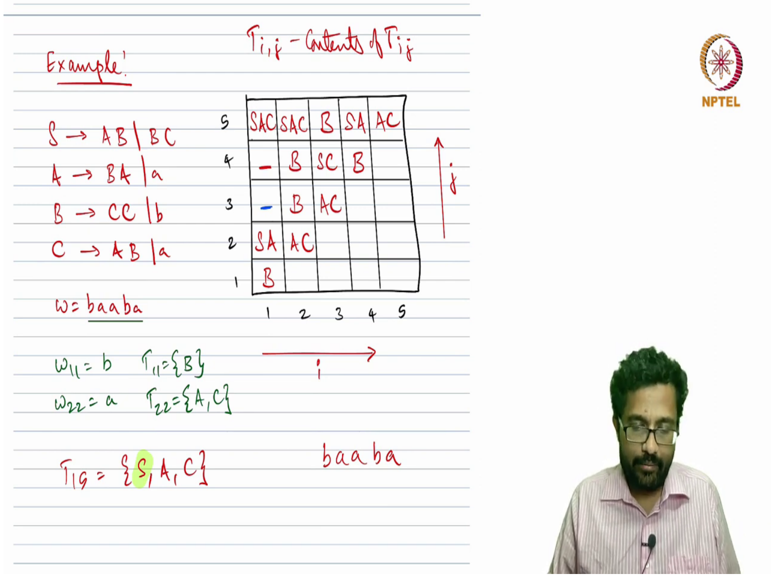
Theory of Computation
Professor Subrahmanyam Kalyanasundaram
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
CYK Algorithm - Part 2 (Example)

(Refer Slide Time: 0:16)



Example:

$T_{i,j}$ - Contents of $T_{i,j}$

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

$w = baaba$

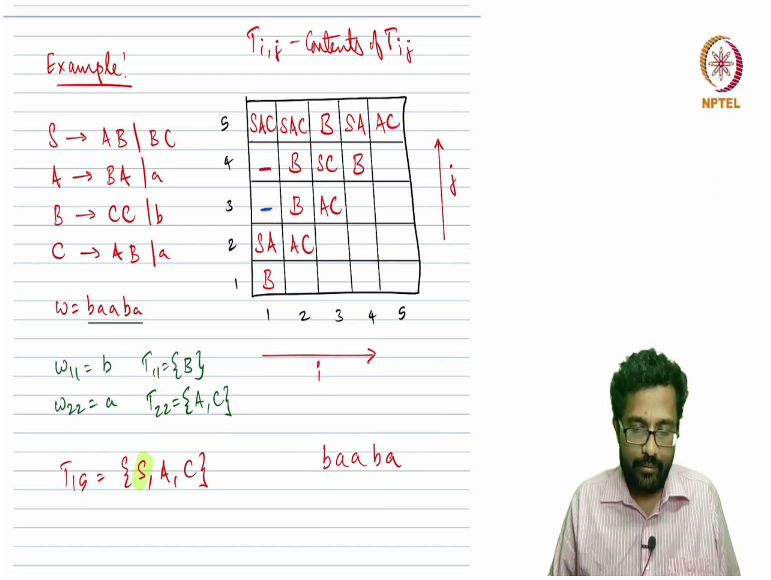
5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

$w_{11} = b \quad T_{11} = \{B\}$
$w_{22} = a \quad T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

baaba

So maybe, let us work out an example and maybe this will make it clear. So, let us see, this is grammar. S gives AB and BC, A gives BA, B gives CC and b, C gives AB and a. And the string that you need to check is "baaba". And we can check that this grammar is in the Chomsky normal form. And this grid that I have built is for writing the T_{ij} ,so I will populate the contents of T_{ij}. Contents of the set T_{ij} will be populated in the respective boxes.



