Homework 5

Walker Bagley

March 24, 2023

1. (a) *Proof.* Take the context free languages $A = \{a^m b^n c^n \mid m, n \geq 0\}$ and $B = \{a^n b^n c^m \mid m, n \geq 0\}$.

$$\text{CFG for } A: \begin{array}{l} S \to aS \mid T \\ T \to bTc \mid \epsilon \end{array} \qquad \text{CFG for } B: \begin{array}{l} S \to Sc \mid T \\ T \to aTb \mid \epsilon \end{array}$$

Any string in language A must have the same number of b's and c's and any string in language B must have the same number of a's and b's. So, any string in $A \cap B$ must meet both conditions, yielding a new language $C = A \cap B = \{a^n b^n c^n \mid n \geq 0\}$. Let G be a CFG that generates C and $|V|$ be the number of nonterminal symbols. By the pumping lemma, there is a p s.t. for any s with $|s| \geq p$, the derivation of s uses the same nonterminal twice in one path. Let $s = a^p b^p c^p$ and write it as $s = uvxyz$ with $|vy| > 0$ and $|vxy| \leq p$. Then by the pumping lemma, $s = uxz \in C$.

Case 1a: v is all a's, y is all a's. Then there are fewer a's than $b's$.
Case 1b: v is all b's, y is all b's. Similar to 1a.
Case 1c: v is all c's, y is all c's. Similar to 1a.
Case 2a: v is all a's, y is all b's. Then there are fewer a's than c's or fewer b's than c's.
Case 2b: v is all b's, y is all c's. Similar to 2a.
Case 3: v or y contains two types of symbols. Then there are more of the third symbol than the others.
Then C is not context free, and context free languages are not closed under intersection. □

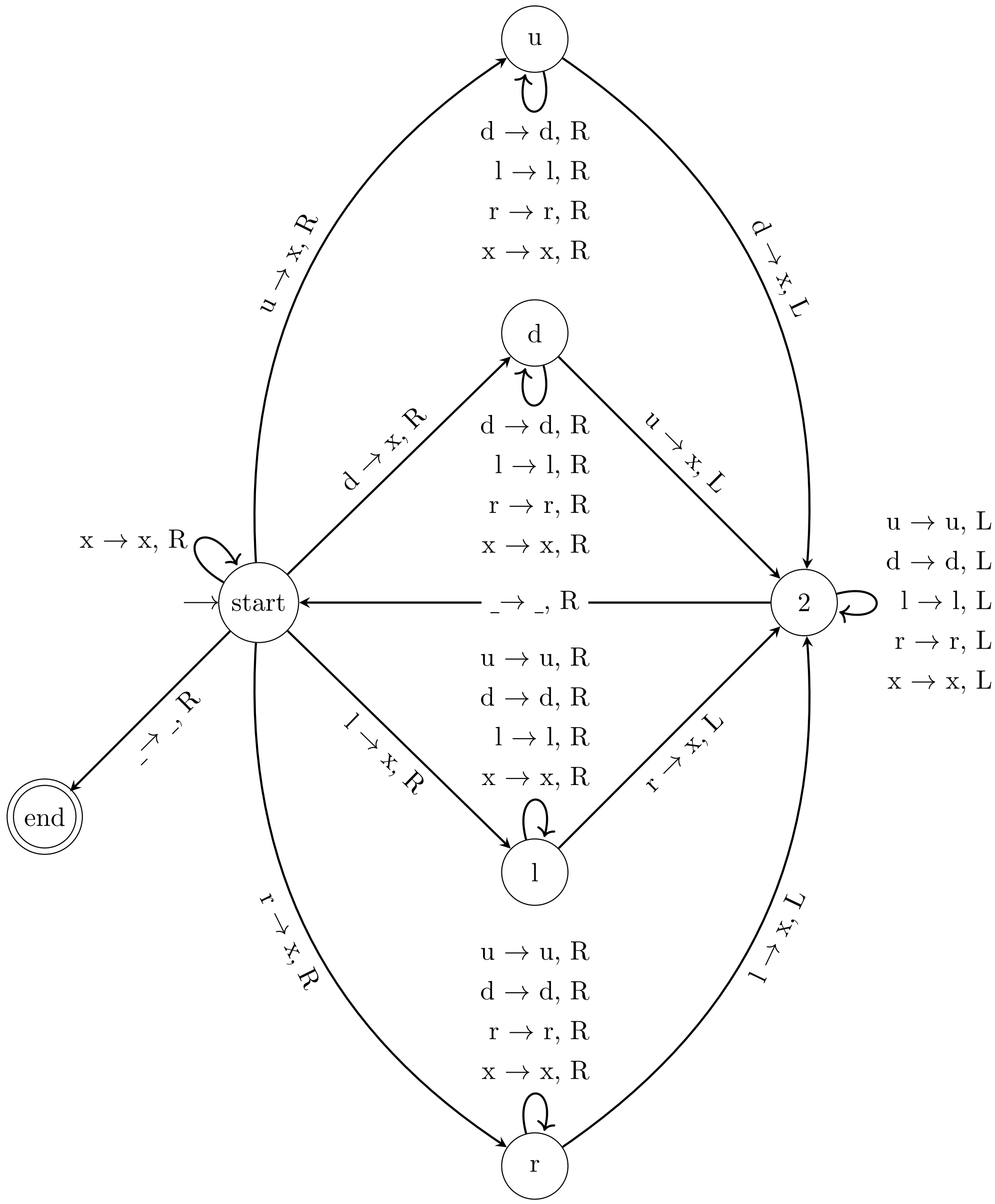
(b) *Proof.* Consider two context free languages, A and B. We know that context free languages are closed under union, so $A \cup B$ is also context free. Towards a contradiction, assume that context free languages are closed under complementation. Then $\overline{A} \cup \overline{B}$ is context free, and going a step further, $\overline{\overline{A} \cup \overline{B}}$ must also then be context free. However, by DeMorgan's, $\overline{\overline{A} \cup \overline{B}} = A \cap B$, but by the proof from question 1a, context free languages are not closed under intersection. Then we have a contradiction, and context free languages are therefore not closed under complementation. □

