CSCI 340: Computational Models

CFG = PDA
Building a PDA for Every CFG

Theorem

*Given a CFG that generates the language L, there is a PDA that accepts exactly L*

Theorem

*Given a PDA that accepts the language L, there exists a CFG that accepts exactly L*

Both of these theorems were discovered independently by Schützenberger, Chomsky, and Evey
CFG to PDA Algorithm

Note: We assume the CFG grammar is defined in CNF

Two forms:

Handling form $N_i \rightarrow N_j N_k$:

Handling form $N_i \rightarrow t$:

Note: non-terminals are pushed in reverse order
CFG to PDA Algorithm

Start of machine:

START → PUSH S → POP

End of machine:

POP → \( \Delta \) READ → \( \Delta \) ACCEPT

If a language should accept \( \lambda \), include:

POP \( \xrightarrow{S} \) POP
Consider the following grammar (in CNF):

\[
\begin{align*}
S & \rightarrow SB \\
S & \rightarrow AB \\
A & \rightarrow CC \\
B & \rightarrow b \\
C & \rightarrow a
\end{align*}
\]
Example

START

PUSH S

READ

PUSH B

PUSH S

PUSH S

PUSH A

PUSH C

READ

PUSH B

PUSH C

ACCEPT

PUSH C

POP

\[ a \]

\[ b \]

\[ \Delta \]

\[ C \]

\[ S \]

\[ 5 / 6 \]
“This is a long proof by constructive algorithm. In fact, it is unquestionably the most torturous proof in the book; parental consent is required”

Pages 327 – 347
PDA to CFG

“This is a long proof by constructive algorithm. In fact, it is unquestionably the most torturous proof in the book; parental consent is required”

Pages 327 – 347