Chomsky Normal Form and Pushdown Automata

Mark Greenstreet, CpSc 421, Term 1, 2006/07

- Chomsky Normal Form
- Push Down Automata

Chomsky Normal Form (CNF)

- A CFG is in Chomsky Normal Form iff
  - Every rule is of the form
    - $A \longrightarrow x$, where A is a variable and x is a terminal, or
    - $A \longrightarrow BC$, where A, B and C are variables.
  - There are no rules of the form $A \longrightarrow \epsilon$ unless A is the start variable.
- We'll show that for every CFG, the is a CFG in Chomsky Normal Form that generates the same language.
- CNF is handy at times for proofs. To prove some property of CFLs we can start by writing:

  Let L be an arbitrary CFL, and let G be a CNF CFG for L

  ...

Every CFL has a CNF Grammar

- Every CFL has a CNF Grammar.

Every CFL has a CNF Grammar

- Every CFL has a CNF Grammar.
- Proof:
  - Let L be an arbitrary CFL, and let G be a CNF CFG for L.
  - QED ☺

Every CFL has a CNF Grammar

- Every CFL has a CNF Grammar.
- Proof:
  - Let L be an arbitrary CFL, and let G be a CFG for L.
  - We will find each rule of G that violates the restrictions of CNF and replace it other rules that generate the same language but satisfy CNF.
    - First, we'll remove ϵ rules: $A \longrightarrow \epsilon$, where A is not the start symbol.
    - Second, we'll remove unit rules: $A \longrightarrow B$, where A and B are variables.
    - Third, we'll convert all rules of the form $A \longrightarrow u$, where u has three or more variables or symbols into multiple rules of the form $A \longrightarrow A_1 A_2$.
    - Fourth, we'll fix all rules of the form $A \longrightarrow Bc$, $A \longrightarrow bC$ or $A \longrightarrow bc$ where A, B, and C are variables, and b and c are terminals.

Every CFL has a CNF Grammar

- Every CFL has a CNF Grammar.
- Proof:
  - Let L be an arbitrary CFL, and let G be a CFG for L.
  - We will find each rule of G that violates the restrictions of CNF and replace it other rules that generate the same language but satisfy CNF.
    - First, we'll remove ϵ rules: $A \longrightarrow \epsilon$, where A is not the start symbol.
    - Second, we'll remove unit rules: $A \longrightarrow B$, where A and B are variables.
    - Third, we'll convert all rules of the form $A \longrightarrow u$, where u has three or more variables or symbols into multiple rules of the form $A \longrightarrow A_1 A_2$.
    - Fourth, we'll fix all rules of the form $A \longrightarrow Bc$, $A \longrightarrow bC$ or $A \longrightarrow bc$ where A, B, and C are variables, and b and c are terminals.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \rightarrow S$
- Eliminate ϵ transitions.
- Remove unit rules.
- Fix rules that produce long strings.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \rightarrow S$
- Eliminate ϵ transitions.
- Remove unit rules.
- Fix rules that produce long strings.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \rightarrow S$
- ➔ Eliminate ϵ transitions. For each rule of the form $A \rightarrow \epsilon$:
  - eliminate the rule, $A \rightarrow \epsilon$, and
  - for every rule of the form $B \rightarrow uAv$, add a new rule $B \rightarrow uv$.
  - This process may produce new ϵ rules. For example, if we have the rules $A \rightarrow B$ and $B \rightarrow \epsilon$, then eliminating $B \rightarrow \epsilon$ produces the rule $A \rightarrow \epsilon$. We eliminate these new ϵ rules in the same way. Because the new rules that we get produces shorter strings of terminals and variables than the ones they were derived from, this process eventually terminates.

  Now we have a grammar where $S \rightarrow \epsilon$ is the only possible ϵ rule.
- Remove unit rules.
- Fix rules that produce long strings.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \longrightarrow S$
- Eliminate ϵ transitions.
- ➔ Remove unit rules. If $A \longrightarrow B$ and B is a variable:
  - eliminate the rule, $A \longrightarrow B$,
  - for every rule of the form $C \longrightarrow uAv$, add a new rule $C \longrightarrow uBv$.

