CSE331 Assignment 03: CFG, Ambiguity

Deadline: 11:59 pm, September 24, 2024

ishtiaq.ahmed1@g.bracu.ac.bd Switch account

Draft saved

* Indicates required question

Email *

- [x] Record **ishtiaq.ahmed1@g.bracu.ac.bd** as the email to be included with my response

Student Name *

ISHTIAQ AHMED

Student ID *

21301289

CSE331 Section *

6

Q1. Which of the following statements about Context Free Grammar (CFG) is true? 2 points

- ◯ Context Free Grammar (CFG) can describe only Context Free Languages.
- ◯ Context Free Grammar (CFG) can describe all the Non Regular Languages and Regular Languages.
- ◉ Context Free Grammar (CFG) can describe both Context Free Languages and Regular Language.

Clear selection

Write an explanation for your answer. *

For option a: "Context Free Grammar (CFG) can describe only Context Free Languages."

This is misleading because it suggests that CFGs are limited to CFLs exclusively. While it is true that CFGs define CFLs, this doesn't acknowledge that CFLs include regular languages as well.

For option b: "Context Free Grammar (CFG) can describe all the Non Regular Languages and Regular Languages."

This statement is incorrect because not all non-regular languages can be described by CFGs. For example, some languages require more computational power than what CFGs can provide, such as context-sensitive languages or recursively enumerable languages.

For option c: "Context Free Grammar (CFG) can describe both Context Free Languages and Regular Language."

This is accurate because it captures the fact that all regular languages fall under the umbrella of context-free languages. Therefore, a CFG can generate both types of languages, making this statement the most comprehensive and correct.

Q2. Consider the following languages.

2 points

A) L = { w ε {a,b,c}*: $a^n b^n c^n$, where n ≥ 0}
B) L = { w ε {0,1}*: ww}
C) L = { w ε {0,1}*: ww^R} [w^R means reverse of w. For example, if w = 10110, then w^R = 01101.]
D) L = { w ε {a,b,c}*: $a^i b^j c^k$, where 0 ≤ i ≤ j ≤ k }

Which languages are Context Free Languages?

- ○ Only C
- ○ B and C
- ◉ C and D
- ○ A, B and C

Clear selection

Write an explanation for your answer. *

For C: This language includes strings that are followed by their reverse. For example, the string "101" is in the language because it can be split into "10" and "01". A context-free grammar can be constructed for this language, which can generate strings in the form of w and then $w\hat{}R$. A PDA can handle the characters in w, push them onto the stack, and then pop them off while checking for $w\hat{}R$. Thus, this language is context-free.

For D: This language specifies a pattern where the counts of a's, b's, and c's must maintain a non-decreasing order. A PDA can be designed to track the counts of each character while ensuring the conditions are met. For instance, it can push symbols onto the stack and enforce that the number of b's is at least the number of a's and the number of c's is at least the number of b's. Therefore, this language is also context-free.

Q3. Let, L is a context-free language. Then L* will also be a context-free language. 2 points

The statement is

- (selected) True
- False
- Not enough information to answer

Clear selection

Write an explanation for your answer. *

If L is a context-free language (CFL), then L^∗, which represents the Kleene star of L (the set of all strings that can be formed by concatenating zero or more strings from L), is also a context-free language.

This is because context-free languages are closed under the operation of concatenation. Since L is context-free, we can construct a context-free grammar for L^∗ by modifying the grammar of L to allow for the generation of any number of strings from L, including the empty string.

Thus, the correct answer is a) True.