Example:

$T_{i,j}$ - Contents of T_{ij}

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

$w =$ baaba

5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

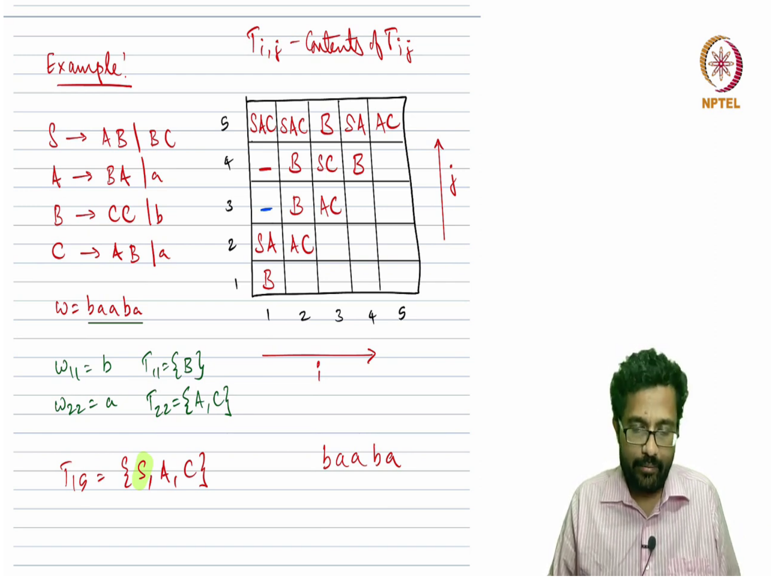
$w_{11} = b \quad T_{11} = \{B\}$

$w_{22} = a \quad T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

baaba

So, for instance the bottom left box, bottom left cell corresponds to T_{11} and earlier when I explained we noticed that w_{11} is a symbol b. So, T_{11} contains just the variable B. And w_{22} is a symbol a and a is derived from both the variable A and the variable C. So, we will write both A and C in the cell 22. So, notice that the columns correspond to i and rows correspond to j and so, the first order is to fill up the diagonals like this (11,22,33,44,55) and then we will slowly move up, we will basically fill the entries above the diagonal (12,23,34,45) we will not fill the entries below the diagonal and they will not be necessary also. The third symbol is also 'a' and which are the variables that generate 'a' again? A and C. Fourth symbol is 'b' , we already know that only the variable B generates this and the fifth symbol is 'a'.



Example:

$T_{i,j}$ - Contents of T_{ij}

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

$w = baaba$

5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

$w_{11} = b \quad T_{11} = \{B\}$

$w_{22} = a \quad T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

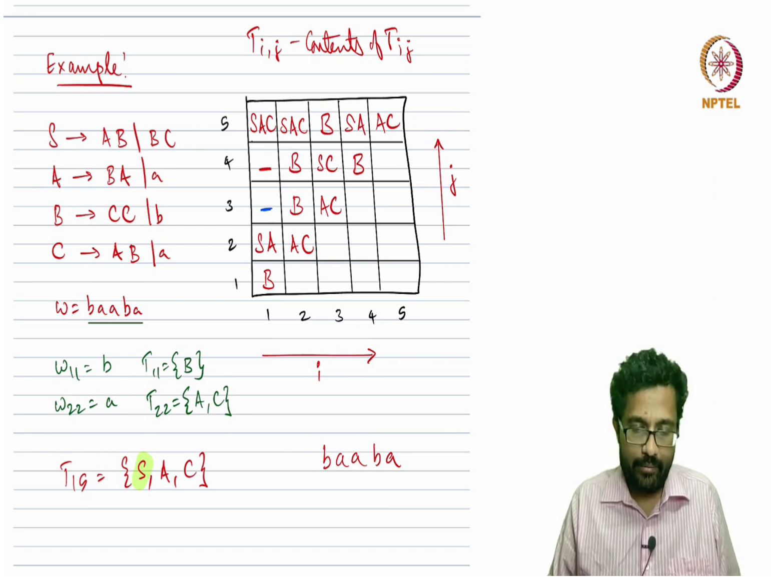
baaba

So, now we have filled the sets corresponding to the single length substrings. Now let us move to the substrings of length 2. So, first is 12, which is to be filled. So, what we need to check is that the only possible split is 11 and 22. So, there are 2 possibilities, so if you see the 11 cells contain B and 22 cell contains AC. So, is there a rule where BA is generated by some variable? or is there a rule where BC is generated by some variable? if so it will be included in the cell 1 2. So, if you see that B A is generated by A, BC is generated by S. So, S and A will be filled in 1 2.

