18.404/6.5400 Michael Sipser
Theory of Computation
Due via Gradescope at noon sharp, Thursday, October 3, 2024 **Problem Set 2**

Read all of Chapters 3 and 4.

0.1 Read and solve, <u>but do not turn in</u>: Book, 2.16 . [CFLs closed under $\cup$, $\circ$, $*$]
Solve by using both CFGs and PDAs.
Observe that this gives another proof that that all regular languages are CFLs.

0.2 Read and solve, <u>but do not turn in</u>: Book, 2.18 . [(CFL $\cap$ regular) is a CFL]
Note, problems marked with $^{\mathsf{A}}$ have solutions in the book.

0.3 Read and solve, <u>but do not turn in</u>: Book, 2.26 . [Chomsky normal form]

0.4 Read and solve, <u>but do not turn in</u>: Book, 2.30c . [CFL Pumping lemma]

(d) not context free
consider 1^(n)3^(n)2^(n)4^(n) where n is even, and greater than k
case 1: lvxl is odd, stops working for i=3
case 2: lvxl is even, stops working for i=2

1. Let $\Sigma = \{1,2,3,4\}$.
   (a) Let $C = \{w \mid w$ has equal numbers of 1s and 2s, and equal numbers of 3s and 4s$\}$. Show that C is not context free. 1^k3^k2^k4^k
   (b) Use (a) to show that the class of CFLs isn't closed under complement and intersection.
   #(1)=#(2) and #(3)=#(4) both are context-free
   (c) Let $D = C \cup (\Sigma\Sigma)^*$. Is D a CFL? Prove your answer.
   (c) not context free
   consider 1^(n)3^(n)2^(n)4^(n) where n is odd, and greater than n.
   case 1: lvxl is even , stops working for i=2
   case 2 : lvxl is odd, stops working for i=3
   (d) Let $E = C \cup \Sigma(\Sigma\Sigma)^*$. Is E a CFL? Prove your answer.

2. Let $G = (V, \Sigma, R, \langle\text{STMT}\rangle)$ be the following grammar. $\Sigma = \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{a:=1}\}$, $V = \{\langle\text{STMT}\rangle, \langle\text{IF-THEN}\rangle, \langle\text{IF-THEN-ELSE}\rangle, \langle\text{ASSIGN}\rangle\}$ and the rules are:

$$\begin{aligned}
\langle\text{STMT}\rangle &\to \langle\text{ASSIGN}\rangle \mid \langle\text{IF-THEN}\rangle \mid \langle\text{IF-THEN-ELSE}\rangle \\
\langle\text{IF-THEN}\rangle &\to \texttt{if condition then } \langle\text{STMT}\rangle \\
\langle\text{IF-THEN-ELSE}\rangle &\to \texttt{if condition then } \langle\text{STMT}\rangle \texttt{ else } \langle\text{STMT}\rangle \\
\langle\text{ASSIGN}\rangle &\to \texttt{a:=1}
\end{aligned}$$

   (a) Show that G is ambiguous. if condition then if condition then a=1 else a=1
   (b) Give a new unambiguous grammar that generates $L(G)$. (exactly the same language) (Explain how your grammar avoids the ambiguity. A formal proof that it is unambiguous is not necessary.)

3. A ***queue automaton*** is like a push-down automaton except that the stack is replaced by a queue. A ***queue*** is a tape allowing symbols to be written only on the left-hand end and read only at the right-hand end. Each write operation (we'll call it a *push*) adds a symbol to the left-hand end of the queue and each read operation (we'll call it a *pull*) reads and removes a symbol at the right-hand end. As with a PDA, the input is placed on a separate read-only input tape, and the head on the input tape can move only from left to right. The input tape contains a cell with a blank symbol following the input, so that the end of the input can be detected. A queue automaton accepts its input by entering a special accept state at any time. Show that a language can be recognized by a deterministic queue automaton iff the language is Turing-recognizable.

if an NPDA is not using its stack, its equivalent to an NFA
so well construct an NFA as the following : has the same states as the NPDA, select the following transitions only : should read “bottom” from stack, shouldnt push anything to stack
take the complement of this NFA and intersect it with given CFG. is this empty?

4. Show that a language is decidable iff some enumerator enumerates the language in string order. (***String order*** is the standard length-increasing, lexicographic order, see text p 14).

5. Let C be a language. Prove that C is Turing-recognizable iff a decidable language D exists such that $C = \{x \mid \exists y \in \{0,1\}^* \ (\langle x, y\rangle \in D)\}$. (Hint: You must prove both directions of the "iff". The $(\longleftarrow)$ direction is easier. For the $(\longrightarrow)$ direction, think of y as providing additional information that allows you to confirm when $x \in C$, but without the possibility of looping.)

6. Consider the problem of testing whether a pushdown automaton ever uses its stack. Formally, let $PUSHER = \{\langle P\rangle \mid P$ is a PDA that pushes a symbol on its stack on some (possibly non-accepting) branch of its computation at some point on some input $w \in \Sigma^*\}$. Show that $PUSHER$ is decidable. (Hint: Use a theorem from lecture to give a short solution.)

7.* (optional) Let the ***rotational closure*** of language A be $RC(A) = \{yx \mid xy \in A$ where $x, y \in \Sigma^*\}$. Show that the class of CFLs is closed under rotational closure.

https://cs.stackexchange.com/questions/7831/easy-proof-for-context-free-languages-being-closed-under-cyclic-shift