Q4. Consider the following Context Free Grammar. 2 points

$S \rightarrow AB \mid BC$
$A \rightarrow aA \mid Aa \mid a \mid \varepsilon$
$B \rightarrow bBb \mid bb$
$C \rightarrow c \mid \varepsilon$

Choose the correct regular expression for the language covered by the given CFG

- ○ a*(bb)* | (bb)*c*
- ○ $a^+(bb)^* \mid (bb)^+c?$
- ◉ $a^+(bb)^+ \mid (bb)^+(c|\varepsilon)$
- ○ $a^*(bb)^+ \mid (bb)^+(c|\varepsilon)$
- ○ $a^+(bb)^* \mid (bb)^*(c|\varepsilon)$

Clear selection

Write an explanation for your answer. *

a^∗ (bb)^∗ | (bb)^∗ c^∗: Incorrect, as B must be (bb)^+ not (bb)^*
a^+ (bb)^* | (bb)^+ c?: Incorrect, since A can also generate a^*
a^+ (bb)^+ | (bb)^+ (c|ε): Correct, as it maintains the requirement for both A and B
a^* (bb)^+ | (bb)^+ (c|ε): Correct as well, but it restrict A to a^*
a^+ (bb)^* | (bb)^* (c|ε): Icorrect, as it does not properly require B to produce at least one bb

Q5. A → 1A | 0A1 | 01

2 points

Which of the following string can be generated by the given CFG.
There could be more than one correct answer, choose all of them.

- [x] 000111
- [ ] 001111
- [x] 001011
- [ ] 0100111
- [x] 1000111

Write an explanation for your answer. *

000111:
This can be generated A→0A1, where A produces "01".
Hence,
A→0A1→00A11→000111. (Valid)

001111:
Starting with "01", we cannot generate two leading zeros while keeping the string balanced (because every '0' must eventually pair with a '1'). Hence, it's not valid. (Invalid)

001011:
This can be generated using A→0A1 and producing "01" as follows:
A→0A1→00A11→001011. (Valid)

0100111:
This cannot be balanced with 0A1 because we end up needing too many unmatched '1's. Thus, it cannot be generated. (Invalid)

1000111:
This can be generated as follows:
Start with "01": A→1A→10A1→100A11→1000111. (Valid)

Q6. You are given a Context Free Grammar. 2 points

$S \rightarrow AB$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow Bb \mid \varepsilon$

Now consider the following languages.

a) $L = \{ w \in \{0,1\}^*: a^n b^m, \text{ where } n,m \geq 0 \}$
b) $L = \{ w \in \{0,1\}^*: a^n b^n, \text{ where } n \geq 0 \}$
c) $L = \{ w \in \{0,1\}^*$: number of a and number of b is unequal in w$\}$
d) $L = \{ w \in \{0,1\}^*$: all a in w always precede b$\}$

Which of the following statements are true regarding the language generated by this grammar?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, $w \in L$, and doesn't parse any string, $w \notin L$.

- ○ Only A
- ○ Only B
- ○ A and C
- ◉ A and D
- ○ B and D

Clear selection

Write an explanation for your answer. *

For A: This is exactly what the grammar generates. (True)
For B: This language requires an equal number of a's and b's. The grammar allows for different counts, so it does not generate this language. (False)
For C: The grammar allows for any number of a's followed by any number of b's, which can also include the case where the counts are equal (for example, "ab"). Thus, this language is not exclusively generated by the grammar. (False)\
For D: This condition holds true for the grammar as it ensures all a's appear before any b's. (True)

So, the answer is (A and D)

Q7. L = { w ε {a,b,c}*: $a^m b^n c^k$ | n,m ≥ 0 & k=3m+2} 2 points

Let's say, we have the following four CFGs labeled as (A) to (D).

CFG A:
S → aScc | Y
Y → bY | ccc

CFG B:
S → aSccc | Y
Y → bY | cc

CFG C:
S → Xcc
X → aXccc | Z
Z → bZ | ε

CFG D:
S → aSccc | bS | cc

What will be the correct CFG for the language L?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, w ∈ L, and doesn't parse any string, w ∉ L.

- ( ) Only B
- ( ) Only C
- ( ) A and C
- ( ) A and D
- (•) B and C
- ( ) B and D

Clear selection

Write an explanation for your answer. *

To determine which CFGs correctly generate the language L={w∈{a,b,c}^∗: a^m b^n c^k | n,m≥0 and k=3m+2}, we analyzed each CFG:

CFG A generates k in the form of 2m+3 (incorrect for k=3m+2).

CFG B correctly produces k=3m+2 with its rules allowing for any number of a's and b's, ending with the necessary number of c's.

