Chomsky Normal Form

Venkatesh Choppella

April 11, 2019

Outline

Chomsky Normal Form

Definition of CNF

A grammar $G = (V, \Sigma, R, S)$ is in Chomsky Normal Form (CNF) if every production is of the form

- $A \to BC$,
- $A \to a$, or
- $S \to \epsilon$.

where

- $B, C \in V \setminus \{S\}$,
- $a \in \Sigma$

Why is CNF useful?

- CNF's are useful because they are *canonical*:
- Every CFG G is convertible to a grammar G' such that
  - G' is in CNF
  - $L(G) = L(G')$
- All parse trees are binary trees. A derivation for s has length $2|s| - 1$ steps, where $s \neq \epsilon$. The derivation for ϵ, if in the language, takes 1 step.
- Membership determined by standard algorithm (*CYK* algorithm).

Example Grammar in CNF

$$G = (\{S, A, B\}, \{a, b\}, R, S)$$

where R is:

$$S \to \epsilon \mid AB$$
$$A \to AA \mid a$$
$$B \to BB \mid BA \mid b$$

Rules for transforming a CFG into CNF via example

Based on Example 2.10 of Sipser 3ed (where it is called G_6).

Initial Grammar G_0

$$S \to A\ S\ A \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \epsilon$$

Transformations

- START: Introduce new start symbol.
- BIN: Ensure rhs of each rule has only two variables.
- DEL: Remove ϵ productions.
- UNIT: Remove unit productions
- TERM: replace non-solitary terminals with nonterminals

START transformation

- 1. Introduce new start symbol S_0 not occurring in G_0.
- 2. Introduce new production $S_0 \to S$.
- 3. This ensures S_0 doesn't occur in the RHS of any production.

Grammar G_1

$$\mathbf{S_0} \to \mathbf{S}$$
$$S \to A\ S\ A \mid aB$$
$$A \to B \mid S$$
$$B \to b \mid \epsilon$$

BIN transformation

- 1. Introduce auxiliary variables and
- 2. transform each rhs of length greater than 2
- 3. so that rhs of each rule is of size atmost 2.

Grammar G_2

$$S_0 \to S$$
$$S \to \mathbf{A}\ \mathbf{A_1} \mid aB$$
$$\mathbf{A_1} \to S\ A$$
$$A \to B \mid S$$
$$B \to b \mid \epsilon$$

Nullable variables in a grammar

- 1. If $X \rightarrow \epsilon$, then X is nullable
- 2. If $Y \rightarrow X_1 \dots X_n$ and each X_i is nullable, then Y is nullable.

DEL transformation Part 1

- 1. Identify nullable variables ($\{B, A\}$).
- 2. for each nullable variable X occurring in the rhs of a production, create a new production replacing that occurrence with ϵ.
- 3. Eliminate all ϵ productions so that the grammar has atmost one ϵ production, from S_0.

Grammar $G_{3.1}$

$$
\begin{aligned}
S_0 &\to S \\
S &\to A\ A_1 \mid \mathbf{A_1} \mid aB \mid \mathbf{a} \\
A_1 &\to S\ A \mid \mathbf{S} \\
A &\to B \mid \epsilon \mid S \\
B &\to b \mid \epsilon
\end{aligned}
$$

DEL transformation Part 2

- 1. Eliminate all ϵ productions so that the grammar has atmost one ϵ production, from S_0, if necessary.

Grammar $G_{3.2}$ (or G_3)

$$
\begin{aligned}
S_0 &\rightarrow S \\
S &\rightarrow A\ A_1 \mid A_1 \mid aB \mid a \\
A_1 &\rightarrow S\ A \mid S \\
A &\rightarrow B \mid \cancel{\epsilon} \mid S \\
B &\rightarrow b \mid \cancel{\epsilon}
\end{aligned}
$$

UNIT xformation Part 1: locate unit rules

- 1. Locate unit rules of the form $X \to Y$.
- 2. For each existing rule $Y \to Z_1 \dots Z_n$, add the rule $X \to Z_1 \dots Z_n$, unless $Y \to Z_1$ is already a unit rule.
- 3. Remove all the unit rules.

Unit rules in Grammar G_3

Unit rules

$$
\begin{aligned}
S_0 &\to S \\
S &\to A\ A_1 \mid A_1 \mid aB \mid a \\
A_1 &\to S\ A \mid S \\
A &\to B \mid S \\
B &\to b
\end{aligned}
$$

UNIT xformation Part 2: inline rhs of unit rules

- 2. For each existing rule $Y \rightarrow Z_1 \ldots Z_n$, add the rule $X \rightarrow Z_1 \ldots Z_n$, unless $Y \rightarrow Z_1$ is already a unit rule.

