CMSC 141 Automata and Language Theory
Context-Free Languages

Mark Froilan B. Tandoc

October 15, 2014

Simplifying Context Free Grammars

- Chomsky Normal Form
  - $V \rightarrow (T + VV)$
- Greibach Normal Form
  - $V \rightarrow TV^*$
- Elimination of unit productions ($V \rightarrow W$), and empty productions ($V \rightarrow \varepsilon$) except for the start state

Chomsky Normal Form

A context-free grammar in Chomsky Normal Form (CNF) have rules of the form:

$A \rightarrow BC$
$A \rightarrow a$

where a is any terminal and A, B, C are any variables - except that B and C may not be the start variable. Also, only the start variable (say S), can have the rule $S \rightarrow \varepsilon$

Chomsky Normal Form

Convert the grammar to CNF

Chomsky Normal Form

Convert the grammar to CNF

Grammar

$$
\begin{aligned}
S &\rightarrow ASA \mid aB \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Chomsky Normal Form

Add a new start variable (say S_0) and have the rule $S_0 \rightarrow S$ where S is the original start state

Grammar

$$
\begin{aligned}
S &\rightarrow ASA \mid aB \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Chomsky Normal Form

Add a new start variable (say S_0) and have the rule $S_0 \to S$ where S is the original start state

Grammar

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

Chomsky Normal Form

Remove the ε rules. $B \rightarrow \varepsilon$

Grammar

$$
\begin{aligned}
S_0 &\rightarrow S \\
S &\rightarrow ASA \mid aB \\
A &\rightarrow B \mid S \\
B &\rightarrow b \mid \varepsilon
\end{aligned}
$$

Chomsky Normal Form

Remove the ε rules. $B \rightarrow \varepsilon$

Grammar

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

Chomsky Normal Form

Remove the ε rules. $A \to \varepsilon$

Grammar

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

Chomsky Normal Form

Remove the ε rules. $A \to \varepsilon$

Grammar

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

Chomsky Normal Form

Remove unit rules. $S \rightarrow S$

Grammar

$$
\begin{array}{ll}
S_0 & \rightarrow S \\
S & \rightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S \\
A & \rightarrow B \mid S \\
B & \rightarrow b
\end{array}
$$

Chomsky Normal Form

Remove unit rules. $S \to S$

Grammar

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

Chomsky Normal Form

Remove unit rules. $S_0 \to S$

Grammar

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

Chomsky Normal Form

Remove unit rules. $S_0 \to S$

Grammar

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

Chomsky Normal Form

Remove unit rules. $A \rightarrow B$

Grammar

$$
\begin{aligned}
S_0 &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
S &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
A &\rightarrow B \mid S \\
B &\rightarrow b
\end{aligned}
$$

Chomsky Normal Form

Remove unit rules. $A \to B$

Grammar

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

Chomsky Normal Form

Remove unit rules. $A \rightarrow S$

Grammar

$$
\begin{aligned}
S_0 &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
S &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
A &\rightarrow S \mid b \\
B &\rightarrow b
\end{aligned}
$$

Chomsky Normal Form

Remove unit rules. $A \to S$

Grammar

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

Chomsky Normal Form

Convert the remaining rules into proper form by adding additional variables and rules

Grammar

$$
\begin{aligned}
S_0 &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
S &\rightarrow ASA \mid aB \mid a \mid SA \mid AS \\
A &\rightarrow b \mid ASA \mid aB \mid a \mid SA \mid AS \\
B &\rightarrow b
\end{aligned}
$$

Chomsky Normal Form

Convert the remaining rules into proper form by adding additional variables and rules

Grammar

$$
\begin{aligned}
S_0 &\to AA_1 \mid UB \mid a \mid SA \mid AS \\
S &\to AA_1 \mid UB \mid a \mid SA \mid AS \\
A &\to b \mid AA_1 \mid UB \mid a \mid SA \mid AS \\
A_1 &\to SA \\
U &\to a \\
B &\to b
\end{aligned}
$$

Chomsky Normal Form

Convert to CNF

$$
\begin{aligned}
S &\to ab \\
S &\to aSb
\end{aligned}
$$

CNF

$$
\begin{aligned}
S &\to AB \mid XB \\
X &\to AY \\
Y &\to AB \mid XB \\
A &\to a \\
B &\to b
\end{aligned}
$$

