Compiler Construction
WA02: Parse Trees

Terence Henriod

September 10, 2015

Abstract

This assignment asks you to prepare written answers to questions on context-free grammars. Each question has a short answer. You may discuss this assignment with other students and work the problems together. However, your writeup should be your own individual work. Remember written assignments are to be turned in in class on the date due.

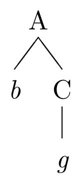
1. Given the Grammar G =

   - $A \to Ba \mid bC$
   - $B \to d \mid eBf$
   - $C \to gC \mid g$

   Determine which strings are in L(G). Construct parse trees for those that are.

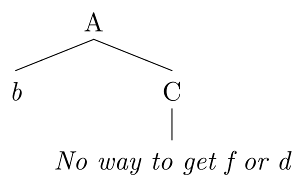
   - bg

     Answer:

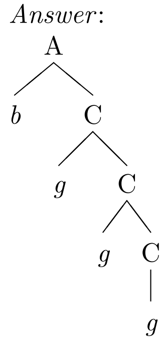
   - bffd

     Answer:
     Not in L(G).

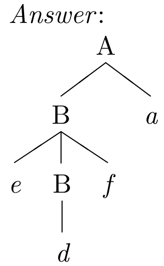
   - bggg

     Answer:

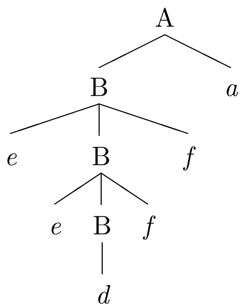
   - edfa

     Answer:

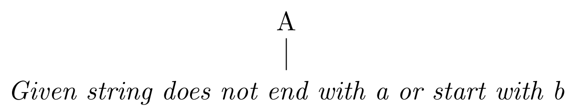
- eedffa

  Answer:



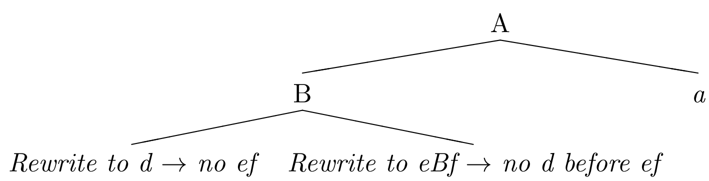
- faae

  Answer:
  Not in L(G).

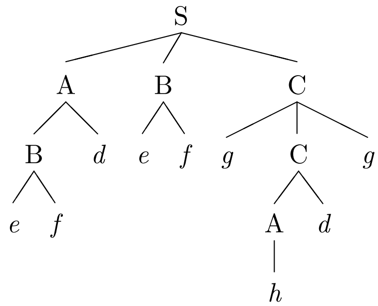


- defa

  Answer:
  Not in L(G).



2. Given the following parse tree,



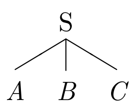
(a) Construct the corresponding rightmost derivation.

Answer:
(Both left and right derivations will be shown simultaneously, along with the sentential form of the string they produce)