Now let us move to the cell 2 3. So, now we have to check the combination of 2 2 and 3 3. So, now there are 4 combinations of the variables in 2 2 and 3 3. So, for AA is there any variable that that derives AA? there is none, For AC again there is none, CA again there is none, CC is derived by B. So, we fill a B here.

The next is 3 4, so we need to check the combinations of 3 3 and 4 4. So, AB is derived by S as well as C, then CB is not derived by anybody, so it is just S and C.

Then 4 5,For this cell we look up 4 4 i,e B and 5 5 i.e AC , this has w_{45} = “ba”, so this is similar to the cell 1 2 because it is exactly the same variables. So, we can just copy that S and A. So, for substrings of length 2 there was only one way to split, the first part basically if you look at this cell 1 2 the way to split is some variable is generating two other variables, the first variable should give rise to 1 1 and the second variable should give rise to 2 2. There is no other way to break a string of length 2, just the first symbol and the second symbol, this is because the empty strings are not there.



Example:

$T_{i,j}$ - Contents of T_{ij}

$S \to AB \mid BC$

$A \to BA \mid a$

$B \to CC \mid b$

$C \to AB \mid a$

w = baaba

5	SAC	SAC	B	SA	AC
4	-	B	SC	B	
3	-	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

$w_{11} = b \quad T_{11} = \{B\}$

$w_{22} = a \quad T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

baaba

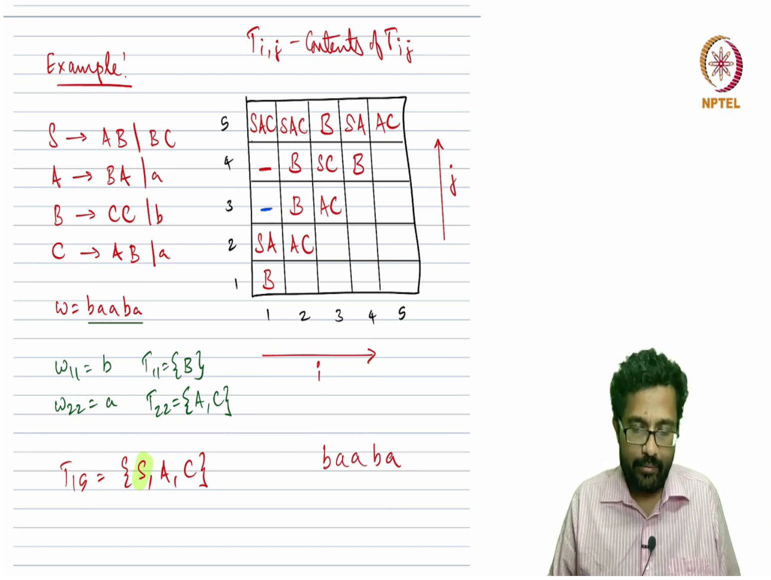
So, now let us move to strings of length 3. So, now let us move to this cell 1 3. So, now there are two ways to split it. So, 1 3 is,basically “baa”, there are 2 ways to split it, it could be the case that, it could be split like this, the first part derives “b” and we know that the part that derives “b” is only the capital B, in this cell 1 1, and the second part derives “aa”, we know that the only way to get this is given by the cell 2 3, this one again B. So now, the only way to get that is by 1 1 and 2 3. So, is there a variable that gives us BB? The answer is, no. So, you cannot get this split.

