Homework: Context-Free Grammar

Learning Objectives:

In this homework, we are going to exercise the following key knowledge points on the topic of context-free grammar (CFG)

1. understanding grammars and CFGs
2. relations of strings and grammar
3. performing derivations and constructing parse trees
4. determining and resolving ambiguity
5. designing a grammar to describe given string patterns

Instructions:

1. Total points: 40 pt
2. Early deadline: Sept 11 (Wed) 11:59 pm, Regular deadline Sept 13 (Fri) 11:59 pm
3. How to submit:
   - Submit your document to Canvas under Assignments, Homework 1
   - Please provide the complete solutions in one pdf file
   - You can write your solutions in latex or word and then convert it to pdf; or you can submit a scanned document with legible handwritten solutions

Questions:

1. (10 pt) Given a string $a0b10c$ and the context free grammar G:
   $S \rightarrow SA|A|SD$
   $A \rightarrow a|b|c$
   $D \rightarrow 0|1$

   (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 $L(G)$

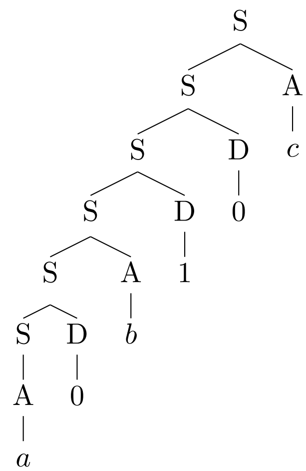
   Sol:

   (a) terminals : $a, b, c, 0, 1$, non-terminals:S, A, D

(b) $S \Rightarrow SA \Rightarrow SDA \Rightarrow SDDA \Rightarrow SADDA \Rightarrow SDADDA \Rightarrow ADADDA \Rightarrow aDADDA \Rightarrow a0ADDA \Rightarrow a0bDDA \Rightarrow a0b1DA \Rightarrow a0b10A \Rightarrow a0b10c$

(c) $S \Rightarrow SA \Rightarrow Sc \Rightarrow SDc \Rightarrow S0c \Rightarrow SD0c \Rightarrow S10c \Rightarrow SA10c \Rightarrow Sb10c \Rightarrow SDb10c \Rightarrow S0b10c \Rightarrow A0b10c \Rightarrow a0b10c$

(d)



(e) 11, 1a, 010bac

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

- terminals: $x, y, z, >, <, 0, 1, (,)$, if, then, else
- non-terminals: S, F, B, T, E, N
- start symbol: S
- production rules:
  $S \rightarrow F | T\ N\ T$
  $F \rightarrow$ if B then $S|$ if B then S else S
  $B \rightarrow (TET)$
  $T \rightarrow x|y|z|1|0$
  $E \rightarrow > \mid <$
  $N \rightarrow + | - | =$

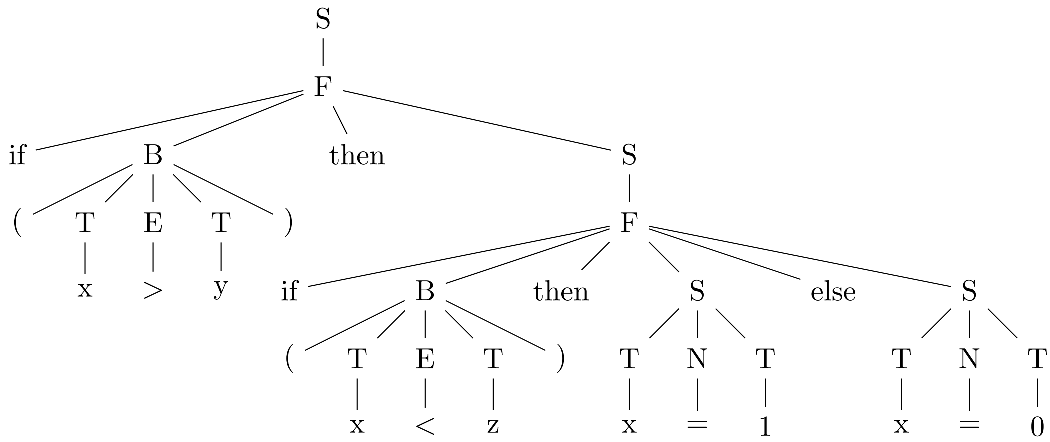
(a) (4 pt) Draw two different parse trees for the string
if $(x > y)$ then
if $(x < z)$ then $x = 1$
else $x = 0$.

