13	V24	V35		
2	V12	V23	V34	V45	
1	V11	V22	V33	V44	V55
	1	2	3	4	5

ℓ						
5						
4						
3						
2	$S \mid A$	B	$S \mid C$	$S \mid A$		
1	B	$A \mid C$	$A \mid C$	B	$A \mid C$	
	1	2	3	4	5	i
	b	a	a	b	a	

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

IUT-ECE

مثال

$S \to AB \mid BC$

$A \to BA \mid \mathrm{a}$

$B \to CC \mid \mathrm{b}$

$C \to AB \mid \mathrm{a}$

$x = \mathrm{baaba}$

$w_{i,l} = w_i w_{i+1} \dots w_{i+l-1}$

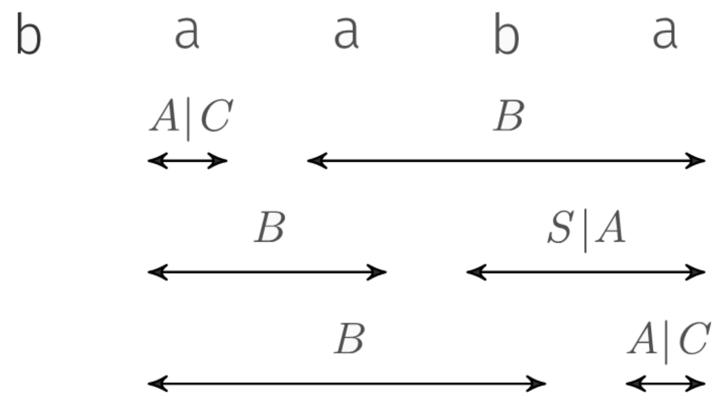
ℓ					
5					
4					
3	-	B	B		
2	$S\mid A$	B	$S\mid C$	$S\mid A$	
1	B	$A\mid C$	$A\mid C$	B	$A\mid C$
i	1	2	3	4	5
	b	a	a	b	a

- برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

مثال

- برای T[2,4]

5					
4					
3	-	B	B		
2	$S\|A$	B	$S\|C$	$S\|A$	
1	B	$A\|C$	$A\|C$	B	$A\|C$
	1	2	3	4	5



$$S \rightarrow AB \mid BC$$
$$A \rightarrow BA \mid \text{a}$$
$$B \rightarrow CC \mid \text{b}$$
$$C \rightarrow AB \mid \text{a}$$

$$T[2,4] = S|A|C$$

مثال

IUT-ECE

$$S \to AB \mid BC$$
$$A \to BA \mid \text{a}$$
$$B \to CC \mid \text{b}$$
$$C \to AB \mid \text{a}$$

$$x = \text{baaba}$$

$$w_{i,l} = w_i w_{i+1} \dots w_{i+l-1}$$

ℓ					
5	$S\|A\|C$				
4	-	$S\|A\|C$			
3	-	B	B		
2	$S\|A$	B	$S\|C$	$S\|A$	
1	B	$A\|C$	$A\|C$	B	$A\|C$
i	1	2	3	4	5
	b	a	a	b	a

○ برای هر زیررشته w_il مقدار V[i,l] (همه متغیرهای که آن را اشتقاق میکنند) را در T[i,l] ذخیره کن.

مثال

$S \to AB \mid BC$

$A \to BA \mid \mathsf{a}$

$B \to CC \mid \mathsf{b}$

$C \to AB \mid \mathsf{a}$

$x = \mathsf{baaba}$

ℓ					
5	$S\|A\|C$				
4	-	$S\|A\|C$			
3	-	B	B		
2	$S\|A$	B	$S\|C$	$S\|A$	
1	B	$A\|C$	$A\|C$	B	$A\|C$
i	1	2	3	4	5
	b	a	a	b	a