So, we do not fill up anything. So, another possible split is this 1 2 and 3 3. So, there are four combinations the combinations being, SA, SC, AA, and AC. So, SA is not there, in the right side of any rule, SC is also not there, in fact it will not be there because S is the start variable. So, Chomsky normal form does not allow it.

AA and AC are also not there. So, what this says is that, there is no variable that allows this split also and even the earlier split there was no variable, which means that we cannot enter anything here. So, I just put a blank here which means there is no variable that generates the string “baa”.

So, now let us look at the entry 2 4. which is “aab”, again there are two ways to consider, two splits to consider “a” and “ab”, 2 2 and 3 4. So, there are four combinations AS, AC which are not there CS is not there, CC is there derived by B. The other possibility is 2 3 and 4 4. So, 2 3 this one and 4 4 which is basically BB and BB is not generated by anything.

So, there is only one way to derive the string “aab”, which is from the variable B. So, variable B gives CC and the first C gives “a” and the second C gives A B and then you get “ab” from that, so that is how it is. Now finally the last string of length 3 is “aba”. So, now let me just consider the two splits possible. So, one is 3 3 and 4 5. So, AS is not there, AA is also not there, CS is not there, CA is also not there. So, this split does not yield anything but then there is one more split to consider before deciding whether the cell is empty the other split is 3 4 and 5 5. So, SA is not there, SC is not there (S is a starting variable), CA is not there, however CC is there, CC is derived by B. So, we enter B here. So, now we have entered the diagonal which corresponds to substrings of length 1, then the substrings of length 2, then the substrings of length 3.



Example:

$T_{i,j}$ - Contents of T_{ij}

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

w = baaba

j					
5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

i

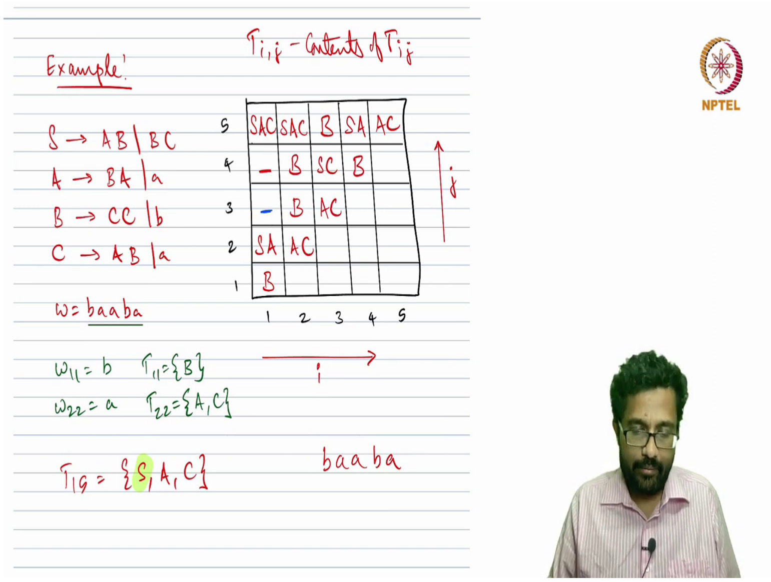
$w_{11} = b$ $T_{11} = \{B\}$

$w_{22} = a$ $T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

baaba

Now, let us move to the substrings of length 4. So, the first substring of length 4 is 1 4. Which corresponds to “baab” , there are three possible splits. So, we could have “b” and then “aab” which corresponds to 1 1 and 2 4, then “ba” and “ab” which is 1 2 and 3 4 and then 1 3 “baa” and “b”. So, three splits to consider. Let us see, 1 1 and 2 4. So, BB is not generated by anything. So, there is no variable that gives BB. For 1 2 and 3 4, S S is not there, S C is not there, A S is not there, A C is also not there, in fact three of these combinations involve S, so we do not need to check them because we cannot have S in the right hand side of the rule. So, there is nothing to fill. Now, finally 1 3 and 4 4 but then for 1 3 itself, there is no variable that generates the substring 1 3. So, that itself is an empty case which means there is no variable that generates the substring 1 4. So, there is no variable that generates “baab”. Now let us consider the other substring of length 4 which is 2 5 which is “aaba”. So, now there are three splits one is 2 2 and 3 5, AB is generated by S and C, CB is generated by nothing. So, there is S and C already two core possibilities, then 2 3 and 4 5, so BS is not there, BA is generated by A, then the possibility is 2 4 and 5 5, BA which is generated by A but A is already there, so we do not need to add that, BC which is generated by S and S is already there. So, we do not need to update this. So, S,A,C all of them generate the substring 2 5, which is “aaba”. So, now we have filled up the table for all the substrings up to length 1, 2, 3 and 4.



$T_{i,j}$ - Contents of T_{ij}

Example:

$S \to AB \mid BC$

$A \to BA \mid a$

$B \to CC \mid b$

$C \to AB \mid a$

$w = baaba$

	1	2	3	4	5
5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				

$w_{11} = b$ $T_{11} = \{B\}$

$w_{22} = a$ $T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$

baaba

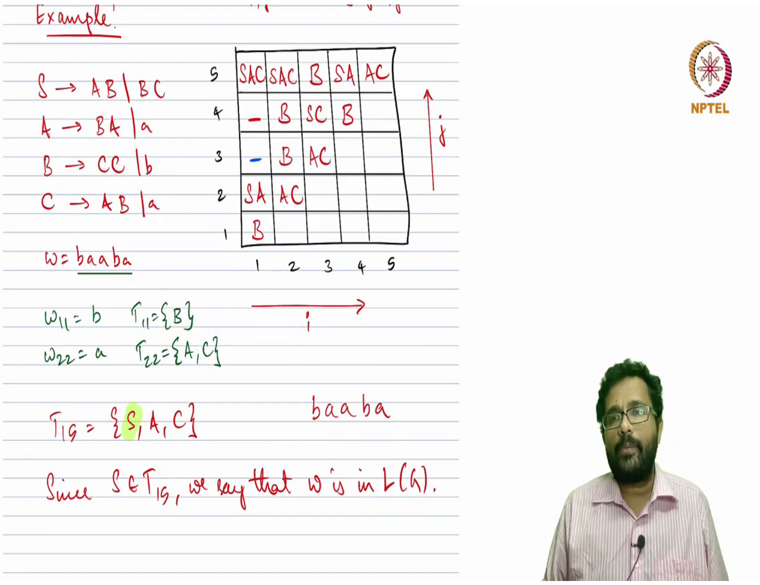
Now finally we have the entire string which is “baaba”. So, because it is of length 5, there are four splits to consider. Now let us see, what are the splits. So for 1 1 and 2 5, B S is not there, B A is generated by A, B C is generated by S. Notice that now we have S in the cell T_{15}, which is the entire string. So, we already know that S derives the entire string which means then the string the string “baaba” is generated by the grammar, which is what our goal was!

So, we can stop it here, but since we started we will just like to complete the algorithm. So, let us just complete the algorithm even though at this point we know that the start variable generates the required state. Now then the other possibility is 1 2 and 3 5, here S B is not there, A B is generated by S as well as C. So, we need to include C here, S is already there.

So, that is another way to derive the same string, S again appears. The next possibility is 1 3 and 4 5, however 1 3 is not derived by any variable. So, which means there is no variable that will be added here. Finally, we have 1 4 and 5 5, but for 1 4 also, there is no variable that derives 1 4. So, that also does not give us anything. So, which means it is complete. We have filled the table for T_{11},T_{22},T_{33},T_{44},T_{55}, then $T_{12}T_{23}T_{34}T_{45}$. Then $T_{13}T_{24}T_{35}$, then $T_{14}T_{25}$, then finally T_{15}.