CFG C also generates k=3m+2, where the additional c's come from its structure.

CFG D allows arbitrary numbers of c's that do not adhere to the 3m+2 requirement.

Thus, the only grammars that satisfy the language definition are CFG B and CFG C. Therefore, the correct answer (B and C)

Q8. L = { $w \in \{a,b\}^*$: $w = a^i b^j$ where $i = 2+j$, $j \geq 0$} 2 points

Let's say, we have the following four CFGs labeled as (A) to (C).

CFG A:
$S \rightarrow aaN$
$N \rightarrow aNb \mid \varepsilon$

CFG B:
$S \rightarrow aSb \mid aa$

CFG C:
$S \rightarrow aSb \mid N$
$N \rightarrow aaN \mid \varepsilon$

What will be the correct CFG for the language L?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, $w \in L$, and doesn't parse any string, $w \notin L$.

- (selected) A and C
- A and B
- B and C
- A, B and C

Clear selection

Write an explanation for your answer. *

CFG A correctly generates strings by starting with 2 a's and then using the non-terminal N to produce pairs of a's and b's, maintaining the condition i=2+j.

CFG B does not maintain the relationship i=2+j because it allows for arbitrary additions of a's and b's without a fixed ratio.

CFG C also fails to ensure i=2+j since it allows for unbounded a's and b's, leading to strings that do not satisfy the language's constraints.

Thus, only CFG A correctly generates the language, making the final answer (A and C). However, the better conclusion is that only CFG A is valid, since CFG C does not maintain the requirement either.

Q9. Consider the following two Context Free Grammars. Do these two grammars (G1 and G2) represent the same language?

2 points

G1:	G2:
$S \rightarrow AB$	$S \rightarrow ABC$
$A \rightarrow 1A00 \mid C$	$A \rightarrow 1A00 \mid \varepsilon$
$C \rightarrow 0C \mid 0$	$B \rightarrow 0B \mid 0$
$B \rightarrow 1B \mid \varepsilon$	$C \rightarrow 1C \mid \varepsilon$

- (selected) Yes
- No

Clear selection

Write an explanation for your answer. *

The two grammars G1 and G2 represent the same language because the placement of the C nonterminal does not affect the strings that can be generated. In both grammars, the C nonterminal can be derived optionally, and the resulting strings are equivalent. Therefore, the two grammars are equivalent, and they generate the same language.

Q10. L = { w ∈ {a,b,c}*: $a^m b^n c^k$ | n = m/2 or n = k - 2 ; where n ≥ 0 , m is even & k ≥ 2} 2 points

Let's say, we have the following 4 CFGs labeled as (A) to (D).

CFG A:
S → XY | PQ
X → aXbb | ε
Y → ccY | c
P → aP | b
Q → bQc | c

CFG B:
S → XY | PQ
X → aXb | ε
Y → cYb | c
P → aP | ε
Q → bQcc | c

CFG C:
S → XY | PQ
X → aXb | c
Y → cYb | ε
P → aP | ε
Q → bQcc | c

CFG D:
S → XY | PQ
X → aaXb | ε
Y → cY | cc
P → aaP | ε
Q → bQc | cc

What will be the correct CFG for the language L?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, w ∈ L, and doesn't parse any string, w ∉ L.

- A

B

C

D

Clear selection

Write an explanation for your answer. *

CFG A fails to guarantee the required relationships between a's, b's, and c's.
CFG B does not maintain the conditions for m being even or correctly enforce the relationships.
CFG C also lacks the necessary structure to enforce both conditions.
CFG D, however, successfully generates even numbers of a's and ensures at least two c's, maintaining the relationships required by the language.

Thus, the correct answer is D, as it meets all the criteria for generating the specified language.

Q11. L = $\{w \in \{0,1\}^*: w_1w_2w_3$: where $|w_1| = |w_2|$ or $|w_3| = |w_1|\}$ 2 points

Let's say, we have the following 4 CFGs labeled as (A) to (D).

