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

\[ X_1 \rightarrow X_2 X_3 \]
\[ X_1 \rightarrow X_3 X_4 \]
\[ X_2 \rightarrow X_2 X_2 \]
\[ \ldots \]
\[ X_3 \rightarrow a \]
\[ X_4 \rightarrow a \]
\[ X_5 \rightarrow b \]
\[ \ldots \]

**Two forms:**

\[ N_i \rightarrow N_j N_k \]
\[ N_i \rightarrow t \]

**Handling form** \( N_i \rightarrow N_j N_k \):

- POP
- PUSH \( N_j \)
- PUSH \( N_k \)

Note: non-terminals are pushed in reverse order

**Handling form** \( N_i \rightarrow t \):

- POP
- READ

2 / 6
CFG to PDA Algorithm

Start of machine:

Start:
START \(\rightarrow\) PUSH S \(\rightarrow\) POP

End of machine:

End:
POP \(\Delta\) \(\rightarrow\) READ \(\Delta\) \(\rightarrow\) ACCEPT

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

If:
PUSH S
Example

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 B

PUSH A

PUSH C

READ

PUSH C

POP

ACCEPT

\[ a \quad B \quad b \quad \Delta \quad A \quad \Delta \]
“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