

NLP Sheet 6

1) What language does the following CFG generate

→ Specified by all the possible strings the grammar can generate.

Given Grammar

$$S \rightarrow X1X1X1X \quad \Rightarrow \quad \begin{array}{l} X \rightarrow 0X \\ X \rightarrow 1X \\ X \rightarrow \epsilon \end{array} \quad \left. \begin{array}{l} X \text{ Can be any binary} \\ \text{String} \end{array} \right\} \quad \text{Q}$$
$$X \rightarrow 0X11X1\epsilon$$
$$S \rightarrow X1X1X1X$$

• hence, S is any binary string with at least 3 1s

• hence, the language generated by the CFG is the set of all binary strings with 3 1s or more.

2) Convert the following grammar to CNF

$$S \rightarrow ASB$$

$$A \rightarrow aASAlal\epsilon$$

$$B \rightarrow SbS1Albb$$

• Recall, CNF requires

→ The grammar to have no ϵ

→ Any rule to take the form

$X \rightarrow YZ$ or $X \rightarrow m$



I. Resolve the S on RHS Violation

$$S_0 \rightarrow S$$

• S_0 is the new Start Symbol

$$S \rightarrow ASB$$

$$A \rightarrow aASAlaIE$$

$$B \rightarrow SbSIAIbb$$

2. Eliminate E from the Grammar

• i.e. eliminate $A \rightarrow E$

• Update any Production where A is on RHS

$$S \rightarrow ASB \quad \text{becomes } S \rightarrow ASBISB$$

$$A \rightarrow aASA \quad \text{becomes } A \rightarrow aASAlaSAIaASlaS$$

$$B \rightarrow A \quad \text{becomes } B \rightarrow AIE$$

• Now need to eliminate $B \rightarrow E$

• Update any Production where B is on RHS

$$S \rightarrow ASB \quad \text{becomes } S \rightarrow ASBIAI$$

$$S \rightarrow SB \quad \text{becomes } S \rightarrow SBIS$$

Thus our grammar becomes

$$S_0 \rightarrow S$$

$S \rightarrow ASBISBIAIS(IIS) \leftarrow S \rightarrow S$ is an identity (remove)

$$S \rightarrow ASBISBIAIS(IIS)$$

$$A \rightarrow aASAlaSAIaASlaS$$

$$B \rightarrow SbSIAIbb$$

and has no E while being equivalent to the original grammar

only allowed if above

3. Resolve Violations by terminals on RHS
- For each, make a dummy nonterminal
 - This violation occurred 5 times in our grammar

$$\left\{ \begin{array}{l} A \rightarrow aASAlaSAlaAslasla \\ B \rightarrow SbsIAIbb \end{array} \right.$$

becomes

$$\left\{ \begin{array}{l} Z \rightarrow a \\ Y \rightarrow b \\ A \rightarrow ZASAlZSAIZASIZSla \\ B \rightarrow SYSIAIYY \end{array} \right.$$

Only allowed if not above

4. Resolve Violations by nonterminals on RHS
- For each replace the nonterminal with what it points to

$$\left\{ \begin{array}{l} S_0 \rightarrow S \\ B \rightarrow SYSIAIYY \end{array} \right.$$

• Only two violations

become

$$S_0 \rightarrow ASBISBIAS$$

$$B \rightarrow SYSIZASAIZSAIZASIZSla IYY$$

Now our grammar is

$$S_0 \rightarrow \underline{\underline{ASBISBIAS}}$$

$$S \rightarrow \underline{\underline{ASBISBIAS}}$$

$$A \rightarrow \underline{\underline{ZASAlZSAIZASIZSla}}$$

$$B \rightarrow \underline{\underline{SYSIZASAIZSAIZASIZSla}} IYY$$

$$Z \rightarrow a$$

$$Y \rightarrow b$$

5. Resolve the Violations by 3 or more non-term. in the RHS

→ by making new nonterminals

$$E \xrightarrow{} \underset{F}{\underbrace{ABC}} \leftrightarrow E \xrightarrow{} \underset{F}{\underbrace{FC}} \quad F \xrightarrow{} AB$$

- Make the following Substitutions

$$\left. \begin{array}{l} Q \rightarrow AS \\ R \rightarrow ZQ \\ T \rightarrow SA \\ V \rightarrow SY \end{array} \right\}$$

was not a violation (don't sub.)

$$S_0 \rightarrow QBISBIAAS$$

$$S \rightarrow QBISBIAAS$$

$$A \rightarrow RAIZTIZQIZSA$$

$$B \rightarrow VSIRAIZTIZQIZSIAIYY$$

$$Z \rightarrow a$$

$$Y \rightarrow b$$

→ The CFG is now clearly in CNF

3. Given the following grammar

• Parsing ababca

$$\begin{array}{l} S \rightarrow AB \\ A \rightarrow CDICF \\ B \rightarrow CIEB \\ C \rightarrow a \\ D \rightarrow b \\ E \rightarrow C \\ F \rightarrow AD \end{array}$$

C	A	∅	∅	∅	∅	∅
9	D	∅	∅	∅	∅	∅
6	G	A	S	∅		
9	D	∅	∅			
6	B,E	∅				
C	C					
9						

neither G ∅ nor AG Prod anything
no CB, AA, ∅D

* ababca is invalid according to this grammar & hence can't be derived from it.

Every pos
45

• Try Parsing aaa bbb cc

C	\emptyset • NO CG.	\emptyset	\emptyset	\emptyset • NO CGA	A • From GF	S • From AB	S • From AB • SB, SE give nothing
a	G	\emptyset	\emptyset • NO GA	A • From GF	F • From AD	\emptyset • NO FD or FB	\emptyset NO FB
a	G	A • From CD	F • From AD	\emptyset • NO FD	\emptyset	\emptyset	\emptyset
a	D	\emptyset • NO DD	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
b	D	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
b	D	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset NO DB
b	D	\emptyset	\emptyset	\emptyset	B, E • From EB	\emptyset	\emptyset
c	B, E	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

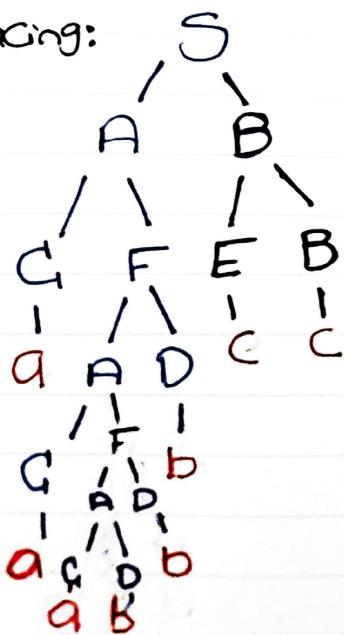
- Observe

□	●	×	●
■			■

} if any of
□, x, ● is \emptyset
then we can
skip it & check 6

- We Can Parse the Sentence
(It's grammatically Correct)
as we reached S at the root.

• By retracing:
'From'



} Only one Parse tree
for aaa bbb cc

- We never made multiple decisions
in the CKY table