So, finally we have that T_{15} is basically the set {S,A,C} which means all these variables can derive the entire string. But what is of interest to us is whether the grammar derives, which means whether the start variable derives and since S is here we know that the start variable or the grammar derives the string.

(Refer Slide Time: 16:28)



Example:

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

$w = baaba$

5	SAC	SAC	B	SA	AC
4	–	B	SC	B	
3	–	B	AC		
2	SA	AC			
1	B				
	1	2	3	4	5

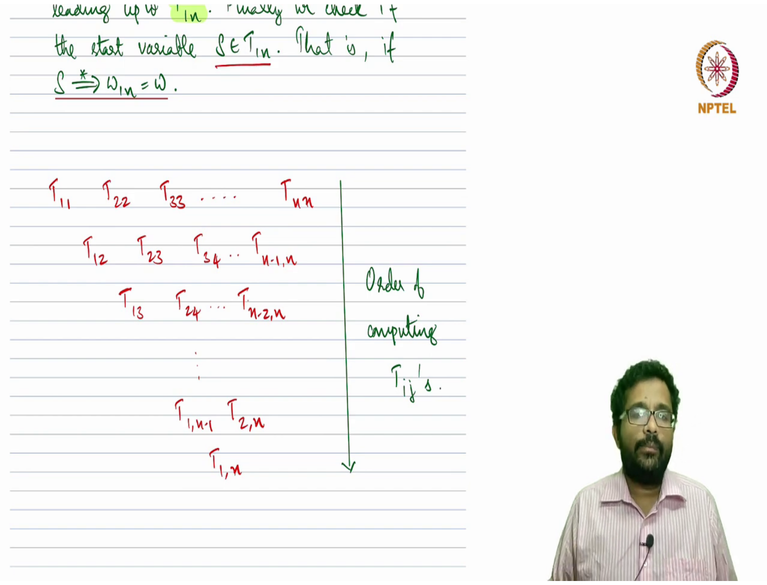
$w_{11} = b \quad T_{11} = \{B\}$
$w_{22} = a \quad T_{22} = \{A, C\}$

$T_{15} = \{S, A, C\}$ baaba

Since $S \in T_{15}$, we say that w is in L(G).

So, we would say that w is in L(G), derived by the grammar, since S in T_{15}, and that completes the illustration of the algorithm.

(Refer Slide Time: 17:03)



leading up to T_{1n}. Finally we check if the start variable $S \in T_{1n}$. That is, if $S \overset{*}{\Rightarrow} w_{1n} = w$.

T_{11} T_{22} T_{33} .... T_{nn}

T_{12} T_{23} T_{34} .. $T_{n-1,n}$

T_{13} T_{24} ... $T_{n-2,n}$

⋮

$T_{1,n-1}$ $T_{2,n}$

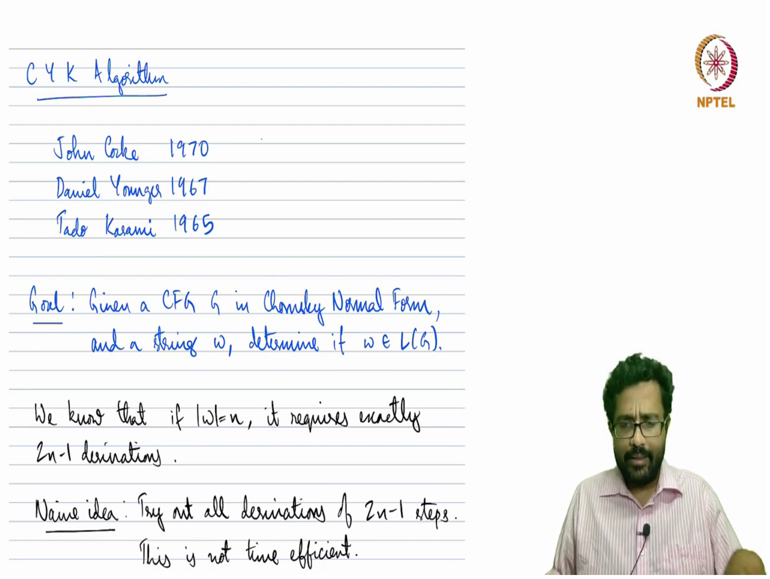
$T_{1,n}$

Order of computing T_{ij}'s.

