Assignment #3

Saturday, March 5, 2022   11:21 PM

2.2

a. We will construct two grammars, one that recognizes *A* and one that recognizes *B*:

A:

$X \to YZ$
$Y \to aY|\varepsilon$
$Z \to bZc|\varepsilon$

B:

$X \to YZ$
$Y \to aYb|\varepsilon$
$Z \to cZ|\varepsilon$

Looking at both grammars, we can see that they are both context-free languages, thus A and B are context-free languages. We now consider the intersection of the two languages, where $A \cap B = \{a^n b^n c^n | \, n \geq 0\}$. We will check if this new language is content-free using the Pumping Lemma:

We first assume that $A \cap B$ is context-free. We choose the string $s = a^p b^p c^p$.
In appeasing condition two of the lemma, $|vy| > 0$, we must consider the two subcases that can arise:

1. Either *v* or *y* contain more than one type of symbol, ($vy = a^p b^p c^p$), then $uv^i xy^i z$ may contain an equal number of a, b, c, but not in the correct order. Thus, it violates the first condition of the lemma and is therefore not within $A \cap B$.
2. Both *v* and *y* contain only one unique letter (*v* or *y* does not have both *a's* and *b's* or *b's* and *c's*). However, this then means that there are not an equal number of *a, b, c* characters in the string. Thus, it violates the first condition of the lemma and is therefore not within $A \cap B$.

One of the two conditions must occur, and since both result in a contradiction, this then means our initial assumption of $A \cap B$ being context-free is false, and thus $A \cap B$ is not a context-free language. Since we have that *A* and *B* are context-free languages independently, but $A \cap B$ is not context-free, this means that the intersection of the languages is not a context-free language. Therefore, languages *A* and *B* are not closed under intersection.

b. We define two arbitrary context-free languages $A = (V_A, \sum, R_A, S_A), B = (V_B, \sum, R_B, S_A)$. We then construct a grammar, *G*, to recognize the union of *A* and *B*, $A \cup B$. We say this grammar is defined as $G = (V, \sum, R, S)$ where $V = V_A \cup V_B$ and $R = R_A \cup R_B \cup \{S \to S_A, S \to S_B\}$. For the sake of contradiction, we will say that *A* and *B* are closed under complementation. Since *A* and *B* are CFL's, we then know that $\bar{A}\ and\ \bar{B}$ are CFL's through our assumption. We then say they are closed under union, as per Kleene's Theorem of CFL's (Lecture 8). Thus, $\bar{A} \cup \bar{B}$ is closed under union and is a CFL. We can extend this even further through complement and say then that $\overline{\bar{A} \cup \bar{B}}$ is a CFL as well. Applying DeMorgan's Law, we see that $\overline{\bar{A} \cup \bar{B}} = A \cap B$, and conclude then that the intersection of *A* and *B* is a CFL and closed under intersection. However, this is a contradition to part A, of which stems from ourt assumption that *A* and *B* are closed under complement. As such, we conclude that our assumption must be incorrect, i.e. CFL's are not closed under complementation.

2.15 Our counter-example grammar will be $G = \{\{S\}, \{(,)\}, \{S \to (S), S \to \varepsilon\}, S\}$. Adding our assumption must be incorrect, i.e. CFL's are not closed under complementation.

2.15 Our counter-example grammar will be $G = \{\{S\}, \{(,)\}, \{S \to (S), S \to \varepsilon\}, S\}$. Adding the new rule of $S \to SS$ provides us with grammar G'. The grammar will generate a language that contains the string as symbols (( ), ( )), but not the star closure of A (A^*). It can be concluded then that the new grammar G' fails to prove that the class of context free languages is closed under star. This is only true if A^* contains the empty string, thus A^* must contain the empty string.