Greibach Normal Form

A context-free grammar in Greibach Normal Form (GNF) have rules of the form:

$V \rightarrow TV^*$

where V can be any variable and T can be any terminal. Only the start variable (say S), can have the rule $S \rightarrow \varepsilon$

Greibach Normal Form

Convert the grammar to GNF

$S \quad \rightarrow a \mid S + S$

GNF

$S \quad \rightarrow a$

$S \quad \rightarrow aPS$ (this makes $+$ right-associative)

$P \quad \rightarrow +$

When in GNF, an input string of length n can always be derived in n steps

Equivalence of PDAs and CFGs

Theorem

Every CFG can be converted into an equivalent PDA, and vice versa

Note that we are referring to non-deterministic PDA (NPDA) because deterministic PDA are weaker than NPDA

CFG TO NPDA

Idea:

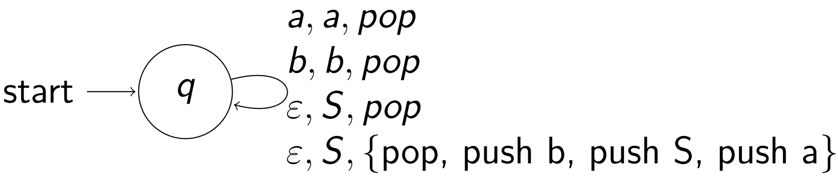
- Use a single state q, with stack alphabet $\Gamma = V \cup T$, and the PDA is accepting by empty stack
- Initial stack symbol is the start variable
- For every terminal symbol a in Σ, add the transition $\delta(q, a, a) = (q, pop)$
- For every empty production $A \to \varepsilon$, add the transition $\delta(q, \varepsilon, A) = (q, pop)$
- For every rule $A \to B_1 B_2 \ldots B_n$, add the transition $\delta(q, \varepsilon, A) = (q, \{pop; pushB_n; pushB_{n-1}; \ldots; pushB_1\})$

CFG to NPDA

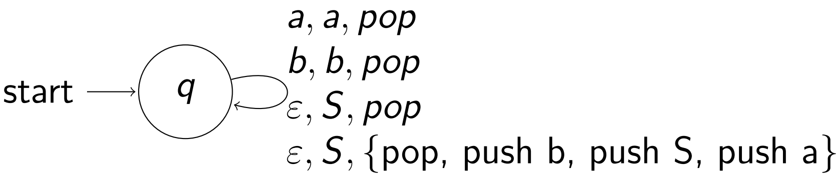
$$S \to \varepsilon \mid aSb \Longrightarrow L(G) = \{a^n b^n : n \geq 0\}$$

Stack alphabet:
$\Gamma = \{S, a, b\}$

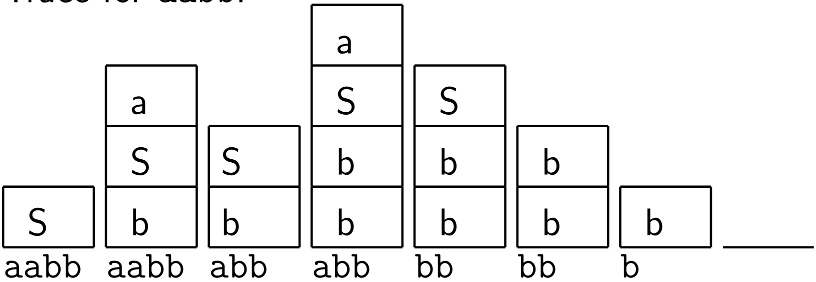
Initial stack symbol: S



CFG to NPDA



Trace for `aabb`:



References

- Previous slides on CMSC 141
- M. Sipser. Introduction to the Theory of Computation. Thomson, 2007.
- J.E. Hopcroft, R. Motwani and J.D. Ullman. Introduction to Automata Theory, Languages and Computation. 2nd ed, Addison-Wesley, 2001.
- E.A. Albacea. Automata, Formal Languages and Computations, UPLB Foundation, Inc. 2005
- JFLAP, www.jflap.org
- Various online LaTeX and Beamer tutorials