Section: Parsing

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

Consider the CFG G:

$$\begin{aligned} S &\rightarrow Aa \\ A &\rightarrow AA \mid ABa \mid \lambda \\ B &\rightarrow BBa \mid b \mid \lambda \end{aligned}$$

Is ba in L(G)? Running time?

New grammar G' is:

$$\begin{aligned} S &\rightarrow Aa \mid a \\ A &\rightarrow AA \mid ABa \mid Aa \mid Ba \mid a \\ B &\rightarrow BBa \mid Ba \mid a \mid b \end{aligned}$$

Is ba in L(G)? Running time?

Top-down Parser:

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

$$\mathbf{S \rightarrow aS \mid b}$$

- **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:

$\mathbf{G=(V,T,S,P)}$
$\mathbf{w,v \in (V \cup T)^*}$
$\mathbf{a \in T}$
$\mathbf{X,A,B \in V}$
$\mathbf{X}_I \in \mathbf{(V \cup T)^+}$

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

If w $\overset{*}{\Rightarrow}$ av then
a is in FIRST(w)
If w $\overset{*}{\Rightarrow}$ λ then
λ is in FIRST(w)

To compute FIRST:

1. **FIRST(a) = {a}**
2. **FIRST(X)**
   - (a) **If X $\rightarrow$ aw then a is in FIRST(X)**
   - (b) **IF X $\rightarrow$ λ then λ is in FIRST(X)**
   - (c) **If X $\rightarrow$ Aw and $\lambda \in$ FIRST(A) then Everything in FIRST(w) is in FIRST(X)**

3. In general, FIRST($X_1X_2X_3..X_K$) =

- **FIRST(X_1)**
- **$\cup$ FIRST(X_2) if λ is in FIRST(X_1)**
- **$\cup$ FIRST(X_3) if λ is in FIRST(X_1) and λ is in FIRST(X_2)**

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

Example:

$$\begin{aligned} \mathbf{S} &\rightarrow \mathbf{aSc} \mid \mathbf{B} \\ \mathbf{B} &\rightarrow \mathbf{b} \mid \lambda \end{aligned}$$

FIRST(B) =

FIRST(S) =

FIRST(Sc) =

Example

$$\mathbf{S} \rightarrow \mathbf{BCD} \mid \mathbf{aD}$$
$$\mathbf{A} \rightarrow \mathbf{CEB} \mid \mathbf{aA}$$
$$\mathbf{B} \rightarrow \mathbf{b} \mid \lambda$$
$$\mathbf{C} \rightarrow \mathbf{dB} \mid \lambda$$
$$\mathbf{D} \rightarrow \mathbf{cA} \mid \lambda$$
$$\mathbf{E} \rightarrow \mathbf{e} \mid \mathbf{fE}$$

FIRST(S) =

FIRST(A) =

FIRST(B) =

FIRST(C) =

FIRST(D) =

FIRST(E) =

Definition: FOLLOW(X) = set of terminals that can appear to the right of X in some derivation.

$$\textbf{If } S \stackrel{*}{\Rightarrow} wAav \textbf{ then } a \textbf{ is in FOLLOW}(A)$$

To compute FOLLOW:

1. **\$ is in FOLLOW(S)**
2. **If $A \rightarrow wBv$ and $v \neq \lambda$ then FIRST(v) - $\{\lambda\}$ is in FOLLOW(B)**
3. **IF $A \rightarrow wB$ OR**
   $A \rightarrow wBv$ and λ is in FIRST(v)
   then
   FOLLOW(A) is in FOLLOW(B)
4. **λ is never in FOLLOW**

Example:

$$\mathbf{S} \rightarrow \mathbf{aSc} \mid \mathbf{B}$$
$$\mathbf{B} \rightarrow \mathbf{b} \mid \lambda$$

FOLLOW(S) =

FOLLOW(B) =

Example:

$$
\begin{aligned}
\mathbf{S} &\rightarrow \mathbf{BCD} \mid \mathbf{aD} \\
\mathbf{A} &\rightarrow \mathbf{CEB} \mid \mathbf{aA} \\
\mathbf{B} &\rightarrow \mathbf{b} \mid \lambda \\
\mathbf{C} &\rightarrow \mathbf{dB} \mid \lambda \\
\mathbf{D} &\rightarrow \mathbf{cA} \mid \lambda \\
\mathbf{E} &\rightarrow \mathbf{e} \mid \mathbf{fE}
\end{aligned}
$$

FOLLOW(S) =

FOLLOW(A) =

FOLLOW(B) =

FOLLOW(C) =

FOLLOW(D) =

FOLLOW(E) =