2.27

a. We will find the two distinct leftmost derivations to demonstrate ambiguity

1st Derivation:

$< STMT >$
$\to < IF - THEN >$
$\to If\ condition\ then < STMT >$
$\to If\ condition\ then < IF - THEN - ELSE >$
$\to If\ condition\ then\ if\ condition\ then < STMT > else < STMT >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else < STMT >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else < ASSIGN >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else\ a \coloneqq 1$

2nd Derivation:

$< STMT >$
$\to < IF - THEN - ELSE >$
$\to If\ condition\ then < STMT > else < STMT >$
$\to If\ condition\ then < IF - THEN > else < STMT >$
$\to If\ condition\ then\ if\ condition\ then < STMT > else < STMT >$
$\to If\ condition\ then\ if\ condition\ then < ASSIGN > else < STMT >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else < STMT >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else < ASSIGN >$
$\to If\ condition\ then\ if\ condition\ then\ a \coloneqq 1\ else\ a \coloneqq 1$

Both derivations end in the same result. However, the derivations diverged in the second step, meaning that it is ambiguous.

b. We will introduce a new variable to disallow adding an < IF-THEN > in the first variable placement of an < IF-THEN-ELSE>. Our new set of variables will be as follows:

$< STMT > \to < IF - THEN > | < ASSIGN > | < IF - THEN - ELSE >$
$< STMTNOIF \to < IF - THEN - ELSE > | < ASSIGN >$
$< IF - THEN - ELSE > \to If\ condition\ then < STMTNOIF > else < STMT >$
$< IF - THEN > \to If\ condition\ then < STMT >$
$< ASSIGN > \to a \coloneqq 1$

2.26

b. $S \to aSb\ |bSa\ |\ \varepsilon$

c. $S \to aSb\ |\ bSa\ |\ aS\ |\ \varepsilon$

2.30

a. We will use the Pumping Lemma to prove $A = \{0^n1^n0^n1^n | n \geq 0\}$ is not a context free language. We start by assuming A is a CFL and observe the string $s = 0^p1^p0^p1^p$. In appeasing condition two of the lemma, $|vy| > 0$, we must consider the two subcases that can arise:

1. Either v or y contain more than one type of symbol, (ex. $y = 0^p1^p$), then uv^ixy^iz may contain an

equal number of $0's$ and $1's$, but not in the correct order. Thus, it violates the first condition of the lemma and is therefore not within A.

2. Both *v* and *y* contain only one unique number. However, this then means that there are not an equal number of *0* and *1* characters in the string. Thus, it violates the first condition of the lemma and is therefore not within *A.*

One of the two cases must occur, and since both result in a contradiction, this then means our initial assumption of A being context-free is false, and thus A is not a context-free language

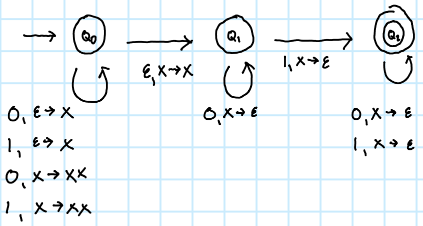
d. We will use the Pumping Lemma to prove $A = \left\{t_1\#t_2\# \dots t_k \middle| k \geq 2, \textit{ each } t_i \epsilon \{a,b\}^*, \textit{ and } t_i = t_j \textit{ for some } i = j\right\}$ is not a context free language. We start by assuming *A* is a CFL and observe the string $s = a^p b^p \# a^p b^p$. In appeasing condition two of the lemma, $|vy| > 0$, we must observe the following five subcases where $s = uv^i xy^i z$:

1. Neither *v* nor *y* can contain *#*, otherwise uv^0xy^0z does not contain the *#* character. Thus, it cannot be a member of *A*.
2. If both *v* or *y* are nonempty and are on the left-hand side of the *#*, then the string uv^3xy^3z is longer on the left side. Thus, the string *s* cannot be a member of *A*.
3. If both *v* or *y* are nonempty and are on the right-hand side of the *#*, then the string uv^3xy^3z is longer on the right side. Thus, the string *s* cannot be a member of *A*.
4. If only one of *v* and *y* are nonempty, then we treat the string as if both occur on the same side of *the #*, i.e. making it unsymmetrical and thus not a member of *A.*
5. If both *v* and *y* are nonempty and include the *#* character, then we know by the third condition of the lemma ($|xyz| \leq p$), that $v = b's$ and $y = a's$. In the case then of uv^3xy^3z, there are more *b*'s on the left hand side of the *#*. Thus, the string is not a member of *A*.

One of the above cases must occur, and since all result in a contradiction, this then means our initial assumption of A being context-free is false, and thus A is not a context-free language.

