CFGs & PARSING

Recall: We showed that CFLs = languages accepted by PDAs

We used Greibach Normal Form, where every rule is of the form

$$A \longrightarrow c B_1 \cdots B_k, \text{ where}$$

$k \geq 0$, $c \in T$, $A, B_1, \ldots, B_k \in NT$.

Today: More about normal forms, and why they are useful.

Normal forms give us a uniform "shape" for a structure.

Main use of CFGs is in parsing.

Given a string ω and a CFG G, does G generate ω?

Constructing + operating over PDA is expensive.

Can we somehow directly operate over the grammar?

Normal forms help here!

Greibach Normal Form not so much, but

There is another, Chomsky Normal Form, where every rule is either of the form $A \to c$ or $A \to BC$, where $c \in T$, $A, B, C \in NT$

Thm: For every CFG G, there is a CFG G_1 in Chomsky Normal Form, and a CFG G_2 in Greibach Normal Form s.t.

$$\mathcal{L}(G_1) = \mathcal{L}(G_2) = \mathcal{L}(G) \setminus \{\varepsilon\}.$$

For any CFL L, $\mathcal{L}$ can be generated by a grammar where every rule is of the form $A \to c$, or $A \to BC$, or (if $\varepsilon \in \mathcal{L}$) $S \to \varepsilon$.

Parsing algorithm: [Cocke, Younger, Kasami : 1961]

Main idea: Determine, for each substring x of ω,
the set of all non-terminals that generate x.

grammar in Chomsky Normal Form

Needs to do this systematically:

First check all substrings of length 1, ($A \to c$?)

then all substrings of length 2 or more

Consider every possible partitioning into two parts and match against $A \to BC$, where
B generates the left part, and
C generates the right part.

Doing this repeatedly?
Use dynamic programming!

$S \to AB \mid BA \mid SS \mid AC \mid BD$

$A \to a \qquad B \to b \qquad C \to SB \qquad D \to SA$

Consider a string $\omega = |a|b|b|b|a|a|$ $\quad n = 6 = |\omega|$

markers: 0 1 2 3 4 5 6

ω_{ij} : substring of ω between markers i and j

Build a table with an entry for every (i,j), where $0 \le i < j \le n$

$j \backslash i$	0	1	2	3	4	5
1						
2						
3						
4						
5						
6						

Fill $T(i,j)$ with the non-terminals that generate ω_{ij}

$j \backslash i$	0	1	2	3	4	5
1	A					
2		B				
3			B			
4				B		
5					A	
6						A

$S \to AB \mid BA \mid SS \mid AC \mid BD$

$A \to a \qquad B \to b$

$C \to SB \qquad D \to SA$

$$\omega = {}_{0}\, a \,{}_{1}\, b \,{}_{2}\, b \,{}_{3}\, b \,{}_{4}\, a \,{}_{5}\, a \,{}_{6}$$

$\omega_{01} = a$ So $T(0,1) = \{A\}$. Similarly, $\omega_{12} = b$, so $T(1,2) = \{B\}$.

If there are multiple non-terminals which yield ω_{ij}, write them all in $T(i,j)$.

Now look at substrings of length 2.

j \ i	0	1	2	3	4	5
1	A					
2	S	B				
3		$\emptyset$	B			
4			$\emptyset$	B		
5				S	A	
6					$\emptyset$	A

$$S \to AB \mid BA \mid SS \mid AC \mid BD$$

$$A \to a \qquad B \to b$$

$$C \to SB \qquad D \to SA$$

$$\omega = {}_{0}|\,a\,|_{1}\,b\,|_{2}\,b\,|_{3}\,b\,|_{4}\,a\,|_{5}\,a\,|_{6}$$

$\omega_{02} = ab$. Break this into two substrings of length 1, $\overset{\omega_{01}}{a}$, and $\overset{\omega_{12}}{b}$. Look for all combinations of non-terminals which can yield these substrings, and look for a non-terminal which goes to this pair.

$T(0,1) = A$, $T(1,2) = B$, and $S \to AB$, so $T(0,2) = \{S\}$.

