CSCE 355, Assignment 5

First we review the pumping lemma for context-free languages.

Definition 1. We say that a language L is *CFL-pumpable* iff

> there exists an integer $p > 0$ such that
> for all strings $s \in L$ with $|s| \geq p$,
> there exist strings u, v, w, x, y with $uvwxy = s$, $|vwx| \leq p$, and $|vx| > 0$, such that
> for every integer $i \geq 0$,
> $uv^i wx^i y \in L$.

We will prove this in class:

Lemma 2 (Pumping Lemma for Context-Free Languages)**.** *For any language L, if L is context-free, then L is CFL-pumpable.*

From now one in this homework handout, "pumpable" will always mean CFL-pumpable.
Here is the contrapositive of the lemma, which is an equivalent statement:

Lemma 3 (Pumping Lemma for CFLs (contrapositive form))**.** *For any language L, if L is not pumpable, then L is not context-free.*

We will use the contrapositive form to prove that certain languages are not CFLs by showing that they are not pumpable. By definition, a language L is *not* pumpable iff

> for any integer $p > 0$,
> there exists a string $s \in L$ with $|s| \geq p$ such that
> for all strings u, v, w, x, y with $uvwxy = s$ and $|vwx| \leq p$ and $|vx| > 0$,
> there exists an integer $i \geq 0$ such that
> $uv^i wx^i y \notin L$.

Here is a template for a proof that a language L is not pumpable (and hence not context-free). Parts in brackets are to be filled in with specifics for any given proof. This is very much analogous to using the pumping lemma for regular languages.

> Given any $p > 0$,
> let $s :=$ [describe some string in L with length $\geq p$].
> Now for any u, v, w, x, y with $uvwxy = s$ and $|vwx| \leq p$ and $|vx| > 0$,
> let $i :=$ [give some integer ≥ 0 which might depend on p, s, u, v, w, x, and y].
> Then we have $uv^i wx^i y \notin L$ because [give some reason/explanation].

Note:

- We cannot choose p. The value of p could be any positive integer, and we have to deal with whatever value of p is given to us.
- We *can* and *do* choose the string s, which will differ depending on the given value of p (so the description of s has to use p somehow). We must choose s to be in L and with length $\geq p$, however.
- We cannot choose u, v, w, x, or y. These are given to us and could be any strings satisfying $uvwxy = s$, $|vwx| \leq p$, and $|vx| > 0$.
- We get to choose $i \geq 0$ based on all the previous values.

Example (given in class): Let $L := \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$. We show that L is not pumpable using the template:

> Given any $p > 0$,
> let $s := \mathtt{a}^p\mathtt{b}^p\mathtt{c}^p$. (Clearly, $s \in L$ and $|s| \geq p$.)
> Now for any u, v, w, x, y with $uvwxy = s$ and $|vwx| \leq p$ and $|vx| > 0$,
> let $i = 0$.
> Then we have $uv^iwx^iy = uv^0wx^0y = uwy \notin L$, which can be seen as follows: Since vwx and is a substring of s of length $\leq p$, it cannot contain both `a` and `c`. Then since $|vx| > 0$, uwy has length strictly less than $|s| = 3p$ but still has p many occurrences of either `a` or `c`. Thus uwy cannot have equal numbers of `a`, `b`, and `c`, and so $uwy \notin L$. It follows that L is not pumpable (hence not a CFL).

Exercises

1. (Optional but worth the effort)

   Textbook Exercise 5.1.1: Design context-free grammars for the following languages:

   (c): The set of all strings of `a`'s and `b`'s that are *not* of the form ww, that is, not equal to any string repeated.

   (d): The set of all strings with twice as many 0's as 1's.

2. We proved that the regular languages are closed under (string) homomorphic images (this is also in the textbook). Is the same true for the context-free languages? Explain.

3. Textbook Exercise 6.1.1(b,c) on pages 233–234 (part (a) has a solution on the book's website): Suppose the PDA $P = (\{q, p\}, \{0, 1\}, \{Z_0, X\}, \delta, q, Z_0, \{p\})$ has the following transition function:

   1. $\delta(q, 0, Z_0) = \{(q, XZ_0)\}$.
   2. $\delta(q, 0, X) = \{(q, XX)\}$.
   3. $\delta(q, 1, X) = \{(q, X)\}$.
   4. $\delta(q, \epsilon, X) = \{(p, \epsilon)\}$.

5. $\delta(p, \epsilon, X) = \{(p, \epsilon)\}$.
6. $\delta(p, 1, X) = \{(p, XX)\}$.
7. $\delta(p, 1, Z_0) = \{(p, \epsilon)\}$.

Starting from the initial ID (q, w, Z_0), show all the reachable ID's when the input w is

b) 0011.

c) 010.

4. Textbook Exercise 6.2.1(b,c) on page 241 (part (a) has a solution on the book's website): Design a PDA to accept [recognize] each of the following languages. You may accept either by final state or by empty stack, whichever is more convenient.

   b) The set of all strings of 0's and 1's such that no prefix has more 1's than 0's.

   c) The set of all strings of 0's and 1's with an equal number of 0's and 1's.

5. Do textbook Exercise 6.3.2 (Exercise 6.3.1 is similar and has a solution on the book's website):

   Textbook Exercise 6.3.2: Convert the grammar

$$\begin{aligned} S &\rightarrow \texttt{a}AA \\ A &\rightarrow \texttt{a}S \mid \texttt{b}S \mid \texttt{a} \end{aligned}$$

   to a PDA that accepts the same language by empty stack.

6. Consider the 1-state restricted PDA $P = (\{q\}, \{0, 1\}, \{X, Z_0\}, \delta, q, Z_0)$, where δ is given by

$$\begin{aligned} \delta(q, 0, Z_0) &= \{(q, \textbf{push } X)\} \qquad & \delta(q, 1, X) &= \{(q, \textbf{pop})\} \\ \delta(q, 0, X) &= \{(q, \textbf{push } X)\} \qquad & \delta(q, \epsilon, Z_0) &= \{(q, \textbf{pop})\} \end{aligned}$$

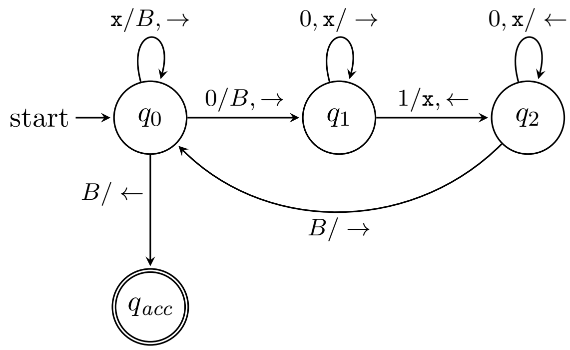
   (a) Using either the method of the book or the method I described in class, convert P to an equivalent context-free grammar.

   (b) (Optional) Simplify your grammar of the last problem as much as possible.

   This is similar to Exercise 6.3.3, which has a solution on the textbook's website.

7. Show that the language $L := \{ww \mid w \in \{0, 1\}^*\}$ is not CFL-pumpable (and hence not a CFL by the Pumping Lemma for CFLs).

8. Consider the standard, 1-tape Turing machine $M := \langle Q, \Sigma, \Gamma, \delta, q_0, B, F\rangle$ with input alphabet $\Sigma := \{0, 1\}$ and tape alphabet $\{0, 1, \texttt{x}, B\}$ (B is the blank symbol) given by the following transition diagram:



Give the complete computation path (sequence of IDs) of M on input "0101" (without the double quotes).