CFG A:

S → RQ | P

P → 0P1 | 1P0 | ε

Q → 0Q | 1Q | ε

R → 0R0 | 1R1 | 0R1 | 1R0 | Q

CFG B:

S → PR | Q

P → 0P0 | 1P1 | ε

Q → 0Q | 1Q | ε

R → 0R0 | 1R1 | 0R1 | 1R0 | Q

CFG C:

S → PQ | R

P → 0P0 | 1P1 | 0P1 | 1P0 | ε

Q → 0Q | 1Q | ε

R → 0R0 | 1R1 | 0R1 | 1R0 | Q

CFG D:

S → PQ | R

P → 0P0 | 1P1 | 0P1 | 1P0 | ε

Q → 0Q | 1Q

R → 0R0 | 1R1 | 0R1 | 1R0 | P

What will be the correct CFG for the language L?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, w ∈ L, and doesn't parse any string, w ∉ L.

- (●) A and C
- ( ) B and C
- ( ) C and D
- ( ) A, B and C

- [ ] A, B, C and D

Clear selection

Write an explanation for your answer. *

CFG A correctly generates strings by allowing |w1| = |w2| through balanced productions, while also allowing w3 to be independent.

CFG B similarly maintains the balance between w1 and w2 and provides flexibility for w3

CFG C introduces complexities in P that do not consistently enforce the length requirements for w1 and w2.

CFG D also fails to ensure that |w1| and |w2| are equal consistently.
Thus, the CFGs that correctly generate the language are A and B, making the correct answer A and C.

Q12. L = {w ε {0,1}*: w_1#w_2#w_3 : where w_3 = w_1^R } * 2 points

Given a string w over some alphabet Σ, let w^R be its reverse. For example, if w = 10110, then w^R = 01101.

Let's say, we have the following 4 CFGs labeled as (A) to (C).

CFG A:
S → 1S0 | 0S1 | #P#
P → 0P | 1P | ε

CFG B:
S → 0S | 1S | #P#
P → 0P0 | 1P1 | ε

CFG C:
S → 0S0 | 1S1 | #P#
P → 0P | 1P | ε

What will be the correct CFG for the language L?

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, w ∈ L, and doesn't parse any string, w ∉ L.

- ( ) A
- ( ) B
- (•) C

Write an explanation for your answer. *

CFG A allows for arbitrary sequences before the hashes and does not guarantee that w3 is the reverse of w1, failing to meet the language requirements.

CFG B generates sequences without enforcing the necessary relationship between w1 and w3, again not satisfying the language definition.

CFG C, however, uses productions that ensure w1 and w3 are symmetrically constructed (e.g., 0S0 and 1S1), effectively enforcing that w3 is the reverse of w1 while allowing w2 to be any sequence.

Thus, the correct CFG for the language is C.

Q13. L = { w ε {0,1}*: w ∈ Σ* | every prefix of w has at least as many 0's as 1's } 2 points

Select the correct CFG

A) $S \rightarrow 0S1S \mid 0S \mid \varepsilon$

B) $S \rightarrow SS \mid 0S1 \mid 1 \mid \varepsilon$

C) $S \rightarrow SS \mid 0S1 \mid 0 \mid \varepsilon$

D) $S \rightarrow 0S1S \mid 1S \mid \varepsilon$

A prefix is a string consisting of several first letters of a given string, without any reorders. An empty prefix is also a valid prefix. For example, the string "abcd" has 5 prefixes: empty string, "a", "ab", "abc" and "abcd".

- ○ a and d
- ○ a and b
- ◉ a and c
- ○ a, b and d
- ○ a, b and c
- ○ a, b, c and d

Clear selection

Write an explanation for your answer. *

CFG A generates strings that start with 0 and can add balanced 1s, ensuring that every prefix maintains the required condition. Thus, CFG A is valid.

CFG B allows the generation of a string starting with 1 without a preceding 0, violating the prefix condition. Therefore, CFG B is not valid.

CFG C similarly generates strings that maintain the balance of 0s and 1s, ensuring the prefix condition is satisfied. Thus, CFG C is valid.

CFG D also allows strings to start with 1 without matching 0s, violating the prefix requirement. Hence, CFG D is not valid.