Fill in the diagonal below the top one this way.

$j \backslash i$	0	1	2	3	4	5
1	A					
2	S	B				
3	C	$\emptyset$	B			
4		$\emptyset$	$\emptyset$	B		
5			$\emptyset$	S	A	
6				D	$\emptyset$	A

$$S \to AB \mid BA \mid SS \mid AC \mid BD$$

$$A \to a \qquad B \to b$$

$$C \to SB \qquad D \to SA$$

$$\omega = {}_{0}\,a\,{}_{1}\,b\,{}_{2}\,b\,{}_{3}\,b\,{}_{4}\,a\,{}_{5}\,a\,{}_{6}$$

$\omega_{03} = abb$. There are two ways to break this string:

ω_{01} — a and bb, or ab and b.

ω_{13} $\qquad \omega_{02}$ $\qquad \omega_{23}$

$T(0,1) = A$, $T(1,3) = \emptyset$

$T(0,2) = S$, $T(2,3) = B$

$C \to SB$, so $T(0,3) = C$

Do this for the diagonal.

$j \backslash i$	0	1	2	3	4	5
1	A					
2	S	B				
3	C	$\emptyset$	B			
4	$\emptyset$	$\emptyset$	$\emptyset$	B		
5		$\emptyset$	$\emptyset$	S	A	
6			S	D	$\emptyset$	A

$$S \to AB \mid BA \mid SS \mid AC \mid BD$$

$$A \to a \qquad B \to b$$

$$C \to SB \qquad D \to SA$$

$$\omega = {}_{0}\, a \,{}_{1}\, b \,{}_{2}\, b \,{}_{3}\, b \,{}_{4}\, a \,{}_{5}\, a \,{}_{6}$$

$\omega_{04} = abbb$ $\quad$ a, bbb or ab, bb or abb, b

None of these has any possible generating non-terminals.

$\omega_{26} = bbaa$ $\quad$ Only possibility: b, baa $\quad$ B $\quad$ D

$S \to BD$. So $T(2,6) = S$.

$S \to AB \mid BA \mid SS \mid AC \mid BD$

$A \to a \qquad B \to b$

$C \to SB \qquad D \to SA$

$\omega = \,_{0}|\, a \,|_{1}\, b \,|_{2}\, b \,|_{3}\, b \,|_{4}\, a \,|_{5}\, a \,|_{6}$

j \ i	0	1	2	3	4	5
1	A					
2	S	B				
3	C	$\emptyset$	B			
4	$\emptyset$	$\emptyset$	$\emptyset$	B		
5	$\emptyset$	$\emptyset$	$\emptyset$	S	A	
6	S	$\emptyset$	S	D	$\emptyset$	A