مثال

$S \rightarrow AB|AC|AA,$
$A \rightarrow CB|a,$
$B \rightarrow AC|b,$
$C \rightarrow CC|b.$

w=bbabb

5	S|A|B				
4	S|A	S|A|B			
3	S	A	S|B		
2	A|C	-	S|B	A|C	
1	B|C	B|C	A	B|C	B|C
	1	2	3	4	5
	b	b	a	b	b

مثال

$$S \rightarrow AB|AC|AA,$$
$$A \rightarrow CB|a,$$
$$B \rightarrow AC|b,$$
$$C \rightarrow CC|b.$$

w=bbabb

5	S|A|B				
4	S|A	S|A|B			
3	S	A	S|B		
2	A|C	-	S|B	A|C	
1	B|C	B|C	A	B|C	B|C
	1	2	3	4	5
	b	b	a	b	b

الگوریتم CYK

The CYK algorithm, as described here, determines membership for any language generated by a grammar in Chomsky normal form. With some additions to keep track of how the elements of V_{ij} are derived, it can be converted into a parsing method. To see that the CYK membership algorithm requires $O\left(n^3\right)$ steps, notice that exactly $n\left(n+1\right)/2$ sets of V_{ij} have to be computed. Each involves the evaluation of at most n terms in (6.8), so the claimed result follows.

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

فرم نرمال گریباخ (Greibach)

Another useful grammatical form is the **Greibach normal form**. Here we put restrictions not on the length of the right sides of a production, but on the positions in which terminals and variables can appear. Arguments justifying Greibach normal form are a little complicated and not very transparent. Similarly, constructing a grammar in Greibach normal form equivalent to a given context-free grammar is tedious. We therefore deal with this matter very briefly. Nevertheless, Greibach normal form has many theoretical and practical consequences.

فرم نرمال گریباخ (Greibach)

DEFINITION 6.5

A context-free grammar is said to be in Greibach normal form if all productions have the form

$$A \to ax,$$

where $a \in T$ and $x \in V^*$.

فرم نرمال گریباخ (Greibach)

IUT-ECE

EXAMPLE 6.9

The grammar

$$
\begin{aligned}
S &\rightarrow AB, \\
A &\rightarrow aA \,|bB|\, b, \\
B &\rightarrow b
\end{aligned}
$$

is not in Greibach normal form. However, using the substitution given by Theorem 6.1, we immediately get the equivalent grammar

$$
\begin{aligned}
S &\rightarrow aAB \,|bBB|\, bB, \\
A &\rightarrow aA \,|bB|\, b, \\
B &\rightarrow b,
\end{aligned}
$$

which is in Greibach normal form.

فرم نرمال گریباخ (Greibach)

IUT-ECE

THEOREM 6.7

For every context-free grammar G with $\lambda \notin L(G)$, there exists an equivalent grammar $\widehat{G}$ in Greibach normal form.

اتوماتای پشته‌ای

- تاکنون با اتوماتای متناهی آشنا شده‌ایم (متناظر با زبان منظم).
- همانطور که بیان کردیم بسیاری از زبان‌ها نیاز به حافظه نامحدود دارند. مانند زبان زیر که یک CFL است:

$$\{a^n b^n \mid n \geq 0\}$$

- اکنون قصد داریم اتوموتنی معرفی کنیم که CFL را تشخیص دهد.
- بنابراین به همان NFA یک stack با حافظه نامحدود اضافه میکنیم.

