Homework 05

Aaron Wang

March 21 2025

1. **Non-closure properties of CFLs**

   (a) [Exercise 2.2a] Use the languages

   $$A = \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^n | m, n \geq 0\}$$

   $$B = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^m | m, n \geq 0\}$$

   to prove that context-free languages are not closed under intersection.

   Observe that A and B are context-free languages as there are CFGs that generate them.

   $$\begin{aligned} S &\to AB \\ A &\to Aa \mid \varepsilon \\ B &\to bBc \mid \varepsilon \end{aligned}$$

   CFG for A with start state S.

   $$\begin{aligned} S &\to AC \\ A &\to aAb \mid \varepsilon \\ C &\to Cc \mid \varepsilon \end{aligned}$$

   CFG for B with start state S.

   Towards a contradiction, assume that context-free languages are closed under intersection. This means that $C = A \cap B$ is a context-free language.

   Now observe that $C = A \cap B = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n | n \geq 0\}$. Suppose C is a context-free language. Let p be the pumping length given by the pumping lemma for context-free languages. Let $s = a^p b^p c^p$ and $s = uvxyz$ such that $|vy| > 0$ and $|vxy| \leq p$. The pumping lemma says that $s' = uv^2xy^2z \in C$.

   Case 1: v contains one type of symbol and y contains one type of symbol.
   In this case, $s' \notin C$ because if v and y contain the same type of symbol, then that symbol occurs more times than the other two and if they contain different symbols, the symbol not contained in v or y occurs less times than the types of symbols in v and y.

   Case 2: v or y contains more than one type of symbol.
   In this case, s' is no longer in the form $\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k$ as the string will be scrambled so $s' \notin C$.

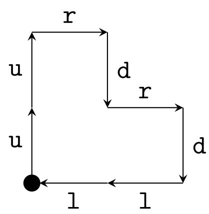
   Thus, by the Pumping Lemma for context-free languages, C is not context-free. Consequently, context-free languages are not closed under intersection.

   (b) [Exercise 2.2b] Use Problem 1a and DeMorgan's law to prove that context-free languages are *not* closed under complementation.

   Let A and B be context-free languages. Observe that $A \cup B$ is context-free because context-free languages are closed under union. Towards a contradiction, assume that context-free languages are closed under complementation. Using this assumption $\overline{A}$ and $\overline{B}$ must also be context-free. Consequently, $\overline{A} \cup \overline{B}$ must be context-free. $\overline{\overline{A} \cup \overline{B}}$ must also be context-free, and by DeMorgan's law $\overline{\overline{A} \cup \overline{B}} = A \cap B$ is context-free. In (1a), we have shown an example, where this is not true. Thus, context-free languages are *not* closed under complementation.

2. **There and back again.** Imagine a robot turtle that you can give instructions `u` (go up 1 cm), `d` (go down 1 cm), `l` (go left 1 cm), `r` (go right 1 cm). A program is a string of instructions.

Let C be the set of programs that make the turtle return to its starting point. For example, `uurdrdll` is in C, as shown in this picture:



(a) Prove that C is not context-free.

Observe $C\{\{\texttt{u},\texttt{r},\texttt{l},\texttt{d}\}|\#\texttt{u} = \#\texttt{d}$ and $\#\texttt{l} = \#\texttt{r}\}$. Towards a contradiction, assume that C is a context-free language. Let p be the pumping length given by the pumping lemma for context-free languages. Let $s = \texttt{u}^p\texttt{r}^p\texttt{d}^p\texttt{l}^p$ and $s = uvxyz$ such that $|uv| > 0$ and $|vxy| \leq p$. The pumping lemma says that $s' = uv^2xy^2z \in C$. Since $|vxy| \leq p$, v and y can only contain (1) the same symbol, or (2) two adjacent symbols.

