CPS 140 - Mathematical Foundations of CS
Dr. S. Rodger
Section: Parsing (handout)

Parsing

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

Review

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?

Remove λ-rules, then unit productions, and then useless productions from the grammar G above. 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

We will use the following functions FIRST and FOLLOW to aid in computing parse tables.

The function FIRST:

Some notation that we will use in defining FIRST and FOLLOW.

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

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

If w $\stackrel{*}{\Rightarrow}$ av then
  a is in FIRST(w)
If w $\stackrel{*}{\Rightarrow}$ 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($X_1 X_2 X_3 .. X_K$) =
   - 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: $L = \{a^n b^m c^n : n \geq 0, 0 \leq m \leq 1\}$

$$\begin{array}{l} S \rightarrow aSc \mid B \\ B \rightarrow b \mid \lambda \end{array}$$

FIRST(B) =

FIRST(S) =

FIRST(Sc) =

Example

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

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.

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

(where w and v are strings of terminals and variables, a is a terminal, and A is a variable)

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:

S $\rightarrow$ aSc | B
B $\rightarrow$ b | λ

FOLLOW(S) =

FOLLOW(B) =

Example:

S $\rightarrow$ BCD | aD
A $\rightarrow$ CEB | aA
B $\rightarrow$ b | λ
C $\rightarrow$ dB | λ
D $\rightarrow$ cA | λ
E $\rightarrow$ e | fE

FOLLOW(S) =

FOLLOW(A) =

FOLLOW(B) =

FOLLOW(C) =

FOLLOW(D) =

FOLLOW(E) =