# University of Nevada, Las Vegas Computer Science 456/656 Fall 2024

## Answers to Assignment 3:

$\mathcal{P}$ means $\mathcal{P}$–TIME.

1. True/False. If the answer is not known to science at this time, enter "O" for Open.

   (i) **T** co-$\mathcal{P} = \mathcal{P}$.

   (ii) **O** co-$\mathcal{NP} = \mathcal{NP}$.

   (iii) **T** co-$\mathcal{P}$–SPACE = $\mathcal{P}$–SPACE.

   (iv) **T** Block placement problems are $\mathcal{NP}$.

   (v) **T** Sliding block problems are $\mathcal{P}$–SPACE.

   (vi) **O** $\mathcal{P}$–SPACE = $\mathcal{NP}$

   (vii) **O** Regular expression equivalence is $\mathcal{P}$.

   (viii) **T** Regular expression equivalence is decidable.

   (ix) **F** Context-free grammar equivalence is decidable.

   (x) **T** Every regular language is context-free.

   (xi) **F** The language C++ is context-free.

   (xii) **F** The intersection of any two context-free languages is context-free.

   (xiii) **F** The complement of any context-free language is context-free.

   (xiv) **T** Every language is countable.

   (xv) **F** For any real number $x$, there is a program that prints the decimal expansion of $x$.

   (xvi) **F** For any real number $x$, there is a machine that decides whether a fraction is less than $x$.

   (xvii) **T** There are only countably many decidable binary languages.

   (xviii) **T** Given a regular grammar $G$ with $n$ variables, there exists an NFA with $n$ states that accepts $L(G)$.

   (xix) **T** $\left\{a^i b^j c^k : i = k\right\}$ is a context-free language.

   (xx) **O** Given an integer $n$ written in binary notation, it is possible to find the prime factors of $n$ in polynomial time.

   (xxi) **T** Given an integer $n$ written in binary notation, it is possible to decide whether $n$ is prime in polynomial time.

   (xxii) **F** Any language generated by a grammar is decidable.

   (xxiii) **T** The complement of any decidable language is decidable.

   (xxiv) **T** The union of any two decidable languages is decidable.

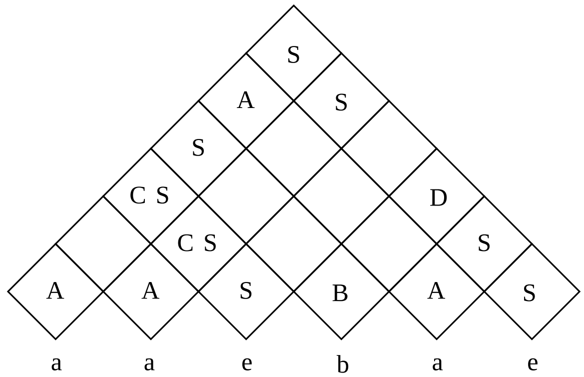
   (xxv) **T** The complement of any undecidable language is undecidable.

   (xxvi) **F** The union of any two undecidable languages is undecidable.

   (xxvii) **F** Every context-free language is accepted by some DPDA.

2. Let $L$ be the language generated by the following CNF (Chomsky Normal Form) grammar.

$S \to AS \,|\, CD \,|\, e$
$C \to AS$
$D \to BS$
$A \to a$
$B \to b$




Use the CYK algorithm to determine whether $aaebae \in L$.

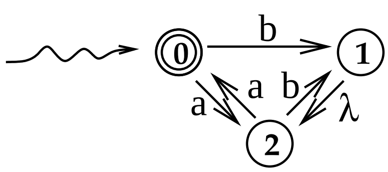
Since the top cell contains the start symbol, $aaebae \in L$.

3. Give a context-free grammar for $L = \{w \in \{a,b\}^* : \#_a(w) > \#_b(w)\}$, that is, strings which have more $a$'s than $b$'s.

This is a surprisingly hard problem. I found this grammar on the internet.

$S \to a$
$S \to aS$
$S \to bSS$
$S \to SbS$
$S \to SSb$

4. Write a regular grammar which generates the language accepted by the NFA illustrated below.




$S \to bA$
$A \to B$
$S \to aB$
$B \to aS$
$B \to bA$

5. List the grammar classes and language classes of the Chomsky hierarchy.

See the handout grammar1.pdf.

6. Give two context-free languages whose intersection is not context-free.

Consider the following three languages.

(a) $\{a^i b^j c^k : i = j\}$
(b) $\{a^i b^j c^k : i = k\}$
(c) $\{a^i b^j c^k : j = k\}$

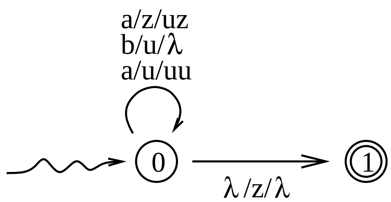
All three of those languages are context-free, but the intersection of any two of them is $\{a^n b^n c^n\}$ which is not context-free.

7. Write a grammar for the Dyck language (using 'a' and 'b' instead of parentheses) and give a derivation of the string abaabb.

1. $S \to aSbS$
2. $S \to \lambda$

$S \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSbS \Rightarrow abaaSbSbS \Rightarrow abaabSbS \Rightarrow abaabbS \Rightarrow abaabb$

8. Draw a PDA which accepts the Dyck language, using $a$ and $b$ instead of left and right parentheses, respectively.




9. In the following, do not write more than necessary. Your answers should be concise and correct.

   (a) Explain the verification definition of the class $\mathcal{NP}$.

   A language $L$ is $\mathcal{NP}$ if and only if there exists a machine $V$ and an integer $k$ such that the following two conditions hold.

      i. If $w \in L$ there exists a string $c$ (called a *certificate* for $w$) such that $V$ accepts the string $w, c$ in $O(n^k)$ time, where $n = |w|$.

      ii. If $w \notin L$ then $V$ does not accept any string of the form $w, c$.

   (b) What could be a certificate to prove that a given Boolean expression is in the language SAT?

   If $E$ is a satisfiable Boolean expression, then a satisfying assignment of $E$ can be used as a certificate for $E$.