So, just to summarize, we are given a grammar in Chomsky normal form and a string and we have to determine whether the string is derived by the grammar. So, what we do is we build these sets T_{11} T_{22} etc where, Tij tells us for i j substring which are the variables that derive the i j substring and we build that starting from the smallest length substrings, so the substrings of length 1, T_{11} T_{22} etc correspond to actually substrings which are single variable single symbols.

And then we move up a substrings of length 2, substring of length 3 and so on. Till the entire string, which is also kind of a degenerate substring and we check whether the start variable is part of T_{1n}.

(Refer Slide Time: 17:53)



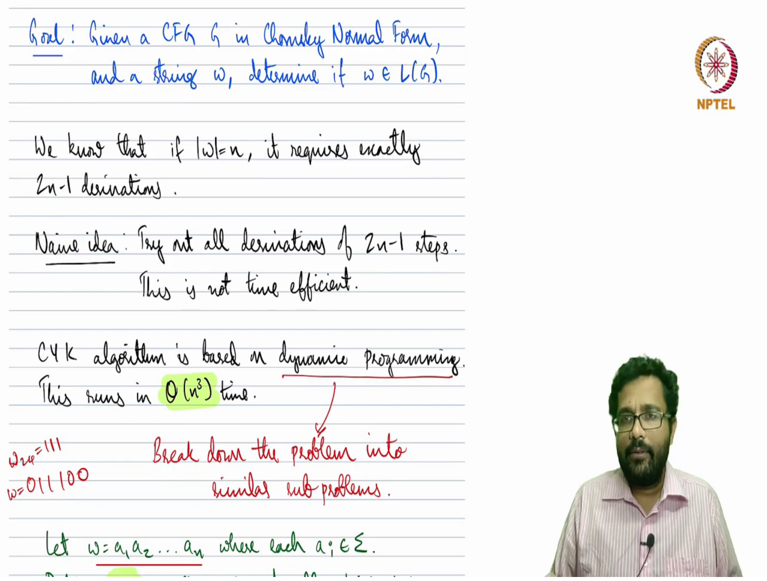
CYK Algorithm

John Cocke 1970
Daniel Younger 1967
Tado Kasami 1965

Goal: Given a CFG G in Chomsky Normal Form, and a string w, determine if $w \in L(G)$.

We know that if $|w| = n$, it requires exactly $2n-1$ derivations.

Naive idea: Try out all derivations of $2n-1$ steps. This is not time efficient.



Goal: Given a CFG G in Chomsky Normal Form, and a string w, determine if $w \in L(G)$.

We know that if $|w| = n$, it requires exactly $2n-1$ derivations.

Naive idea: Try out all derivations of $2n-1$ steps. This is not time efficient.

CYK algorithm is based on dynamic programming. This runs in $O(n^3)$ time.

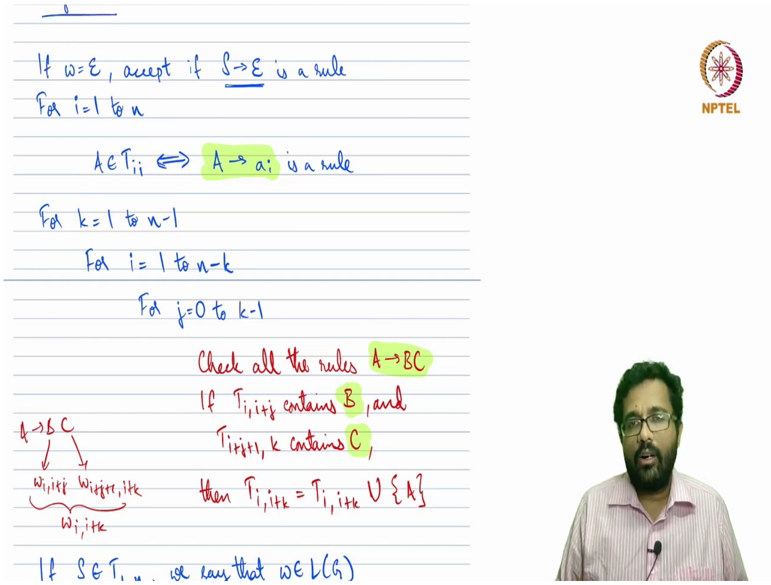
Break down the problem into similar sub problems.

