CS 374 HW 3 Problem 2

quddus2, Hieu Huynh, Aldo Sanjoto

TOTAL POINTS

100 / 100

QUESTION 1

1 2A **50 / 50**

- ✓ **- 0 pts** Correct
- **- 37.5 pts** IDK
- **- 15 pts** No terminals in grammar
- **- 10 pts** Missing/incorrect explanation of non-terminals
- **- 5 pts** Partially correct explanation of non-terminals
- **- 10 pts** Minor mistake in CFG
- **- 50 pts** Incorrect CFG
- **- 15 pts** CFG doesn't allow 0 of all or some characters
- **- 15 pts** CFG doesn't maintain correct ordering of letters

QUESTION 2

2 2B **50 / 50**

- ✓ **- 0 pts** Correct
- **- 37.5 pts** IDK
- **- 10 pts** Incorrect explanation
- **- 40 pts** Incorrect CFG
- **- 10 pts** Minor error in CFG
- **- 5 pts** Minor mistake in explanation

Z 2a.)

$S \to aSe \mid X \mid \varepsilon$

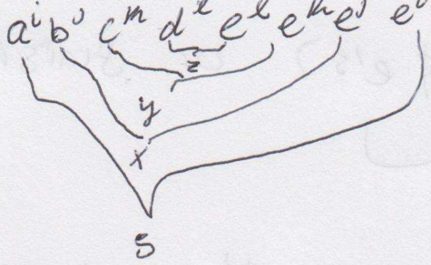
$X \to bXe \mid Y \mid \varepsilon$

$Y \to cYe \mid Z \mid \varepsilon$

$Z \to dZe \mid \varepsilon$

$\Leftrightarrow a^i b^j c^m d^l e^t$

#



for our language the number of e's is the total number a's b's c's & d's in our language. So every a contributes to an e, every b contributes to an e and so on up to character d.

~~Non~~ terminals: a, b, c, d, e

~~B~~ . Non terminals: S, X, Y, Z

Observing the production rule we start
with the ~~Start~~ S→ production where every time
we add an a an e is also added since a portion of the
number of e's is contributed by the number of a's in the string.
Moreover from the production rule S we can go back to S again
to repeat the a & e characters (ie aa...ee) or move on
to adding the b character or be finished by adding nothing → ε

⇓

Similar ~~the~~ Same thing is happening in production rule X, & Y but we just replace the a for b in production rule X (ie aabb...eeee) & we replace the a for c in production rule Y→ (ie aabbcc... eeeeee) From production rule Y we can go to production rule Z which will add ~~the same number of e's as~~ one e for every d added and from production rule Z either we can repeadictly add (d's & e's) or finish and go to → ε

(aabbcc dd...ee eeeee)

1 2A 50 / 50

- ✓ - 0 pts Correct
- - 37.5 pts IDK
- - 15 pts No terminals in grammar
- - 10 pts Missing/incorrect explanation of non-terminals
- - 5 pts Partially correct explanation of non-terminals
- - 10 pts Minor mistake in CFG
- - 50 pts Incorrect CFG
- - 15 pts CFG doesn't allow 0 of all or some characters
- - 15 pts CFG doesn't maintain correct ordering of letters

2B) $L = \{ w \in \{0,1\}^* \mid$ there's a prefix x of w s.t. $\#_1(x) > \#_0(x) \}$

Grammar for $w = x \cdot y$ s.t. $\#_1(x) > \#_0(x)$.

Firstly, y can be any strings. Thus, it has the following grammar: $Y \rightarrow 0Y \mid 1Y \mid \varepsilon$

Second, according to Lemma 5.3 (Lecture 5 notes), string that has the same number of 0s and 1s has the following grammar: $T \rightarrow 0T1 \mid 1T0 \mid TT \mid \varepsilon$

For $\#_1(w) > \#_0(w)$ grammar, we have 2 cases:

Case 1: w start with 0:

$w = x \cdot y$ with x, y are strings such that $\#_0(x) = \#_1(x)$ and $\#_1(y) > \#_0(y)$

Case 2: w start with 1:

$w = 1 \cdot x$ with x's string s.t. $\#_0(x) = \#_1(x)$

$w = 1 \cdot x$ with x's string s.t. $\#_1(x) > \#_0(x)$

The grammar becomes $X \rightarrow TX \mid 1T \mid 1X$

Combining all grammar results into the following:

$S \rightarrow TS \mid 1T \mid 1S$

$T \rightarrow 0T1 \mid 1T0 \mid TT \mid \varepsilon$

$Y \rightarrow 0Y \mid 1Y \mid \varepsilon$

which satisfies string of w that has a prefix x such that $\#_1(x) > \#_0(x)$.

2 2B 50 / 50

- ✓ - **0 pts** Correct
- **- 37.5 pts** IDK
- **- 10 pts** Incorrect explanation
- **- 40 pts** Incorrect CFG
- **- 10 pts** Minor error in CFG
- **- 5 pts** Minor mistake in explanation