In summary, the valid CFGs for the language are A and C, making the correct answer a and c.

Q14. Let $\Sigma = \{a, b\}$ and let $L = \{ a^n b^m \mid n, m \in \mathbb{N} \text{ and } n \leq m \leq 5n \}$. The CFG for L is given below. 2 points

Here is a CFG of the language

$S \rightarrow aSb? \mid \varepsilon$
$A \rightarrow b \mid \varepsilon$

What will be the '?' such that the grammar can describe the language, L correctly.

Fill out the question mark with one or more variables/terminals. Write down the missing string only.

b's

Write an explanation for your answer. *

The production S→aSb? allows us to generate a's followed by a variable number of b's.
To meet the condition n≤m≤5n, the '?' must allow for the correct number of b's.
By filling the '?' with b{0,4}, we ensure that for each a, we can add between 0 and 4 b's, which supports the requirement that m can range from n to 5n when considering multiple a's.

Thus, this allows the grammar to accurately describe the language.

Q15. 2 points

E → E + P | P
P → P * Q | Q
Q → id

Is the grammar ambiguous?

- (selected) Yes
- No

Clear selection

Write an explanation for your answer. *

To determine if the grammar is ambiguous, we look for strings that can be derived in more than one way, resulting in different parse trees.

For example, the string id + id * id can be parsed as:
(id)+(id∗id) where E→E+P and P→id∗id.
Alternatively, it can be parsed as id+(id∗id) by treating P first and then adding E.
Since there are multiple valid parse trees for the same string, the grammar is ambiguous.
Thus, the answer is Yes.

Q16. You are given two Context Free Grammar G1 and G2. 2 points

G1:
S→(S)S | ε

G2:
S→(S) | SS | ε

Now, consider the string w= (( ) (( ))) ((( ) ( ))) ( )

Which of the following statements is true?

- ○ S can be generated using only G1
- ○ S can be generated using only G2
- ◉ S can be generated using G1 and G2
- ○ S cannot be generated using either G1 or G2

Clear selection

Write an explanation for your answer. *

Grammar G1:
Productions: S→(S)S | ϵ
This grammar constructs balanced parentheses by matching each opening parenthesis with a closing one. It can generate the string w through recursive applications of S→(S)S, allowing for a well-formed structure.

Grammar G2:
Productions: S→(S) | SS | ϵ
This grammar also generates balanced parentheses but allows combining multiple sequences. It can derive w by splitting the string into two parts using S→SS and generating the required parentheses structure for each part.

Both grammars can generate the string w, making the correct answer S can be generated using G1 and G2.

Q17. Consider the two Context free grammars G1 and G2. Which of the following is L(G1) ∩ L(G2)?

2 points

G1:	G2:
$S \rightarrow ASA$	$S \rightarrow ASA$
$S \rightarrow 0$	$S \rightarrow \varepsilon$
$A \rightarrow 0 \mid 1$	$A \rightarrow 0 \mid 1$

- ○ L = {w ∈ Σ* | w contains 0 in the middle.}
- ○ L = {w ∈ Σ* | length of w is even.}
- ◉ L = {w ∈ Σ* | w is a even length string with 0 in the middle.}
- ○ L = {w ∈ Σ* | w is a odd length string with 0 in the middle.}
- ○ None of the above.

Clear selection

Write an explanation for your answer. *

The intersection of the languages generated by G1 and G2 is the set of strings that satisfy both conditions:

Having a 0 in the middle: G1 ensures this condition with its production S → ASA.
Having even length: G2 ensures this condition with its productions, especially S → ε which allows for the empty string (even length).
Therefore, the intersection of the languages will be strings that have both a 0 in the middle and an even length. This matches the description of the language L.

Q18. Consider the following Context Free Grammar, G.

G:
P → XP1 | XQ0
Q → XQ1 | XR0
R → XRX | #
X → 0 | 1

Now, answer the following questions.

What is the shortest string can be parsed from the grammar, G? If there are multiple correct answer write only one of those. 2 points

00#0

Write an explanation for your answer. *

Starting with P, the production P→XQ0 seems promising. Choosing X→0 gives a base of 0 and allows further expansion with Q.
For Q, using Q→XR0 allows for another minimal production where X→0 again leads to 00R0.
Finally, substituting R→# produces the string 00#0.

