Homework: Context-Free Grammar

Questions:

1. (10 pt) Consider the context free grammar G: $S \to SbS | *S | a+ | c$ and string $a+b*c$

   (a) (2 pt) What are the terminals and non-terminals of the grammar?

   (b) (2 pt) Give a leftmost derivation for the string

   (c) (2 pt) Give a rightmost derivation for the string

   (d) (2 pt) Give a parse tree for the string

   (e) (2 pt) Write 3 strings using the terminals that do not belong to the language of the grammar $L(G)$

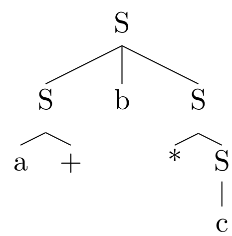
   Sol:

   (a) (2pt) terminals: $a, c, +, *$, non-terminals: S

   (b) (2pt) left-most derivation: $S \to SbS \to a+bS \to a+b*S \to a+b*c$

   (c) (2pt) right-most derivation: $S \to SbS \to Sb*S \to Sb*c \to a+b*c$

   (d) (2pt)



   (e) (2pt) aa, $+ac$, $a+c$

2. (10 pt) Consider the following grammar with:

   - terminals: $a, b, c, !, \&, <$
   - non-terminals: S, E, F
   - start symbol: S
   - production rules:
     $S \to E | c$
     $E \to E \; ! \; F | F$
     $F \to F \; \& \; F | F < F | a | b$

   (a) (2 pt) Draw two different parse trees for the string $b \; ! \; a < b \; \& \; a$.

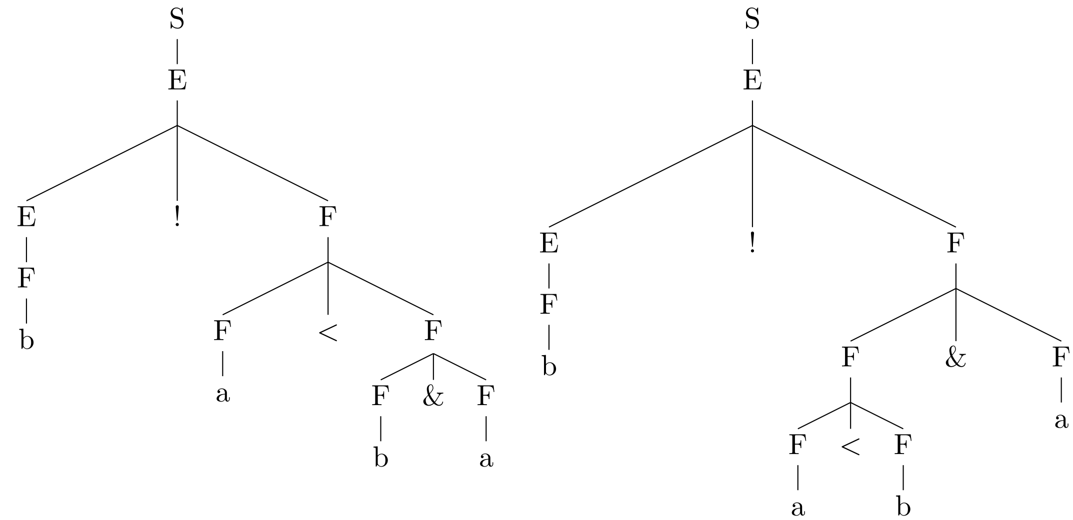
   (b) (4 pt) Modify the grammar to remove ambiguity.

   (c) (1 pt) Draw the parse tree for the string using new grammar

   (d) (3 pt) Explain how your new grammar modifies the parse trees you drew in the first step to remove ambiguity

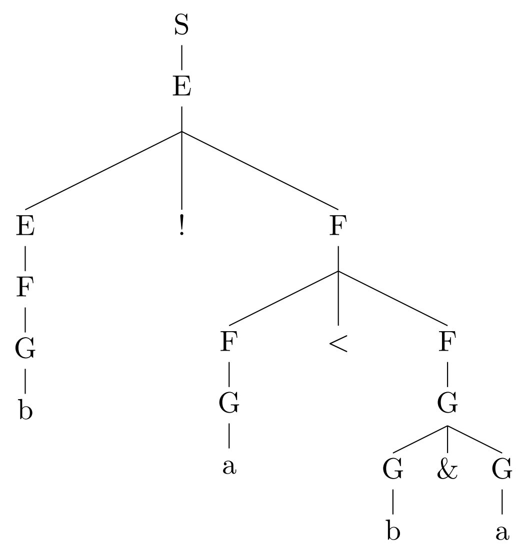
Sol

(a) (2 pt)