اتوماتای پشته‌ای

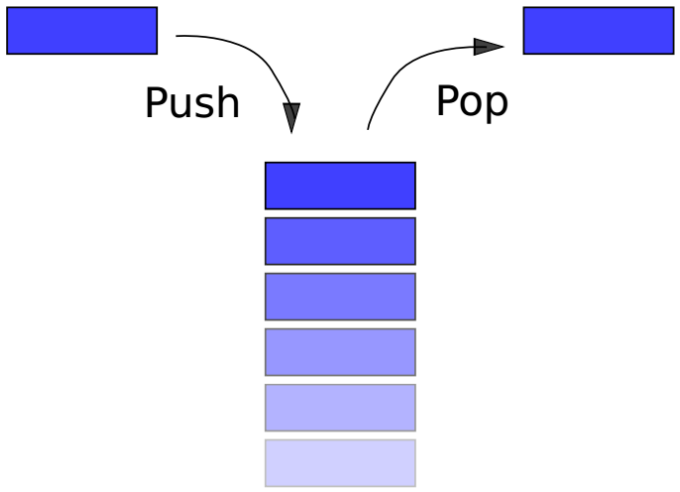
$$\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$$

اتوماتای پشته‌ای

Pushdown automata are equivalent in power to context-free grammars. This equivalence is useful because it gives us two options for proving that a language is context free. We can give either a context-free grammar generating it or a pushdown automaton recognizing it. Certain languages are more easily described in terms of generators, whereas others are more easily described by recognizers.

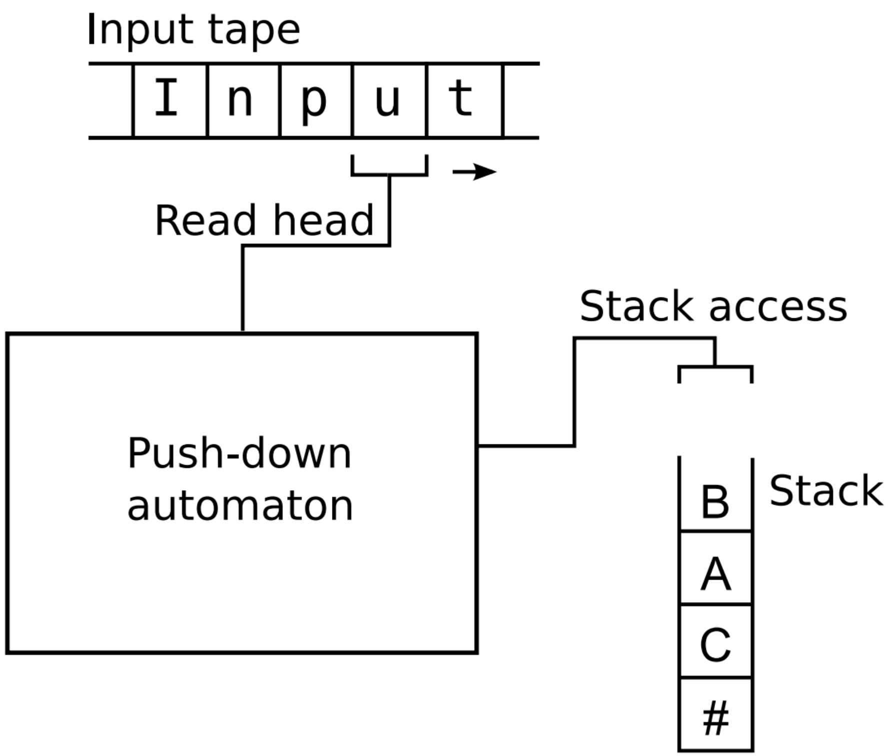
اتوماتای پشته‌ای

- پشته از اصل LIFO پیروی کرده و دو عملگر اصلی push و pop دارد.



اتوماتای پشته‌ای

- در هر گام، معمولا یک سمبل از ورودی و یک سمبل از پشته خوانده می‌شود، سپس ضمن بروزرسانی حالت اتوماتا، معمولا یک سمبل هم در پشته نوشته می‌شود یا بروز می‌شود یا ...



اتوماتای پشته‌ای

IUT-ECE

- قصد داریم مانند FA، برای PDA نیز دیاگرام رسم کنیم.

- برای همین منظور، هر فلش شامل یک سه تایی است: سمبل ورودی، سمبل پاپ شده/ سمبل پوش شده.

- مثال:

0x/1

readSymbol , poppedSymbol / pushedSymbol