CS 181 Homework 6

Charles Zhang, 305-413-659

May 9, 2021

Problem 1

$$\Sigma = \{a, b\}$$

$$\{a^i b^j \mid 0 \le i \le j\}$$

CFG:
$G = (V, \Sigma, R, S)$, with $V = \{S, A, B\}$ and R = the rule set below.

$$S \longrightarrow \epsilon \mid aAb$$

$$A \longrightarrow aAb \mid B \mid \epsilon$$

$$B \longrightarrow bB \mid b$$

Justification:
The specified language is made up of strings of as followed by bs, where the number of as is less than or equal to the number of bs. This CFG works by first accepting the empty string. Otherwise, it builds strings with equal numbers of as and bs using the rule $A \longrightarrow aAb$, and then adds extra bs using the rule $B \longrightarrow bB \mid b$. This allows us to say that there will be at least as many bs as as.

I believe this CFG is unambiguous, since there is only one way to generate a given number of as in the string using $A \longrightarrow aAb$. Due to this, any extra bs must be generated using the rule $B \longrightarrow bB \mid b$, which means there is also only one way to generate a given number of bs in the string. Finally, the empty string can only be generated by the rule $S \longrightarrow \epsilon$. Taken together, this tells us there is only one way to generate each valid string.

Problem 2

$$\Sigma = \{0, 1\}$$
$$L = \{xx^R x \mid x \in \Sigma^\star\}$$

Proof (by contradiction):

- Assume L is context-free.
- Let p be the pumping length given by the pumping lemma for CFLs.
- Let $s = 0^p 1^p 1^p 0^p 0^p 1^p$, noting that $s \in L$, as we can say that the substring $x = 0^p 1^p$ and, as proved in HW1, $x^R = 1^p 0^p$, therefore, $s = xx^R x$.
- With s being a member of L and having a length greater than p, the pumping lemma for CFLs guarantees that s can be split into five substrings of the form $s = uvxyz$ such that $^{(1)}$for each $i \geq 0$, $uv^i xy^i z \in L_2$, $^{(2)}|vy| > 0$, and $^{(3)}|vxy| \leq p$.
- Note that s can be split into 3 substrings of length $2p$, $A = x$, $B = x^R$, and $C = x$.
- Note that each substring can be split into 2 substrings such that one substring contains p 0s and the other contains p 1s.
  - We will call these substrings A_0, A_1, B_0, B_1, C_0, and C_1, such that $s = A_0 A_1 B_1 B_0 C_0 C_1$.
- By condition (3) of the pumping lemma for CFLs, we know that $|vxy|$ has length of at most p.
- This tells us that there are 2 general cases: one where $|vxy|$ falls entirely within one of the substrings A, B, or C, and one where it doesn't.
- Create the string s' by pumping s using $i = 2$.
- Case 1:
  - In the case that vxy lies within 1 of the 3 substrings A, B, and C, there are 2 subcases: vxy is made up entirely of 1 symbol or vxy contains both symbols.
  - Case 1a:
    * In the first subcase, we know that one of A_0, A_1, B_0, B_1, C_0, or C_1 will be pumped.
    * This means the total number of 0s or 1s will increase, while the other stays constant.
    * Note that all runs of 0s and 1s are separated by a run of the other symbol.
    * If the number of 0s or 1s increases, we know that the length of exactly 1 such run will increase.
    * However, in order to satisfy the condition that $w = xx^R x$, this increase in length must be reflected in the other 2 runs of that symbol as well.
    * Therefore, the resulting string s' will not be in L.
  - Case 1b:
    * In the second subcase, we know a string containing 1 or more 0s and 1 or more 1s will be pumped.

        * In this case, a pattern of alternating runs of 0s and 1s will emerge within the substring.
        * However, this pattern will not occur in either of the other substrings.
        * This clearly breaks the condition that $w = xx^Rx$.
    - This tells us that the pumped string s' must no longer be a member of L.
    - This tells us that pumped strings of this case fail the pumping lemma for CFLs.
- Case 2:
    - In the case that vxy lies between 2 of the 3 substrings A, B, and C, there are 2 subcases: vxy lies between A and B, and vxy lies between B and C.
    - In both of the subcases, the substring disjoint from vxy is equal to x.
    - We know this substring remains the same between s and s'.
    - Note that x has an equal number of 0s and 1s, and, since strings in L have the form $w = xx^Rx$, we know that both s and s' must also have an equal number of 0s and 1s.
    - Since the substrings of the form A_0, A_1, etc. are each of length p, we know that, at most, vxy overlaps 2 such substrings.
    - Case 2a:
        * vxy crosses from A to B.
        * By the constraint that vxy only overlaps 2 of A_0, A_1, etc., we know that vxy must cross from A_1 to B_1.
            · More explicitly, it is impossible for vxy to overlap with A_0 or B_0.
        * This tells us that vxy is made up entirely of 1s
        * As a result, pumping vxy increases the number of 1s in the string, but keeps the number of 0s constant.
        * This violates the condition that the string s' has an equal number of 0s and 1s, telling us that $s' \notin L$.
    - Case 2b:
        * vxy crosses from B to C.
        * By the constraint that vxy only overlaps 2 of A_0, A_1, etc., we know that vxy must cross from B_0 to C_0.
            · More explicitly, it is impossible for vxy to overlap with B_1 or C_1.
        * This tells us that vxy is made up entirely of 0s
        * As a result, pumping vxy increases the number of 0s in the string, but keeps the number of 1s constant.
        * This violates the condition that the string s' has an equal number of 0s and 1s, telling us that $s' \notin L$.

  - Both possible subcases fail to create a string $s' \in L$ after pumping, telling us that pumped strings of this case fail the pumping lemma for CFLs.

- Both possible cases fail the pumping lemma for CFLs, which contradicts the assumption that L is context-free $\Rightarrow\!\Leftarrow$

Problem 3

R and S are FSLs
C is a language which can be represented by a PDA
G is a language which can be represented by an unambiguous CFG
A is a language which can be represented by an ambiguous CFG
I is an inherently ambiguous CFL
L is a language which *cannot* be represented by a CFG
X, given that $X \cup S = L$

1 FSL | 2 CFL and not FSL | 3 Inherently Ambiguous CFL | 4 Non-CFL

a) $R \cap L$: It could be a non-CFL; so it could be 1, 2, 3, or 4. Since L is a non-CFL language, we can't apply any closure properties to narrow it down any further.

b) A: It could be 1, 2, or 3. Languages that can be represented by CFGs are CFLs by definition. If all we know is that A can be represented by an ambiguous CFG, we have no indication of if A is inherently ambiguous or not. In addition, we have no indication of if A is strictly a CFL.

c) $\bar{C}$: It could be 1, 2, 3, or 4. As discussed in lecture, languages that can be represented by a PDA are CFLs, which are not closed under complementation, so we have no indication of what the resulting language may be.

d) G: It could be 1 or 2. If G can be represented by an unambiguous CFG, then it's a CFL, and it's obviously not inherently ambiguous or non-CFL. However, FSLs are a subset of CFLs, so it may be an FSL.

e) $\bar{L}$: It could be 2, 3, or 4. Since FSLs are closed under complementation, and L cannot be represented by a CFG (meaning it's definitely not an FSL), $\bar{L}$ can't be an FSL. CFLs and non-FSLs don't have the same restrictions, so it is possible for $\bar{L}$ to be one of them.

f) X: It must be 4. Since CFLs and FSLs are closed under union, if X was a CFL or FSL, the result of unioning it with S, an FSL, would be a CFL or FSL. Instead, it's a non-CFL, so X must be a non-CFL too.