(b) (2 pt) Modify the grammar to remove ambiguity.

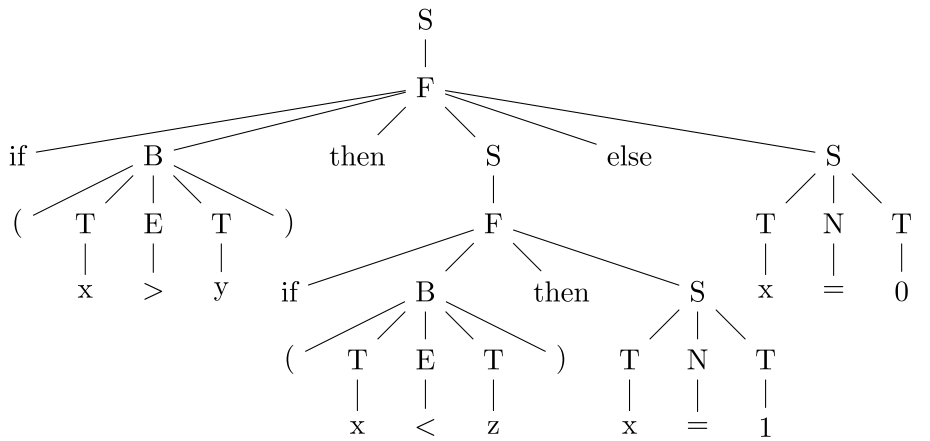
(c) (2 pt) Draw the parse tree for the string using new grammar

(d) (2 pt) Explain how your new grammar modifies the parse trees you drew in the first step to remove ambiguity

Sol:



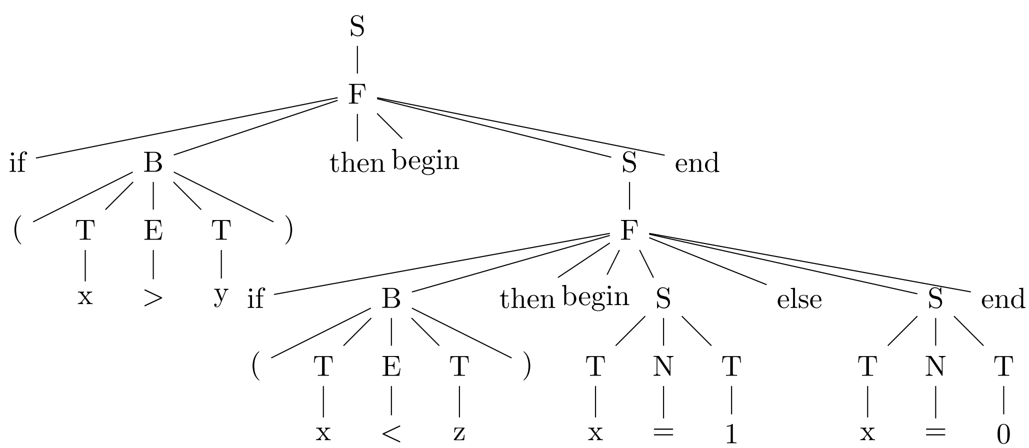
(a)



