

# Homework: Context-Free Grammar

Due-date: Jan 31 at 11:59am Submit online on Canvas (**Format: pdf**; No other format allowed)

*Homework must be individual's original work. Collaborations and discussions of any form with any students or other faculty members are not allowed. If you have any questions and/or concerns, post them on Piazza and/or ask 342 instructor and TAs.*

## Learning Outcomes

- Knowledge and application of Grammars
- Knowledge of Mathematical Formalisms in Programming Languages

## Questions

1. True/False. Justify your answer.

- (a) Two or more grammars may generate the same language.
- (b) Ambiguity in the meaning of a grammatically valid sentence implies that the grammar, from which the sentence is generated, is ambiguous.
- (c) A language is ambiguous if there is some ambiguous grammar that generates it.
- (d) Any palindrome over the letters  $a$  and  $b$  can be generated by the following grammar:  
 $S \rightarrow aSa \mid bSb \mid \epsilon$ .  
( $\epsilon$  denotes empty string)
- (e) The result of the Java expression

`++y + ++x * 2`

is 53 when  $y$  is 10 and  $x$  is 20.

(10pts)

2. The language of a grammar  $G$  is denoted by  $L(G)$ .  $L(G)$  is the set of all strings that are generated by  $G$ . Given two grammars  $G_1$  and  $G_2$ , we say that  $G_1$  is more expressive than  $G_2$  (denoted by  $G_1 > G_2$ ) when  $L(G_2) \subset L(G_1)$ . That is, the strings generated by  $G_2$  can be generated by  $G_1$  but not the other way around.

Consider the following grammars and order them as per  $>$  relation described above. Justify your answer.

G1:  $S \rightarrow a \mid b \mid aSa \mid bSb \mid \epsilon$

G2:  $S \rightarrow a \mid b \mid Sa \mid Sb \mid \epsilon$

G3:  $S \rightarrow a \mid b \mid aa \mid bb \mid aSa \mid bSb \mid \epsilon$

( $\epsilon$  denotes empty string)

(6pts)

3. Consider the following grammar for expressions.

$E \rightarrow E + E \mid E / T \mid T$

$T \rightarrow T - R \mid T * R \mid R$

$R \rightarrow \text{Number}$

- (a) Is the above grammar ambiguous? Justify your answer.
- (b) What are the associativities and precedence-order of the operators:  $+$ ,  $-$ ,  $*$ ,  $/$ ? Justify your answer.

- (c) Draw a derivation sequence and corresponding parse trees for the following expressions (assume `Number` denotes any integer):

- i.  $1 + 2 - 3 - 4$ .
- ii.  $1 + 2 * 3 / 4 / 5 - 6$ .

(16pts)

4. We plan to define a new programming language. A statement in this language can be of the following form:

- an assignment statement `var := E`, where `E` is an arith-expression involving variables, integers and some arithmetic operators;
- a conditional statement `if BE then statement` where `BE` is a boolean expression representing relationship between two arith-expressions.
- unconditional statement `goto id` where `id` is a label of some statement
- a statement may have a label — `id statement`, where `id` denotes the label of statement
- a compound statement which starts with a `begin` and ends with an `end` and between these delimiters contains at least two statements separated by `;` (i.e., the last statement is not proceeded by `;`).

You can assume typical infix grammar for arithmetic expression and boolean expressions: for instance  $x+1$  is an arithmetic expression  $x < 20$  is a boolean expression.

- (a) Does the following program satisfies the above rules. Justify your answer.

```
L0  begin
L1  x := x + 1;
L2  if x < 20 then goto L1;
L3  y := x
    end
```

- (b) Does the following grammar (start symbol is `P`) correctly represents the above rules. That is, if a program follows the rules then it is also generated by the grammar; and if a program is generated by the grammar then it also follows the rules. Justify your answer.

$$\begin{aligned}
 P &\longrightarrow L \text{ Stmt} \\
 L &\longrightarrow \text{id} \\
 St &\longrightarrow Var := E \mid \text{if } BE \text{ then } St \mid \text{goto } L \mid \text{begin } StSeq \text{ end} \\
 StSeq &\longrightarrow St \mid St; StSeq
 \end{aligned}$$

Assume that the grammar for variable names is denoted by `Var`, the grammar for type of labels is denoted by `id`, grammar for all possible arith-expressions and all possible boolean expressions as per the rules are given by `E` and `BE`, respectively.

(8pts)