  Now we have a grammar where $S_0 \longrightarrow \epsilon$ is the only possible ϵ rule and every rule produces a string of one terminal, or at least two terminals and/or variables.
- Fix rules that produce long strings.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \rightarrow S$
- Eliminate ϵ transitions.
- Remove unit rules.
- Fix rules that produce long strings.
  $A \rightarrow u_1 u_2 u_3 \ldots u_k$ becomes $A \rightarrow u_1 A_2$, $A_2 \rightarrow u_2 A_3$, $\ldots A_{k-1} \rightarrow u_{k-1} u_k$.
  Now, each rule produces a single terminal or a string of length two. or is the rule $S_0 \rightarrow \epsilon$.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \longrightarrow S$
- Eliminate ϵ transitions.
- Remove unit rules.
- Fix rules that produce long strings.
- ➔ Replace terminals with variables. For each rule $A \longrightarrow Bc$ or $A \longrightarrow bC$ where B and C are variables and b and c are terminals:
  - replace $A \longrightarrow Bc$ with $A \longrightarrow BU_c$, $A \longrightarrow bC$ with $A \longrightarrow U_bC$, and $A \longrightarrow BC$ with $A \longrightarrow U_bU_b$.
  - Introduce new rules: $U_b \longrightarrow b$, etc. $C \longrightarrow uAv$,

  Now, each rule produces two variables, one terminal, or is the rule $S_0 \longrightarrow \epsilon$.
- We have produced a CNF grammar that generates L.
  $\square$.

Proof: $\forall$ CFL $\exists$ a CNF Grammar

- Let L be a language, and let G be a CFG for L with start variable S.
- Introduce a new start variable, S_0, and the rule $S_0 \rightarrow S$
- Eliminate ϵ transitions.
- Remove unit rules.
- Fix rules that produce long strings.
- Replace terminals with variables.
- We have produced a CNF grammar that generates L.
  $\square$.

Example: Converting to CNF

- The original grammar:

$$\begin{array}{lcll} S & \longrightarrow & ASA & \mid \ \mathsf{a}B \\ A & \longrightarrow & B & \mid \ S \\ B & \longrightarrow & \mathsf{b} & \mid \ \epsilon \end{array}$$

Example: Converting to CNF

- Introduce a new start variable, S_0:

$$
\begin{array}{rcll|rcll}
 & & & & S_0 & \rightarrow & S & \\
S & \rightarrow & ASA & \mid \; \mathrm{a}B & S & \rightarrow & ASA & \mid \; \mathrm{a}B \\
A & \rightarrow & B & \mid \; S & A & \rightarrow & B & \mid \; S \\
B & \rightarrow & \mathrm{b} & \mid \; \epsilon & B & \rightarrow & \mathrm{b} & \mid \; \epsilon
\end{array}
$$

Example: Converting to CNF

- Eliminate ϵ rules:

$$\begin{array}{rcllll|rcllllll} S_0 & \to & S & & & & S_0 & \to & S & & & & \\ S & \to & ASA & \mid & \mathsf{a}B & & S & \to & ASA & \mid & \mathsf{a}B & \mid & \mathsf{a} \\ A & \to & B & \mid & S & & A & \to & B & \mid & S & \mid & \epsilon \\ B & \to & \mathsf{b} & \mid & \epsilon & & B & \to & \mathsf{b} & & & & \end{array}$$

Example: Converting to CNF

- Eliminate ϵ rules:

$$
\begin{array}{lll|lll}
S_0 & \rightarrow & S & S_0 & \rightarrow & S \\
S & \rightarrow & ASA \mid \text{a}B \mid \text{a} & S & \rightarrow & ASA \mid \text{a}B \mid \text{a} \mid \\
 & & & & & AS \mid SA \mid S \\
A & \rightarrow & B \mid S \mid \epsilon & A & \rightarrow & B \mid S \\
B & \rightarrow & \text{b} & B & \rightarrow & \text{b}
\end{array}
$$