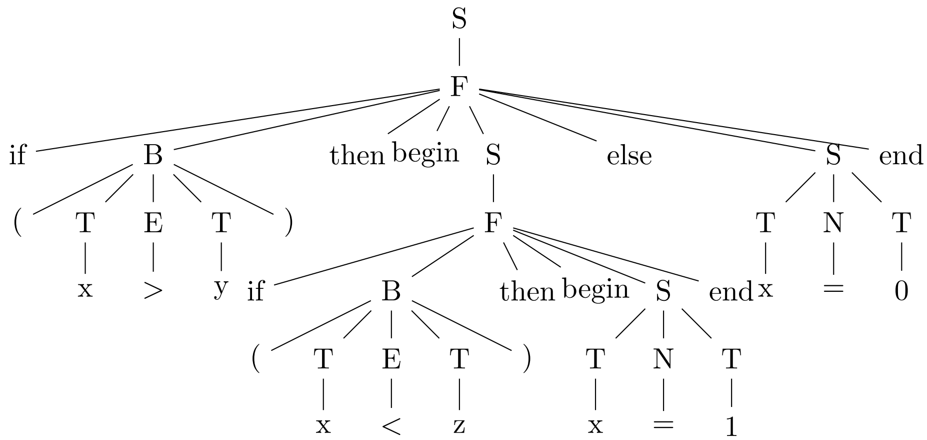
(b) We should change the production rule

$F \rightarrow$ if B then $S|$ if B then S else S to

$F \rightarrow$ if B then begin S end $|$ if B then begin S else S end



(c)



(d) By adding delimiters, we removed ambiguity. It is easy to find the two parser tree will generate two different strings. The first one will generate ”if (x>y) then begin if (x<z) then begin x = 1 else x = 0 end end”, and the second one will generate ”if (x>y) then begin if (x<z) then begin x = 1 end else x = 0 end”. In this way, we know which ”if” the ”else” follows.

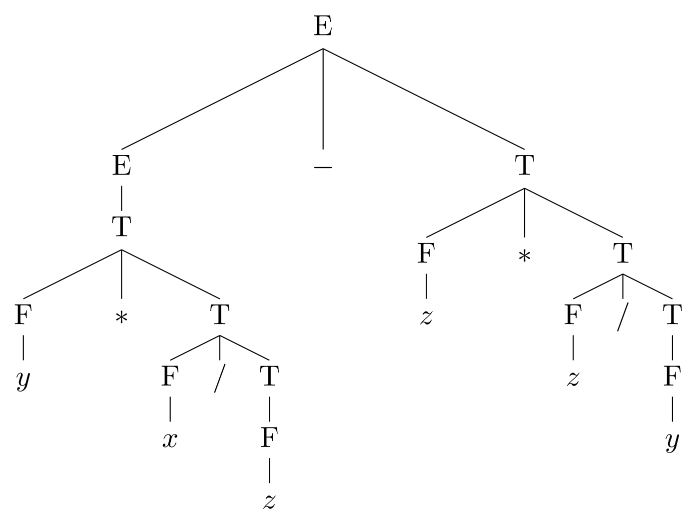
3. (10 pt) Consider the following grammar:

- 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) The operators + and − are left-associative because the head E appears at the left side of the operator, and the operators ∗ and / are right-associative because the head T appears at the right side of the operator.

(b) The operators ∗ and / have the same precedence and a higher precedence over + and − because they are further down in the grammar rules, which mean they are evaluated before.

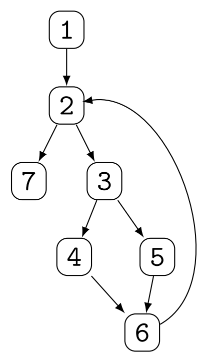
(c) 1)x / z
2)y * x/z
3)z / y
4)z/y + z
5)y*x/z - z/y+z

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

(a) (2 pt) Write a grammar that describes the strings $0^*1^+2^*$.

(b) (3 pt) Write a grammar that describes the strings 0^n1^m, where $n > m$.

(c) (5 pt) Given a graph below, where 1 is an entry and 7 is an exit, we can generate paths like 127, 1234627, 1235627, 12356234627 ... Write a grammar that describes these paths.



Sol:

(a) $S \rightarrow B|AB|BC|ABC$
$A \rightarrow 0|0A$
$B \rightarrow 1|1B$
$C \rightarrow 2|2C$

(b) $S \rightarrow 0|0S|0S1$

(c) $S \rightarrow 1A$
$A \rightarrow 2F|2B$
$B \rightarrow 3C|3D$
$C \rightarrow 4E$
$D \rightarrow 5E$
$E \rightarrow 6A$
$F \rightarrow 7$