# Session 17

Chomsky Normal Form

## Chomsky Normal Form (CNF)

- For any CFG $G = (V, \Sigma, S, P)$ there is a CFG $G' = (V', \Sigma, S, P')$ in CNF so that $L(G') = L(G) - \{\Lambda\}$
- A CFG is in CNF if every production is of one of these two type:

  $A \to BC$

  $A \to a$

  where $A$, $B$ and $C$ are variables, and $a$ is a terminal symbol
- If a grammar is unambiguous (i.e. it is already unambiguous or there is an unambiguous grammar generating the same language) its corresponding grammar in CNF is also unambiguous!



## Chomsky's Normal Form

- Consider $G = (V, \Sigma, S, P)$ and $S \Rightarrow^* x$ where $x \in \Sigma^*$ and $|x| = k$
  - Let $l$ be the length of a string
  - Let $t$ be the number of terminal symbols
  - For $S : l + t = 1 + 0 = 1$
  - For $x : l + t = k + k = 2k$
- If there are no $\Lambda$-*productions* (i.e. of form $A \to \Lambda$) and *Unit productions* (i.e. of form $T \to F$), for any derivation $\alpha \Rightarrow \beta$:

  $l + t$ of $\beta > l + t$ of $\alpha$

  - $\beta$ has either more variables, increasing $l$, or more terminals, increasing $t$, or both



## Chomsky's Normal Form

- An interesting property:
  - If there are no $\Lambda$-*productions* (i.e. of form $A \to \Lambda$) and *Unit productions* (i.e. of form $T \to F$), for any derivation $\alpha \Rightarrow \beta$: the value of $l + t$ is increased by rewriting a variable by a production of form:

    $A \to \gamma$ where $\gamma \in (V \cup \Sigma)^*$
  - In particular, $l + t$ increases by one if the productions have the form:

    $A \to BC$ (i.e. $l$ is increased in one)

    $A \to a$ (i.e. $t$ is increased in one)
  - So, a derivation $S \Rightarrow^* x$ (from $l + t = 1$ to $l + t = 2k$) has at most $2k - 1$ productions!



## Removing $\Lambda$-*productions*:

- A *nullable* variable in a CFG $G = (V, \Sigma, S, P)$ is defined as follows:
  - If there is a production of form $A \to \Lambda$ in $P$ then $A$ is nullable
  - If $P$ contains the production $A \to B_1B_2\ldots B_n$ where $B_1B_2\ldots B_n$ are nullable then $A$ is nullable
  - No other variables are nullable



## Removing $\Lambda$-*productions*:

- Given CFG $G = (V, \Sigma, S, P)$ construct a CFG $G1 = (V, \Sigma, S, P1)$ with no $\Lambda$-*productions* as follows:
  - Let $P1 = P$
  - Find all nullable variables in $V$
  - For every production $A \to \alpha$ in $P$, augment $P1$ with every production that can be obtained from $A \to \alpha$ by deleting one or more occurrences of nullable variables in $\alpha$
  - Delete all $\Lambda$-*productions* from $P1$, duplications of a production and productions of form $A \to A$

## Removing $\Lambda$-*productions*:

- Let the CFG G = ($V$, $\Sigma$, $S$, $P$) where $P$ are the productions:

  $S \to AACD$
  $A \to aAb \mid \Lambda$
  $C \to aC \mid a$
  $D \to aDa \mid bDb \mid \Lambda$

- Eliminating $\Lambda$-transitions: nullable variables are $A$ and $D$:

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid C$
  $A \to aAb \mid ab$
  $C \to aC \mid a$
  $D \to aDa \mid bDb \mid aa \mid bb$

- Eliminating nullvaribales in a CFG is like eliminating $\Lambda$-transitions in a NFA-$\Lambda$



## Removing unit-*productions*:

- Let the CFG G = ($V$, $\Sigma$, $S$, $P$) where $P$ has no $\Lambda$-productions:

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid C$
  $A \to aAb \mid ab$
  $C \to aC \mid a$
  $D \to aDa \mid bDb \mid aa \mid bb$