Example: Converting to CNF

- Eliminate unit rules:

$$
\begin{array}{lcllllllll|lcllllllll}
S_0 & \rightarrow & S & & & & & & & & S_0 & \rightarrow & ASA & \mid & \mathrm{a}B & \mid & \mathrm{a} & \mid \\
 & & & & & & & & & & & & AS & \mid & SA & & & \\
S & \rightarrow & ASA & \mid & \mathrm{a}B & \mid & \mathrm{a} & \mid & & & S & \rightarrow & ASA & \mid & \mathrm{a}B & \mid & \mathrm{a} & \mid \\
 & & AS & \mid & SA & \mid & S & & & & & & AS & \mid & SA & & & \\
A & \rightarrow & B & \mid & S & & & & & & A & \rightarrow & \mathrm{b} & \mid & ASA & \mid & \mathrm{a}B & \mid \\
 & & & & & & & & & & & & \mathrm{a} & \mid & AS & \mid & SA & \\
B & \rightarrow & \mathrm{b} & & & & & & & & B & \rightarrow & \mathrm{b} & & & & &
\end{array}
$$

Example: Converting to CNF

- Fix rules that produce long strings:

$$
\begin{array}{rcllllll}
S_0 & \rightarrow & ASA & \mid & \mathsf{a}B & \mid & \mathsf{a} & \mid \\
 & & AS & \mid & SA & & & \\
S & \rightarrow & ASA & \mid & \mathsf{a}B & \mid & \mathsf{a} & \mid \\
 & & AS & \mid & SA & & & \\
A & \rightarrow & \mathsf{b} & \mid & AA_1 & \mid & \mathsf{a}B & \mid \\
 & & \mathsf{a} & \mid & AS & \mid & SA & \\
 & & & & & & & \\
B & \rightarrow & \mathsf{b} & & & & &
\end{array}
\quad\Bigg|\quad
\begin{array}{rcllllll}
S_0 & \rightarrow & AA_1 & \mid & \mathsf{a}B & \mid & \mathsf{a} & \mid \\
 & & AS & \mid & SA & & & \\
S & \rightarrow & AA_1 & \mid & \mathsf{a}B & \mid & \mathsf{a} & \mid \\
 & & AS & \mid & SA & & & \\
A & \rightarrow & \mathsf{b} & \mid & AA_1 & \mid & \mathsf{a}B & \\
 & & \mathsf{a} & \mid & AS & \mid & SA & \\
A_1 & \rightarrow & SA & & & & & \\
B & \rightarrow & \mathsf{b} & & & & &
\end{array}
$$

Example: Converting to CNF

- Replace terminals with variables:

$$
\begin{array}{rclllllll}
S_0 & \rightarrow & AA_1 & \mid & \text{a}B & \mid & \text{a} & \mid \\
 & & AS & \mid & SA & & & \\
S_0 & \rightarrow & AA_1 & \mid & \text{a}B & \mid & \text{a} & \mid \\
 & & AS & \mid & SA & & & \\
A_1 & \rightarrow & SA & & & & & \\
A & \rightarrow & \text{b} & \mid & AA_1 & \mid & \text{a}B & \mid \\
 & & \text{a} & \mid & AS & \mid & SA & \\
A_1 & \rightarrow & SA & & & & & \\
 & & & & & & & \\
B & \rightarrow & \text{b} & & & & &
\end{array}
$$

$$
\begin{array}{rclllllll}
S_0 & \rightarrow & AA_1 & \mid & U_aB & \mid & \text{a} & \mid \\
 & & AS & \mid & SA & & & \\
S_0 & \rightarrow & AA_1 & \mid & U_aB & \mid & \text{a} & \mid \\
 & & AS & \mid & SA & & & \\
A_1 & \rightarrow & SA & & & & & \\
A & \rightarrow & \text{b} & \mid & AA_1 & \mid & U_aB & \\
 & & \text{a} & \mid & AS & \mid & SA & \\
A_1 & \rightarrow & SA & & & & & \\
U_a & \rightarrow & \text{a} & & & & & \\
B & \rightarrow & \text{b} & & & & &
\end{array}
$$