2. (a) *Proof.* Let C be the language defined by all sequences of moves that return to start. Then every string in C must contain n u's and d's and m l's and r's where $n, m \geq 0$. Let G be a CFG that generates C and $|V|$ be the number of nonterminal symbols. By the pumping lemma, there is a p s.t. for any s with $|s| \geq p$, the derivation of s uses the same nonterminal twice in one path. Let $s = \mathrm{u}^p \mathrm{l}^p \mathrm{d}^p \mathrm{r}^p$ and write it as $s = wvxyz$ with $|vy| > 0$ and $|vxy| \leq p$. Then by the pumping lemma, $s = wxz \in C$.

Case 1a: v is all u's and y is all u's. Then there are fewer u's than d's.
Case 1b: v is all l's and y is all l's. Similar to 1a.
Case 1c: v is all d's and y is all d's. Similar to 1a.
Case 1d: v is all r's and y is all r's. Similar to 1a.
Case 2a: v is all u's, y is all l's. Then there are fewer u's than d's or fewer l's than r's.
Case 2b: v is all l's, y is all d's. Similar to 2a.
Case 2c: v is all d's, y is all r's. Similar to 2a.
Case 3: v or y contains two types of symbols. Then there are an imbalanced number of u's and d's or l's and r's.

Thus, $s = wxz \notin C$ so C is not context free. □

(b)



3. (a) The Turing machine that recognizes STRETCH on $\Sigma^* = \{0,1\}^*$ will read in a symbol and overwrite it with an x. It will then read the next symbol and if it matches the previous, the machine overwrites it with an x and returns to the first x in the tape. It overwrites this x with the symbol it just read and then moves to the right to the next symbol that is not x. It repeats this process until there are no symbols left after the x's, when it overwrites all of them with a blank character and accepts the string. If the machine gets stuck at any point in this process, it rejects the string.

   i. Read in a 0 or 1 and overwrite it with an x
   ii. Read the next 0 or 1, and if it is not the same as the last symbol, reject
   iii. Otherwise, return to first x and replace with 0 or 1, whichever was last read
   iv. Sweep through all x's to the next 0, 1 or blank
   v. If the symbol is a 0 or 1, then return to (i) and if it is a blank space then overwrite the remaining x's with spaces and accept

   (b) *Proof.* Let M be a Turing machine that decides a Turing decidable language L over $\Sigma = \{0,1\}$. Then the following Turing machine decides STRETCH(L):

i. Run stages (i)-(v) of 3a to yield an unstretched string in Σ^*
ii. Simulate M on remaining string

□

(c) *Proof.* Let M be a Turing machine that recognizes some arbitrary Turing recognizable language L. Then we can modify the Turing machine from 3a to account for any symbols in the alphabet Σ of L by replacing the rules for 0's and 1's with the same rules for the entirety of Σ. Then we have a Turing machine that "unstreches" any string and decides the stretched function, which we can combine with M to get a Turing machine that recognizes STRETCH(L). Since the TM that recognizes the stretch operation is a decider and M is a recognizer, and because a decider is a more specific instance of a recognizer, then the combination of them both must be a recognizer. □