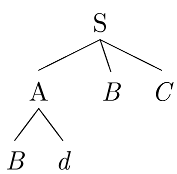
<u>Left</u>

S

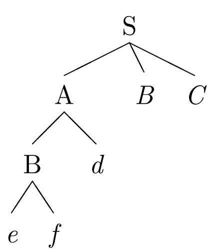
ABC



BdBC



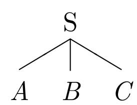
efdBC



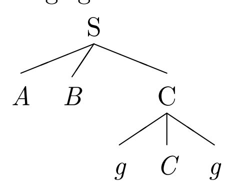
<u>Right</u>

S

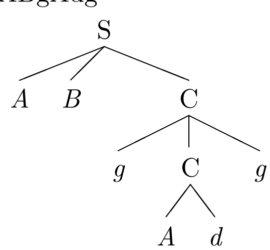
ABC



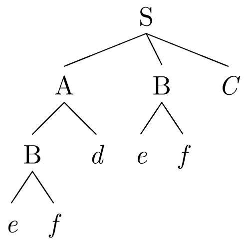
ABgCg



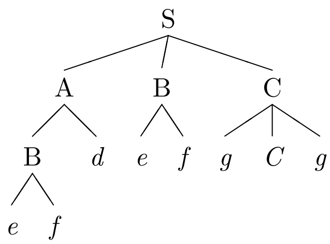
ABgAdg



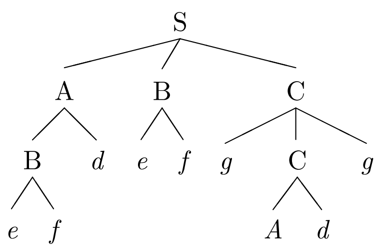
efdefC



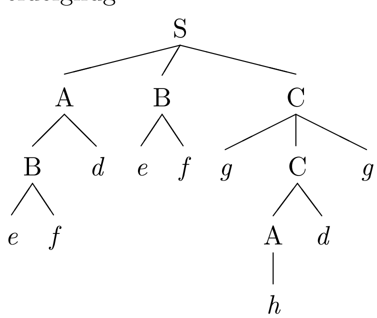
efdefgCg



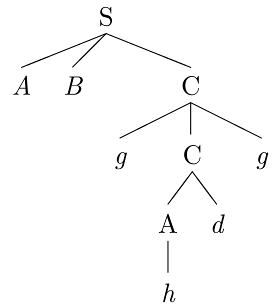
efdefgAdg



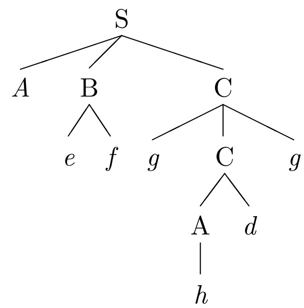
efdefghdg



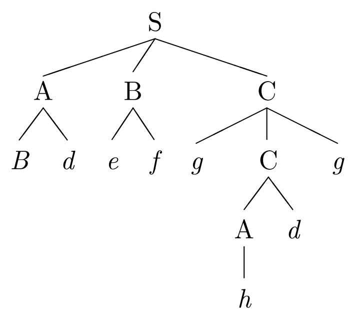
ABghdg



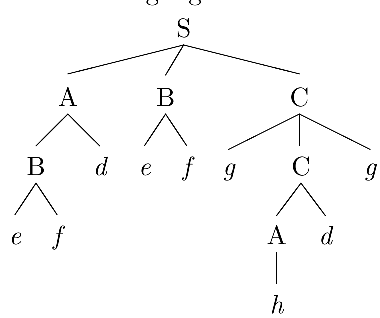
Aefghdg



Bdefghdg



efdefghdg



(b) Construct as much of the grammar as can be determined from the parse tree.

Answer:
G =

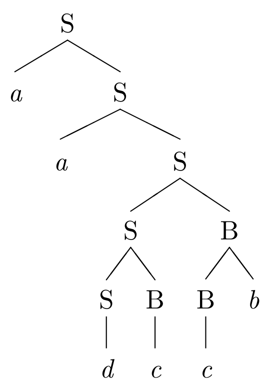
- $S \rightarrow ABC$
- $A \rightarrow Bd \mid h$
- $B \rightarrow ef$
- $C \rightarrow gCg \mid Ad$

3. Given the grammar

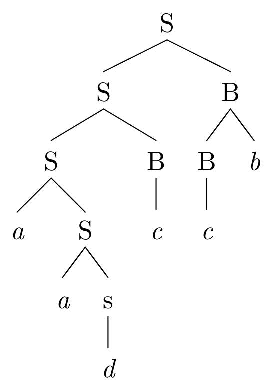
- $S \rightarrow aS \mid SB \mid d$
- $B \rightarrow Bb \mid c$

Show that this grammar is ambiguous by showing two parse trees for the string aadccb.

Answer:
One:



Another one:



4. Our usual expression grammar (Old Faithful)

   - $E \to E + T \mid E - T \mid T$
   - $T \to T * F \mid T/F \mid F$
   - $F \to (E) \mid i$

   can't be used as is in a predictive parser because it has left recursion. Someone had the idea that he could remove left recursion from this grammar by changing it to

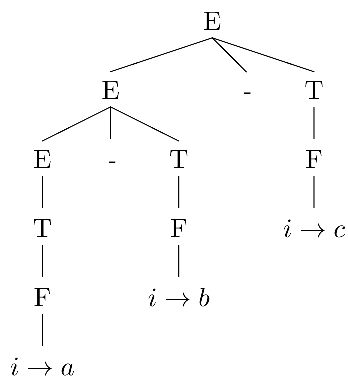
   - $E \to T + E \mid T - E \mid T$
   - $T \to F * T \mid F/T \mid F$
   - $F \to (E) \mid i$

   Show that this isn't such a good idea by:

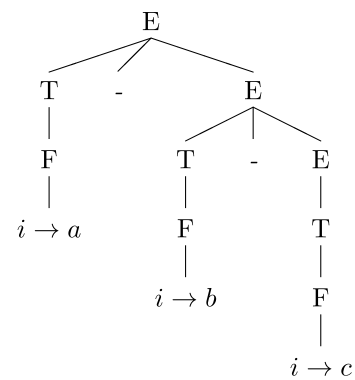
   (a) Drawing parse trees for $a - b - c$ using both the original grammar and the revised grammar.

   Answer:

   Original:



   Modified:



   (b) Using the trees as a guide to determine the results of the corresponding computations if $a = 3$, $b = 5$, $c = 8$.

   Answer:

   Original: $a - b - c \to (a - b) - c = (3 - 5) - 8 = -10$
   Modified: $a - b - c \to a - (b - c) = 3 - (5 - 8) = 6$

   Comment: Apparently the change to the grammar was not a good idea because it changed the grammar to be a consequentially different one than the original.