$w_{24} = 111$
$w = 011100$

Let $w = a_1 a_2 \ldots a_n$ where each $a_i \in \Sigma$.

So, that is the C Y K algorithm, if you are familiar with dynamic programming, this kind of algorithmic paradigm is kind of very standard.

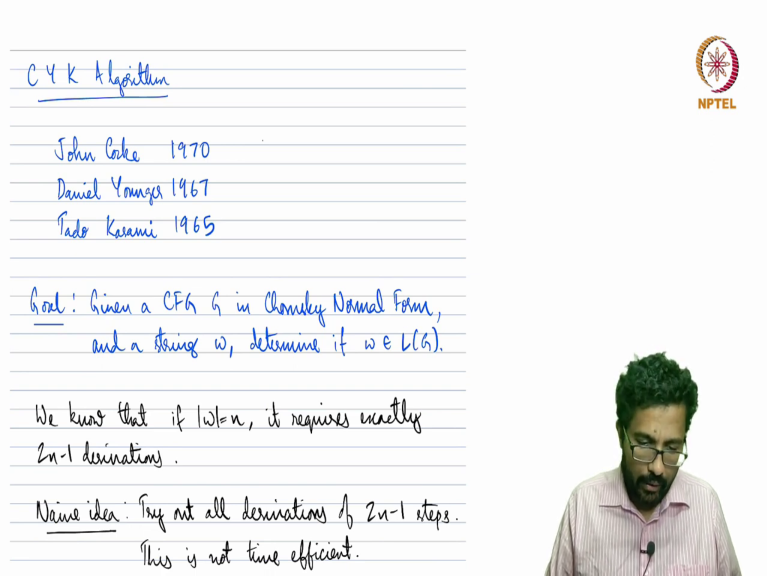
(Refer Slide Time: 18:15)



```
If $w = \varepsilon$, accept if $S \to \varepsilon$ is a rule
For $i = 1$ to $n$
        $A \in T_{ii} \iff A \to a_i$ is a rule
For $k = 1$ to $n-1$
    For $i = 1$ to $n-k$
        For $j = 0$ to $k-1$
            Check all the rules $A \to BC$
            If $T_{i,i+j}$ contains $B$, and
               $T_{i+j+1,k}$ contains $C$,
            then $T_{i,i+k} = T_{i,i+k} \cup \{A\}$
If $S \in T_{1,n}$, we say that $w \in L(G)$
```

$$A \to BC$$

$$\underbrace{w_{i,i+j} \quad w_{i+j+1,i+k}}_{w_{i,i+k}}$$



C Y K Algorithm

John Cocke 1970

Daniel Younger 1967

Tado Kasami 1965

Goal: Given a CFG G in Chomsky Normal Form, and a string w, determine if $w \in L(G)$.

We know that if $|w| = n$, it requires exactly $2n-1$ derivations.

Naive idea: Try out all derivations of $2n-1$ steps. This is not time efficient.

And because the main time consuming part is dominated by this triple for loop, the running time is order n^3. So, that is the running time and the correctness is fairly evident from the way we constructed this. That completes the explanation of the C Y K algorithm, which, given a grammar in Chomsky normal form, is used to determine whether a string is derived from the grammar. And that is it as far as lecture 18 is concerned. Thank you.