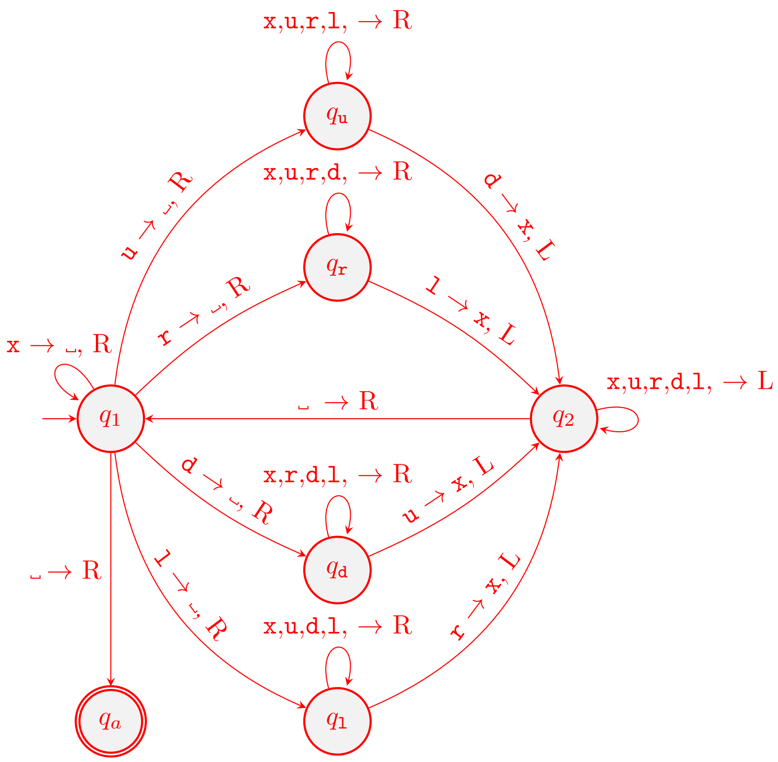
Case 1: v and y contain only one type of character. In this case, only one character will be pumped and s' will have more of that one character than its counter part, leading to the turtle being too far in that one direction, thus $s' \notin C$.

Case 2: v and y contain two adjacent symbols. In this case, the characters being pumped will be: `u` and `r`, `r` and `d`, or `d` and `l`. Since none of these pairs are complements, s' will not lead the robot turtle back to the starting point

In both cases, $s' \notin C$. Thus by the Pumping Lemma for context-free languages, C is not context free.

(b) Write a **formal description** of a Turing machine that decides C.

Let C be described by $M = (Q, \Sigma, \Gamma, \delta, q_1, q_a, q_{\text{reject}})$ where $Q = \{q_1, q_2, q_{\texttt{u}}, q_{\texttt{r}}, q_{\texttt{d}}, q_{\texttt{l}}, q_a, q_{\text{reject}})\}$, $\Sigma = \{\texttt{u},\texttt{r},\texttt{d},\texttt{l}\}, \Gamma = \{\texttt{u},\texttt{r},\texttt{d},\texttt{l},\texttt{x},\sqcup\}$, and δ is shown below with the state diagram where every missing transition leads to a reject state. For completeness, the head moves right on every transition into the reject state. (E.g. $\delta(q_{\texttt{u}}, \sqcup) = (q_{\text{reject}}, \sqcup, \text{R})$).



3. **Turing closure properties.** Let $\Sigma = \{0, 1\}$. Recall in HW2 we defined

$$\text{STRETCH}(w_1 w_2 \cdots w_n) = w_1 w_1 w_2 w_2 \cdots w_{n-1} w_{n-1} w_n w_n.$$

for any string $w_1 w_2 \cdots w_n \in \Sigma^*$ This induces an operation on languages,

$$\text{STRETCH}(L) = \{\text{STRETCH}(w) | w \in L\}.$$

(a) Write an **implementation-level** description of a Turing machine S that, on input $v \in \Sigma^*$, decides whether $v = \text{STRETCH}(u)$ for some u. Moreover, if S accepts v, then when it halts, the contents of the tape should be u. For example, if the input is `001100`, S should accept and the final contents of the tape should be `010`. But if the input is `001101`, S should reject.

Intuition behind S is this.
Steps 1 is base case, we have reached the end of string.
Steps 2-4: Check if the next two characters are the same.
Steps 5-6: Shift the remaining string to the left by 1 character until you reach space.
Step 7: Rewind to where we were before and continue.

$S =$ "On input string w"

1. If this symbol is $\sqcup$: *accept*
2. Check the symbol, move R, check the symbol.
3. If the symbols in stage 1 are different: *reject*.
4. If they are the same: move L, mark the symbol, move R
5. If symbol is $\sqcup$: go to step 7
6. Copy symbol one to the right to this spot, move R, go to step 5
7. move L until the you reach the marked symbol, unmark it, move R, go to step 1

(b) Prove that if L is a Turing-decidable language over Σ, then STRETCH(L) is also Turing-decidable. You should let M be a Turing machine that decides L, then use your answer to 3a to give an **implementation-level** description of a Turing machine that decides STRETCH(L). One of the lines of your description can be "Simulate M."

Let M' be the Turing machine that describes STRETCH(L)

$M' =$ "On input string w"

1. Simulate S
2. If S rejects w: *reject*.
3. If S accepts w: Move head to start of tape[1] and simulate M on the current tape.
4. *accept* or *reject* as M would.

(c) Prove that if L is a Turing-recognizable language, then STRETCH(L) is also Turing-recognizable.

Use the same answer as (b) with one change: 4. *accept*, *loop*, or *reject* as M would.

This change is necessary because M can now loop so M' can as well.

[1] In the case that this is too high level, what one could do is add a preprocessing step, Mark the first two symbols, Run S, and then at this point rewind to marked symbol, unmark it and continue.