The length of a string, w can be expressed by |w|. 2 points

Find out how many distinct strings can be generated using the given Context Free Grammar, G, such that 9 <= |w| <= 11. Write the numeric value only.

36

Write an explanation for your answer. *

Productions Overview:
The productions allow strings to be built recursively with X producing either 0 or 1.
The structure also incorporates P, Q, and R that can contribute to the total length based on their recursive definitions.

String Length Contributions:
Each X adds 1 to the length.
Productions like P→XP1 and P→XQ0 effectively increase the length through recursion and terminal symbols.

R can terminate with #, which doesn't contribute to the string length but allows flexibility in structure.

Counting Distinct Strings:
By examining combinations that fit the lengths of 9, 10, and 11, we considered how many Xs and how the recursive structure could combine.
Each length was systematically calculated to ensure no duplicates were counted, resulting in a final count.
Thus, the total number of distinct strings generated by the grammar in the specified length range is 36.

Q19. Consider the following Context Free Grammar, G. 2 points

G:
S → aaS | abS | baS | bbS | X
X → aaY | baY
Y → aY | bY | ε

Write a eight length string that starts with "bb" and has only one parse three. Write the string only.

bbabaaab

Write an explanation for your answer. *

I started with "bb" as required.
To ensure a unique parse tree, I followed "bb" with "abaaab":
After "bb", we use

S→bbS which leads to a unique continuation with S→X.
Then, X→aaY ensures specific structure, where Y can generate "a" and "b" but is kept simple here to maintain uniqueness.

The complete string "bbabaaab" meets the length requirement and has a unique derivation based on the rules of the grammar.
Thus, the string is bbabaaab.

Q20. L = {w ε {a,b,c}*: $a^i b^j c^k$| i, j, k ≥ 0 and if i > j then k = i - j, else k = 0}

2 points

Which of the following CFGs generates L?

CFG A:
S → AB | AC
A → aA | ε
B → bB | b
C → aCc | ε

CFG B:
S → AB | A
A → aA | a | ε
B → bBc | b

CFG C:
S → A | B
A → aAb | Ab | ε
B → aBc | aCc
C → aCb | ε

CFG D:
S → AB | AC
A → aAb | ε
B → bB | ε
C → aCc | ε

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, w ∈ L, and doesn't parse any string, w ∉ L.

- ( ) A
- ( ) B
- ( ) C
- (•) D

Clear selection

Write an explanation for your answer. *

CFG D correctly captures these relationships through its productions:

A generates balanced pairs of a's and b's, allowing for cases where k=0 when i≤j.
The use of C ensures that when i>j, c's are generated to match the difference i−j.
In contrast, the other grammars fail to enforce the necessary conditions on the counts of c's correctly. Thus, CFG D accurately represents the language L.

Q21. L = {w ε {0,1}*: each b in w is followed by at least two a}

Note, for a language L, the CFG will be correct if and only if it can parse all the strings, $w \in L$, and doesn't parse any string, $w \notin L$.

Which of the following Context Free Grammar can generate L?

$S \rightarrow XY$
$X \rightarrow aX \mid \varepsilon$
$Y \rightarrow ZY \mid \varepsilon$
$Z \rightarrow PQ$
$P \rightarrow ba$
$Q \rightarrow aQ \mid a$

1 point

- (●) Correct
- ( ) Incorrect

Clear selection

$P \rightarrow aP \mid bQ \mid \varepsilon$
$Q \rightarrow aR \mid bT$
$R \rightarrow aS \mid bT$
$S \rightarrow aS \mid bQ \mid \varepsilon$
$T \rightarrow aT \mid bT$

1 point

- ○ Correct
- ◉ Incorrect

Clear selection

$S \rightarrow AbR \mid A$
$A \rightarrow aA \mid \varepsilon$
$R \rightarrow aAabR \mid A$

1 point

- ◉ Correct
- ○ Incorrect

Clear selection

Submit Clear form