(b) (4 pt) production rules:
$S \rightarrow E|c$
$E \rightarrow E \; ! \; F|F$
$F \rightarrow F < F|G$
$G \rightarrow G \; \& \; G|a|b$

(c) (1 pt)



(d) (3 pt) The new grammar uses operator precedence to disambiguate the grammar. In the new grammar the operator & has precedence over the operator < as it is pushed towards the bottom of the parse tree

3. (10 pt) Consider the following grammar where S is the start variable:

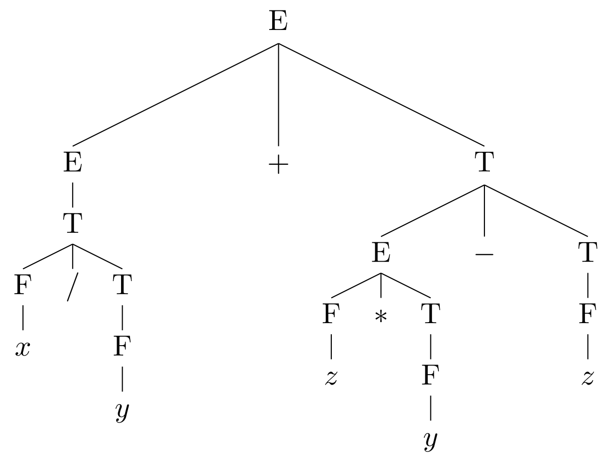
- terminals: $x, y, z, +, , *, /$
- non-terminals: E, T, F, V

- start symbol: E
- production rules:
  $E \rightarrow E + T | E - T | T$
  $T \rightarrow F * T | F/T | F$
  $F \rightarrow x | y | z$

(a) (4 pt) What is the associativity of the operators $+$, $-$, $*$ and $/$; explain why.

(b) (3 pt) What is the precedence of $+$,$-$, $*$ and $/$; explain why.

(c) (3 pt) Given a parse tree



Explain how the value of the string is generated.

Sol

(a) (4 pt) The operators $+$ and $-$ is left-associative because the head E appears at the left side of the operator, and the operators $*$ and $/$ is right-associative because the head T appears at the right side of the operator.

(b) (3 pt) The operators $*$ and $/$ have the same precedence and a higher precedence over $+$ and $-$ because they are further down in the grammar rules, which means they are evaluated before.

(c) (3 pt) We first generate the values from the leaf nodes x, y, z, y, z. We then traverse the parse tree and apply the operators first to the leaf nodes and computes x/y and $z * y$. We continue to apply operators recursively in a bottom up fashion until we reach the root node. In this process, we generate $z * y - z$ and then finally $x/y + z * y - z$.

4. (10 pt) Design CFGs for the given languages:

(a) (4 pt) Write a grammar that describes the strings $0^n 1^m$ where $m > n$.

(b) (6 pt) Write a grammar that describes a "baby language" and list two most interesting sentences he/she will say:

- the baby has very limited vocabularies "apple" "puppy" "owl" "eat" "mama" "baba" "big" "red"
- the baby knows a few basic rules to put together a sentence: (1) a sentence can be a noun, a verb or an adjective (2) a sentence can consist of a verb and one or more noun(s) (adjectives are optional), where a noun should be before the verb and/or after the verb, and an adjective only can be immediately before the noun.

Sol

(a) Two possible solutions:
$S \rightarrow A1$
$A \rightarrow BC|\epsilon$
$B \rightarrow 0A|\epsilon$
$C \rightarrow A1$
or
$S \rightarrow AB$
$A \rightarrow 0A1|\epsilon$
$B \rightarrow 1C$
$C \rightarrow B|\epsilon$

(b) $S \rightarrow TVT|N|A$
$T \rightarrow ANT|\epsilon$
$A \rightarrow big|red|\epsilon$
$N \rightarrow apple|puppy|owl|mama|baba$
$V \rightarrow eat$

The English description above still can be ambiguous. If you have different answers from above but you have written down your assumptions to clarify your understandings of the rules, we will still provide you credits accordingly.

Example strings that belong to the baby language:
eat
red
baba eat
mama eat apple
big owl eat apple
eat big apple red apple