2.47

a.



b. We define the CFG for B as follows: $G = \{S, T, U, V, X\}, \{0,1, \varepsilon\}, R, S\}$ where R is the following:

$S \rightarrow TU$
$T \rightarrow VX$
$U \rightarrow V1V \mid V1X \mid X1T \mid T1T$
$V \rightarrow 00^* \mid \varepsilon$
$X \rightarrow 11^* \mid \varepsilon$

2.8

a. We construct the following grammar G_1:

$S \rightarrow 0T0 \mid 1T0 \mid 0T1 \mid 1T1$
$T \rightarrow s_1Ts_2 \mid 1, where\ \forall s_1, s_2\ \epsilon\ \{0,1\}^2 \cup \{0,1\}$

This generates a string around T. If it takes n+1 steps to generate a string s with this grammar, could be any string of the form $\sum^a 1 \sum^b$ where $n \leq a, b \leq 2n - 1$. Let us then take a string $x \in C_1$ with $|x| > 4$, as the grammar described above generates all the strings in C_1 shorter than five symbols. We now assume that $x = \sum^{i} 1 \sum^{j}$ with $i + j = n - 1, i < 2j - 1$ and $j \leq 2i - 1$. Without loss of generality, we will assume that $i < j$ and we claim that G_1 generates x in $i + 1$ steps. We then take the rule of $T \rightarrow s_1Ts_2$, with s_1 being one symbol long all the time and s_2 being two symbols long exactly $j - i$ times with $j < 2i$, implying $j - i < i$.

Thus, we have created a CFG for C_1, so it must therefore must be a CFL.

2.55 Per the DK test, every accept condition must have it such that all handles are forced, or that each accept state can only have rule within it. We can define the language of G_1 the union of two languages, $X = \{a^n b^n | \, n \geq 1\}$ *and* $Y = \{a^n b^{2n} | n \geq 1\}$. We will use the string *aaabbb* and *aaabbbbbb* to illustrate how the grammar violates the DK test. We first use the rule $S \rightarrow aSb|ab$ on *aaabbb:*

$$aaabbb \rightarrow aaSbb \rightarrow aSb \rightarrow S \rightarrow R.$$

We then use the rule $T \rightarrow aTbb \,|\, abb$ on *aaabbbbbb:*

$$\mathrm{aaabbbbbb} \rightarrow \mathrm{aaTbbbb} \rightarrow \mathrm{aTbb} \rightarrow \mathrm{T} \rightarrow \mathrm{R}$$

We see then that the leftmost reduction is the only one possible for the examples above, but that the handles are unequal, despite the fact that the initial parts of the string agree. Given that the accept state we have discovered consists of two separate rules, this violates the principles of the DK Test. Thus, we must say that the grammar G_1 is not a DCFG.

2.99

a. We can show that the language is not regular using the Myhill-Nerode theorem.

We define some infinite set of strings where $X = \{ \, (^n 0 \, | \, n \geq 0\}$. For all m, where $0 \leq m \leq n - 1$, $(^n 0$ is distinguishable from all $(^m 0$ because there exists some z, $z =)^n$, such that:

$$(^n 0)^n \in L_{REG}$$
$$(^n 0)^m \notin L_{REG}$$

where L_{REG} = { w ∈ Σ* | w is a valid REGULAR EXPRESSION}. Since X is an infinite set, it means that L_{REG} has a infinite index, thus proving it is not regular.

b.

We define some grammar that accepts L_{REG}:

$$S \rightarrow 1S \mid 0S \mid S \cup S \mid S^* \mid (S) \mid S \circ \mid €$$

Since we are able to create a grammar that describes L_{REG}, then we can say that it is a CFL.

c. Note: The special comma looks like ❜

⟨CFG⟩ → {⟨RULES⟩}
⟨RULES⟩ → ⟨RULE⟩ | ⟨RULE⟩❜⟨RULES⟩
⟨RULE⟩ → ⟨ ⟪NAME⟫ ⇒ ⟪STRING⟫ ⟩ |
⟨ ⟪NAME⟫ ⇒ ⟪STRING⟫ ⟪NAME⟫ ⟩ |
⟨ ⟪NAME⟫ ⇒ ⟪NAME⟫ ⟪STRING⟫ ⟩ |
⟨ ⟪NAME⟫ ⇒ ⟪NAME⟫ ⟪STRING⟫ ⟪NAME⟫ ⟩ |
⟨ ⟪NAME⟫ ⇒ ⟪NAME⟫ ⟪NAME⟫ ⟩ |
⟨ ⟪NAME⟫ ⇒ ⟪NAME⟫ ⟩
⟨NAME⟩ → A | B | C | … | X | Y | Z
⟨STRING⟩ → 0 | 1 | €