5. Find the *FIRST* and *FOLLOW* sets for the grammar

   - $S \rightarrow ABC$
   - $A \rightarrow a \mid Cb \mid \epsilon$
   - $B \rightarrow c \mid dA \mid \epsilon$
   - $C \rightarrow e \mid f$

   Answer:

   First Sets:
   Recall the rules for constructing the first set:

   - If X is a terminal, then $First(X) = \{x\}$
   - If a production $X \rightarrow \epsilon$ exists, then $\epsilon \in First(X)$
   - For non-terminal X and symbols (terminal and non-terminal) $Y_1...Y_k$, if a production of $X \rightarrow Y_1...Y_k$ exists, then $First(Y_1...Y_k) \in First(X)$
   - $First(Y_1...Y_k)$ is equal to $First(Y_1) \cup ... \cup First(Y_i) - \{\epsilon\}$, where i indicates the first Y s.t. $\epsilon \notin First(Y_i)$. If there is no such i, then $First(Y_1...Y_k) = First(Y_1) \cup ... \cup First(Y_k) \cup \{\epsilon\}$

$$
\begin{aligned}
First(S \rightarrow ABC) =& \{\} \cup First(ABC) \\
=& First(A) \cup First(B) \cup First(C) \ - \ \{\epsilon\} \\
=& \{\} \cup \{\} \cup \{\} - \{\epsilon\} \\
=& \{a, c, d, e, f\}
\end{aligned}
$$

$$
First(S) = \{a, c, d, e, f\}
$$

$$
\begin{aligned}
First(A \rightarrow a) =& First(a) \\
=& \{a\}
\end{aligned}
$$

$$
\begin{aligned}
First(A \rightarrow Cb) =& First(C) \\
=& \{e, f\}
\end{aligned}
$$

$$
First(A \rightarrow \epsilon) = \{\epsilon\}
$$

$$
\begin{aligned}
First(A) =& \{a\} \cup \{e, f\} \cup \{\epsilon\} \\
=& \{a, e, f, \epsilon\}
\end{aligned}
$$

$$
\begin{aligned}
First(B \rightarrow c) =& First(c) \\
=& \{c\}
\end{aligned}
$$

$$
\begin{aligned}
First(B \rightarrow dA) =& First(d) \\
=& \{d\}
\end{aligned}
$$

$$
First(B \rightarrow \epsilon) = \{\epsilon\}
$$

$$\begin{aligned}First(B) =& First(c) \cup First(dA) \cup \{\epsilon\}\\ =& \{c\} \cup First(d) \cup \{\epsilon\}\\ =& \{c\} \cup \{d\} \cup \{\epsilon\}\\ =& \{c, d, \epsilon\}\end{aligned}$$

$$\begin{aligned}First(C \to e) =& First(e)\\ =& \{e\}\end{aligned}$$

$$\begin{aligned}First(C \to f) =& First(f)\\ =& \{f\}\end{aligned}$$

$$\begin{aligned}First(C) =& First(e) \cup First(f)\\ =& \{e\} \cup \{f\}\\ =& \{e, f\}\end{aligned}$$

$$\begin{aligned}First(a) =& \{a\}\\ First(b) =& \{b\}\\ First(c) =& \{c\}\\ First(d) =& \{d\}\\ First(e) =& \{e\}\\ First(f) =& \{f\}\end{aligned}$$

Follow Sets:
Recall the rules for constructing the follow set:

- The $FollowSet$ for terminals is undefined.
- $\$ \in Follow(S)$, where \$ is the input termination symbol and S is the start symbol of the grammar
- If X appears in the RHS of a production of the form $Y \to aXb$, where a is an arbitrary string (including ϵ) and b is an arbitrary symbol (terminal or not), then $First(b) - \{\epsilon\} \in Follow(X)$
- If X appears in the RHS of a production of the form $Y \to aXb$ and $\epsilon \in First(b)$, then $Follow(Y) \in Follow(X)$
- If X appears in the RHS of a production of the form $Y \to aX$, then $Follow(Y) \in Follow(X)$

$$Follow(S : S\ is\ start) = \{\$\}$$

$$Follow(S) = \{\$\}$$

$$\begin{aligned}Follow(A : S \to ABC) =& First(BC) - \{\epsilon\}\\ =& First(B) \cup First(C) - \{\epsilon\}\\ =& \{c, d, \epsilon\} \cup \{e, f\} - \{\epsilon\}\\ =& \{c, d, e, f\}\end{aligned}$$

$$
\begin{aligned}
Follow(A : B \rightarrow dA) =& Follow(B) \\
=& \{e, f\}
\end{aligned}
$$

$$
\begin{aligned}
Follow(A) =& \{c, d, e, f\} \cup \{e, f\} \\
=& \{c, d, e, f\}
\end{aligned}
$$

$$
\begin{aligned}
Follow(B : S \rightarrow ABC) =& First(C) - \{\epsilon\} \\
=& \{e, f\} - \{\epsilon\} \\
=& \{e, f\}
\end{aligned}
$$

$$
Follow(B) = \{e, f\}
$$

$$
\begin{aligned}
Follow(C : S \rightarrow ABC) =& Follow(S) \\
=& \{\$\}
\end{aligned}
$$

$$
\begin{aligned}
Follow(C : A \rightarrow Cb) =& First(b) \\
=& \{b\}
\end{aligned}
$$

$$
\begin{aligned}
Follow(C) =& \{\$\} \cup \{b\} \\
=& \{\$, b\}
\end{aligned}
$$