Grammar $G_{4.1}$

$$S_0 \rightarrow \mathbf{A\ A_1} \mid \mathbf{a\ B} \mid \mathbf{a} \mid \cancel{S}$$

$$S \rightarrow A\ A_1 \mid \cancel{A_1} \mid \mathbf{S\ A} \mid aB \mid a$$

$$A_1 \rightarrow S\ A \mid S \mid \mathbf{A\ A_1} \mid \mathbf{aB} \mid \mathbf{a}$$

$$A \rightarrow \cancel{B} \mid \cancel{S} \mid \mathbf{b}$$

$$\mid \mathbf{A\ A_1} \mid \mathbf{S\ A} \mid \mathbf{aB} \mid \mathbf{a}$$

$$B \rightarrow b$$

UNIT xformation Part 3: eliminate unit rules

- 3. Remove all unit rules.

Grammar $G_{4.2}$ **(**G_4**)**

$$
\begin{aligned}
S_0 &\rightarrow A\ A_1 \mid a\ B \mid a \\
S &\rightarrow A\ A_1 \mid S\ A \mid aB \mid a \\
A_1 &\rightarrow S\ A \mid A\ A_1 \mid aB \mid a \\
A &\rightarrow b \\
&\qquad \mid A\ A_1 \mid S\ A \mid aB \mid a \\
B &\rightarrow b
\end{aligned}
$$

TERM transformation

- 1. Identify non-solitary terminal productions: of the form $X \rightarrow \alpha\ a\ \beta$, where $\alpha\beta \neq \epsilon$.
- 2. Create new non-terminal N_a and a production $N_a \rightarrow a$.
- 3. Replace each non-solitary terminal production with $X \rightarrow \alpha\ N_a\ \beta$.

TERM Part 1: Identify non-solitary terminals

- Identify non-solitary terminal productions.

Grammar G_4

$$
\begin{aligned}
S_0 &\to A\ A_1 \mid a\ B \mid a \\
S &\to A\ A_1 \mid S\ A \mid aB \mid a \\
A_1 &\to S\ A \mid A\ A_1 \mid aB \mid a \\
A &\to b \\
&\qquad \mid A\ A_1 \mid S\ A \mid aB \mid a \\
B &\to b
\end{aligned}
$$

TERM Part 2: Create new productions

- 2. Create new non-terminal N and a production $N \rightarrow a$.

Grammar $G_{5.1}$

$$
\begin{aligned}
S_0 &\rightarrow A\ A_1 \mid a\ B \mid a \\
S &\rightarrow A\ A_1 \mid S\ A \mid aB \mid a \\
A_1 &\rightarrow S\ A \mid A\ A_1 \mid aB \mid a \\
A &\rightarrow b \\
&\qquad \mid A\ A_1 \mid S\ A \mid aB \mid a \\
B &\rightarrow b \\
\mathbf{N} &\rightarrow \mathbf{a}
\end{aligned}
$$

TERM Part 3: Replace non-solitary terminal productions

- 3. Replace each non-solitary terminal production $X \rightarrow \alpha a \beta$ with

$$X \rightarrow \alpha N_a \beta$$

Grammar $G_{5.2}$ **(**G_5**)**

$$
\begin{aligned}
S_0 &\rightarrow A\ A_1 \mid N\ B \mid a \\
S &\rightarrow A\ A_1 \mid S\ A \mid N\ B \mid a \\
A_1 &\rightarrow S\ A \mid A\ A_1 \mid N\ B \mid a \\
A &\rightarrow b \\
&\quad \mid A\ A_1 \mid S\ A \mid N\ B \mid a \\
B &\rightarrow b \\
\mathbf{N} &\rightarrow \mathbf{a}
\end{aligned}
$$

Ordering of transformations matters

The order is important!

- *START* results in constant increment to grammar size.
- *BIN* before *DEL* gives linear increase in grammar size.
- *DEL* before *BIN* could result in exponential blowup of resultant grammar.
- *UNIT* causes quadratic increase.
- *TERM* causes linear increase.

$$\mathit{START} \rightarrow \mathit{BIN} \rightarrow \mathit{DEL} \rightarrow \mathit{UNIT} \rightarrow \mathit{TERM}$$

gives overall quadratic increase.

Comparison with Example 2.10 in Sipser 3ed.

- The book does not name the transformations.
- It seems to use *DEL* (Step 2 in Example 2.10) before $BIN (Step 4.).
- Total 19 productions (compared to 18 here).