Section: Parsing

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

Consider the CFG $G$:

\[
\begin{align*}
S & \rightarrow Aa \\
A & \rightarrow AA \mid ABa \mid \lambda \\
B & \rightarrow BBa \mid b \mid \lambda
\end{align*}
\]

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

New grammar $G'$ is:

\[
\begin{align*}
S & \rightarrow Aa \mid a \\
A & \rightarrow AA \mid ABa \mid Aa \mid Ba \mid a \\
B & \rightarrow BBa \mid Ba \mid a \mid b
\end{align*}
\]

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

- Start with $S$ and try to derive the string.

\[ S \to aS \mid b \]

- 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 \text{FIRST}:

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

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

\[ \text{If } w \Rightarrow^* av \text{ then} \]
\[ a \text{ is in FIRST}(w) \]
\[ \text{If } w \Rightarrow^* \lambda \text{ then} \]
\[ \lambda \text{ is in FIRST}(w) \]
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, \( \text{FIRST}(X_1X_2X_3..X_K) = \)

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

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

FIRST(B) = 
FIRST(S) =
FIRST(Sc) =
Example

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

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

If \( S \xrightarrow{*} wAav \) then
   \( a \) is in \( \text{FOLLOW}(A) \)

To compute \( \text{FOLLOW} \):

1. \( \$ \) 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:

\begin{align*}
S & \rightarrow aSc \mid B \\
B & \rightarrow b \mid \lambda
\end{align*}

\text{FOLLOW}(S) = \\
\text{FOLLOW}(B) =
Example:

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

\[
\begin{align*}
\text{FOLLOW}(S) & = \\
\text{FOLLOW}(A) & = \\
\text{FOLLOW}(B) & = \\
\text{FOLLOW}(C) & = \\
\text{FOLLOW}(D) & = \\
\text{FOLLOW}(E) & =
\end{align*}
\]