$\omega_{05} = abbba$

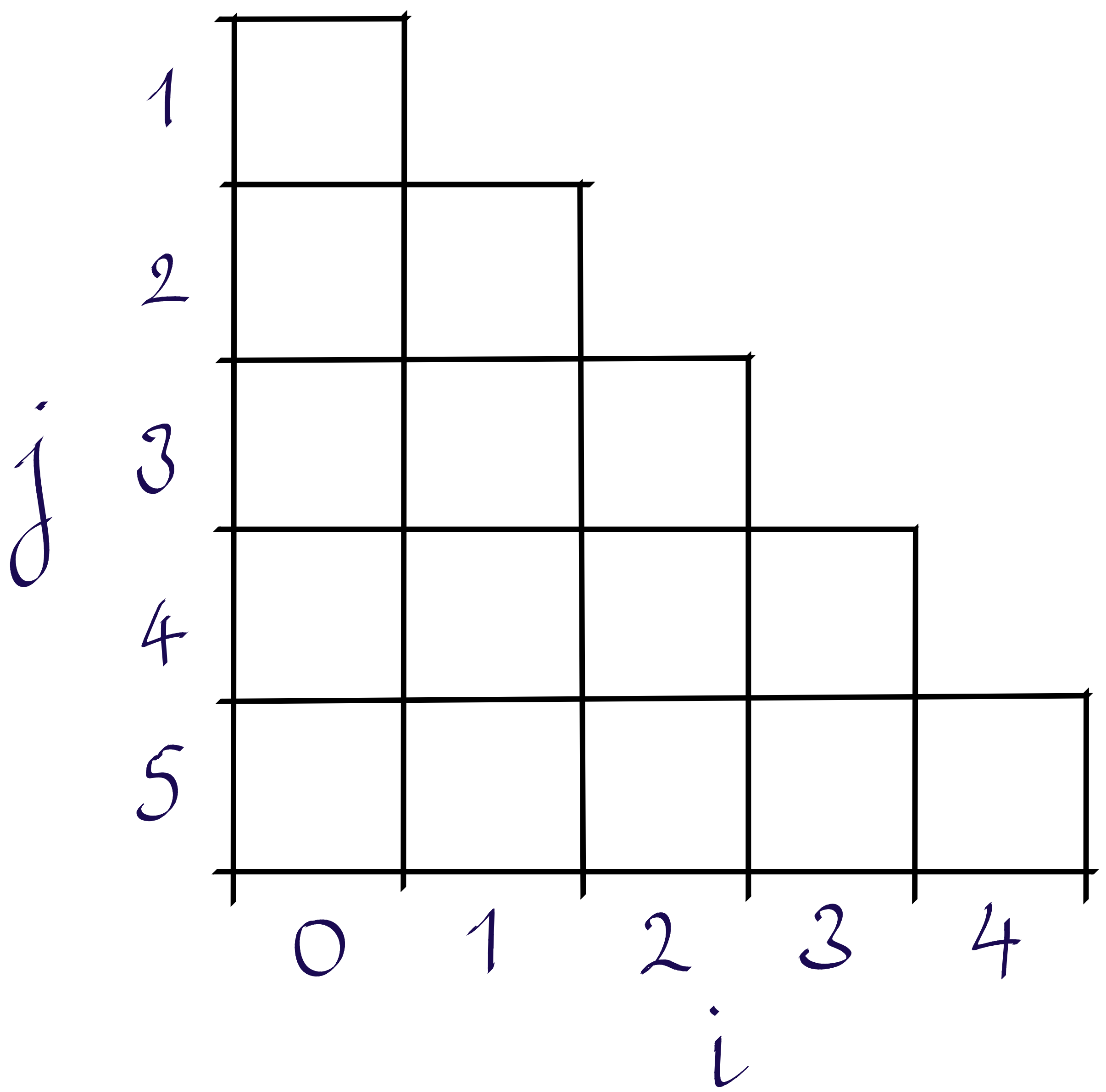
$\omega_{16} = bbbaa$

$\omega_{06} = \omega : \underbrace{ab}_{S},\ \underbrace{bbaa}_{S} \qquad S \to SS$

Since $S \in T(0,6)$, this string is generated by the grammar.

```
for i := 0 to n − 1 do                          /* first do substrings of length 1 */
  begin
    T_{i,i+1} := ∅;                             /* initially assign the empty set */
    for A → a a production of G do
      if a = x_{i,i+1} then T_{i,i+1} := T_{i,i+1} ∪ {A}
  end;
for m := 2 to n do                              /* for each length m ≥ 2 */
  for i := 0 to n − m do                        /* for each substring of length m */
    begin
      T_{i,i+m} := ∅;                           /* initially assign the empty set */
      for j := i + 1 to i + m − 1 do            /* for all breaks of the string */
        for A → BC a production of G do
          if B ∈ T_{i,j} ∧ C ∈ T_{j,i+m}
            then T_{i,i+m} := T_{i,i+m} ∪ {A}
    end;
```

If $S \to \varepsilon$ in G, it is used only if ε is provided as input;
NEVER ELSE!



$S \to AB \mid BA \mid SS \mid AC \mid BD$

$A \to a \qquad B \to b$

$C \to SB \qquad D \to SA$

$\omega = {}_{0}\,a\,{}_{1}\,b\,{}_{2}\,a\,{}_{3}\,b\,{}_{4}\,a\,{}_{5}$

https://www.youtube.com/watch?v=17re2zDBuOM