- Eliminating unit-productions: $S \to C$

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid aC \mid a$
  $A \to aAb \mid ab$
  $C \to aC \mid a$
  $D \to aDa \mid bDb \mid aa \mid bb$



## Normalizing form of productions

- Right sides only variables or only terminals:

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid aC \mid a$
  $A \to aAb \mid ab$
  $C \to aC \mid a$
  $D \to aDa \mid bDb \mid aa \mid bb$

- Replace $S \to aC$ by $S \to X_aC$ *and* $X_a \to a$:

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid X_aC \mid a$
  $A \to X_aAX_b \mid X_aX_b$
  $C \to X_aC \mid a$
  $D \to X_aDX_a \mid X_bDX_b \mid X_aX_a \mid X_bX_b$
  $X_a \to a$
  $X_b \to b$



## Obtain Chomsky normal form

- Replace $S \to ABC\alpha$ by $S \to AT$ and $T \to \mathrm{BC}\alpha$

  $S \to AACD \mid ACD \mid AAC \mid CD \mid AC \mid X_aC \mid a$
  $A \to X_aAX_b \mid X_aX_b$
  $C \to X_aC \mid a$
  $D \to X_aDX_a \mid X_bDX_b \mid X_aX_a \mid X_bX_b$
  $X_a \to a$
  $X_b \to b$

- Obtain Chomsky Normal Form:

  $S \to AT_1 \quad T_1 \to AT_2 \quad T_2 \to CD$
  $S \to AU_1 \quad U_1 \to CD$
  $S \to AV_1 \quad V_1 \to AC$



## The Grammar in CNF:

$S \to AT_1 \quad T_1 \to AT_2 \quad T_2 \to CD$
$S \to AU_1 \quad U_1 \to CD$
$S \to AV_1 \quad V_1 \to AC$
$S \to CD \mid AC \mid X_aC \mid a$
$A \to X_aW_1 \quad W_1 \to AX_b \quad A \to X_aX_b$
$C \to X_aC \mid a$
$D \to X_aY_1 \quad Y_1 \to DX_a$
$D \to X_bZ_1 \quad Z_1 \to DX_b$
$D \to X_aX_a \mid X_bX_b$
$X_a \to a$
$X_b \to b$



## An analogy!

“Ambiguous” CFG grammar $\Leftrightarrow$ *RE* or NFA-$\Lambda$

Nullable variables $\Leftrightarrow$ $\Lambda$-closure

Removing $\Lambda$-productions (e.g. $C \to aC \mid a$) $\Leftrightarrow$ NFA

CNF $\Leftrightarrow$ DFA

## Ambiguity as a device to express abstraction!

“Ambiguous” CFG grammar $\Longleftrightarrow$ *RE* or NFA-Λ

Nullable variables $\Longleftrightarrow$ Λ-closure

Removing Λ-productions $\Longleftrightarrow$ NFA

$C \to aC \mid \Lambda \;\; \Rightarrow C \to aC \mid a$

The implementation: CNF $\Longleftrightarrow$ DFA

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

## Chomsky’s Normal Form

- *Chomsky normal form* (CNF):

  $A \to BC$

  $A \to a$

- Regular grammar:

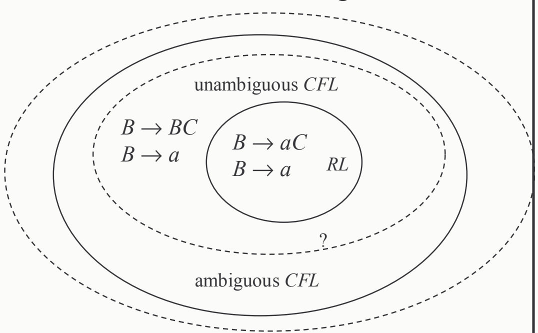
  $B \to aC$

  $B \to a$

- From regular languages to unambiguous CFL (almost!)

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003

## Is there a class of ambiguous CFL




- There is no algorithm to tell whether a grammar is ambiguous
- There is no way to tell when a language is inherently ambiguous!

Dr. Luis Pineda, IIMAS, UNAM & OSU-CIS, 2003