Section: Parsing

Parsing: Deciding if $x \in \Sigma^*$ is in $L(G)$ for some CFG $G$.

Consider the CFG $G$:

$$
S \rightarrow Aa \\
A \rightarrow AA \mid ABa \mid \lambda \\
B \rightarrow BBa \mid b \mid \lambda
$$

Is $ba$ in $L(G)$? Running time?

New grammar $G'$ is:

$$
S \rightarrow Aa \mid a \\
A \rightarrow AA \mid ABa \mid Aa \mid Ba \mid a \\
B \rightarrow BBa \mid Ba \mid a \mid b
$$

Is $ba$ in $L(G)$? Running time?
Top-down Parser:

- Start with S and try to derive the string.

\[
S \rightarrow aS \mid b
\]

- Examples: LL Parser, Recursive Descent
Bottom-up Parser:

- Start with string, work backwards, and try to derive $S$.

- Examples: Shift-reduce, Operator-Precedence, LR Parser
The function FIRST:

\[ G = (V, T, S, P) \]
\[ w, v \in (V \cup T)^* \]
\[ a \in T \]
\[ X, A, B \in V \]
\[ X_I \in (V \cup T)^+ \]

Definition: \( \text{FIRST}(w) \) = the set of terminals that begin strings derived from \( w \).

\[ \text{If } w \xrightarrow{*} av \text{ then } \]
\[ a \text{ is in FIRST}(w) \]
\[ \text{If } w \xrightarrow{*} \lambda \text{ then } \]
\[ \lambda \text{ is in FIRST}(w) \]
To compute FIRST:

1. \( \text{FIRST}(a) = \{a\} \)

2. \( \text{FIRST}(X) \)
   
   (a) If \( X \rightarrow aw \) then
       \( a \) is in \( \text{FIRST}(X) \)
   
   (b) IF \( X \rightarrow \lambda \) then
       \( \lambda \) is in \( \text{FIRST}(X) \)
   
   (c) If \( X \rightarrow Aw \) and \( \lambda \in \text{FIRST}(A) \)
       then
       Everything in \( \text{FIRST}(w) \) is in \( \text{FIRST}(X) \)
3. In general, FIRST($X_1 X_2 X_3 ... X_K$) =

- FIRST($X_1$)
- $\cup$ FIRST($X_2$) if $\lambda$ is in FIRST($X_1$)
- $\cup$ FIRST($X_3$) if $\lambda$ is in FIRST($X_1$) and $\lambda$ is in FIRST($X_2$)
  ...
- $\cup$ FIRST($X_K$) if $\lambda$ is in FIRST($X_1$) and $\lambda$ is in FIRST($X_2$) ...
  and $\lambda$ is in FIRST($X_{K-1}$)
- $\{\lambda\}$ if $\lambda \notin$ FIRST($X_J$) for all $J$
Example:

\[
S \rightarrow aSc \mid B \\
B \rightarrow b \mid \lambda
\]

\[
\text{FIRST}(B) = \\
\text{FIRST}(S) = \\
\text{FIRST}(Sc) =
\]
Example

\[
\begin{align*}
S & \rightarrow \text{BCD} \mid aD \\
A & \rightarrow \text{CEB} \mid aA \\
B & \rightarrow b \mid \lambda \\
C & \rightarrow dB \mid \lambda \\
D & \rightarrow cA \mid \lambda \\
E & \rightarrow e \mid fE
\end{align*}
\]

FIRST(S) = 
FIRST(A) = 
FIRST(B) = 
FIRST(C) = 
FIRST(D) = 
FIRST(E) =
Definition: $\text{FOLLOW}(X) = \text{set of terminals that can appear to the right of } X \text{ in some derivation.}$

If $S \Rightarrow^* wAav$ then

$a$ is in $\text{FOLLOW}(A)$

To compute $\text{FOLLOW}$:

1. $\$\text{ is in } \text{FOLLOW}(S)$

2. If $A \rightarrow wBv$ and $v \neq \lambda$ then
   $\text{FIRST}(v) - \{\lambda\}$ is in $\text{FOLLOW}(B)$

3. IF $A \rightarrow wB$ OR
   $A \rightarrow wBv$ and $\lambda$ is in $\text{FIRST}(v)$
   then
   $\text{FOLLOW}(A)$ is in $\text{FOLLOW}(B)$

4. $\lambda$ is never in $\text{FOLLOW}$
Example:

\[
S \rightarrow aSc \mid B \\
B \rightarrow b \mid \lambda
\]

FOLLOW(S) = 

FOLLOW(B) =
Example:

\[ S \rightarrow BCD \mid aD \]
\[ A \rightarrow CEB \mid aA \]
\[ B \rightarrow b \mid \lambda \]
\[ C \rightarrow dB \mid \lambda \]
\[ D \rightarrow cA \mid \lambda \]
\[ E \rightarrow e \mid fE \]

\text{FOLLOW}(S) =
\text{FOLLOW}(A) =
\text{FOLLOW}(B) =
\text{FOLLOW}(C) =
\text{FOLLOW}(D) =
\text{FOLLOW}(E) =