**University of Nevada, Las Vegas**
**Computer Science 456/656 Spring 2020**

## Answwers to Final Examination May 14, 2020

`Thu May 14 18:07:59 PDT 2020`

The entire examination is 405 points.

1. [5 points each] True or False. If the question is currently open, write "O" or "Open."

   (i) **T** The complement of every regular language is regular.

   (ii) **F** The complement of every context-free language is context-free.

   (iii) **T** The complement of any $\mathcal{P}$-TIME language is $\mathcal{P}$-TIME.

   (iv) **O** The complement of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (v) **T** The complement of any $\mathcal{P}$-SPACE language is $\mathcal{P}$-SPACE.

   (vi) **T** The complement of every recursive language is recursive.

   (vii) **F** The complement of every recursively enumerable language is recursively enumerable.

   (viii) **T** Every language which is generated by a general grammar is recursively enumerable.

   (ix) **F** The context-free membership problem is undecidable.

   (x) **T** The factoring problem, where inputs are written in binary notation, is co-$\mathcal{NP}$.

   (xi) **T** If $L_1$ reduces to $L_2$ in polynomial time, and if $L_2$ is $\mathcal{NP}$, and if $L_1$ is $\mathcal{NP}$-complete, then $L_2$ must be $\mathcal{NP}$-complete.

   (xii) **F** Given any context-free grammar $G$ and any string $w \in L(G)$, there is always a unique leftmost derivation of $w$ using $G$.

   (xiii) **F** For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.

   (xiv) **F** The question of whether two regular expressions are equivalent is known to be $\mathcal{NP}$-complete.

   (xv) **T** The halting problem is recursively enumerable.

   (xvi) **T** The union of any two context-free languages is context-free.

   (xvii) **F** The question of whether a given Turing Machine halts with empty input is decidable.

   (xviii) **T** The class of languages accepted by non-deterministic finite automata is the same as the class of languages accepted by deterministic finite automata.

(xix) **F** The class of languages accepted by non-deterministic push-down automata is the same as the class of languages accepted by deterministic push-down automata.

(xx) **T** The class of languages accepted by non-deterministic Turing Machines is the same as the class of languages accepted by deterministic Turing Machines.

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

(xxii) **T** If $L_1$ reduces to $L_2$ in polynomial time, and if $L_2$ is $\mathcal{NP}$, then $L_1$ must be $\mathcal{NP}$.

(xxiii) **T** Let $\pi$ be the ratio of the circumference of a circle to its diameter. The problem of whether the $n^{\text{th}}$ digit of the decimal expansion of $\pi$ for a given $n$ is equal to a given digit is decidable.

(xxiv) **T** There cannot exist any computer program that can decide whether any two C++ programs are equivalent.

(xxv) **T** Every context-free language is in the class $\mathcal{P}$-TIME.

(xxvi) **T** Every regular language is in the class $\mathcal{NC}$

(xxvii) **T** The language of all binary numerals for multiples of 23 is regular.

(xxviii) **F** The language of all binary strings which are the binary numerals for prime numbers is context-free.

(xxix) **T** Every context-free grammar can be parsed by some non-deterministic top-down parser.

(xxx) **T** If anyone ever proves that $\mathcal{P} = \mathcal{NP}$, then all one-way encoding systems will be insecure.

(xxxi) **T** If a string $w$ is generated by a context-free grammer $G$, then $w$ has a unique leftmost derivation if and only if it has a unique rightmost derivation.

(xxxii) **T** A language $L$ is in $\mathcal{NP}$ if and only if there is a polynomial time reduction of $L$ to SAT.

(xxxiii) **T** A language $L$ is in $\mathcal{P}$-SPACE if and only if there is a polynomial time reduction of $L$ to some contexc-sensitive language.

(xxxiv) **T** A language $L$ is in recursively enumerable if and only if there is a recursive reduction of $L$ to the halting problem.

(xxxv) **T** A language $L$ is in $\mathcal{P}$-TIME if and only if there is an $\mathcal{NC}$ reduction of $L$ to Boolean satisfiability.

2. [5 points] What class of machines accepts the class of context free languages?

   PDA's

3. [5 points] What class of machines accepts the class of recursively enumerable languages?

   TM's

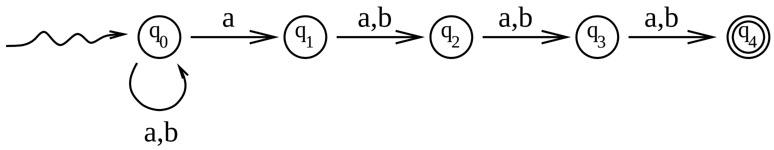
4. [20 points] Using the context-free grammar with start symbol $S$ and productions listed below, write two different leftmost derivations (not parse trees) of the string *iibwaanea*

$S \to a$
$S \to bLn$
$S \to wS$
$S \to iS$
$S \to iSeS$
$L \to \lambda$
$L \to SL$

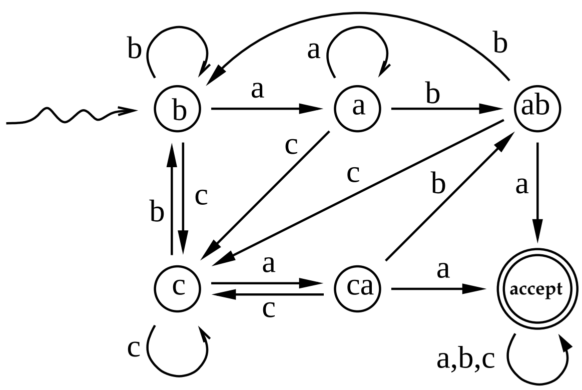
$S \Rightarrow iS \Rightarrow iiSeS \Rightarrow iibLneS \Rightarrow iibSLneS \Rightarrow iibwSLneS \Rightarrow iibwaLneS \Rightarrow iibwaSLneS \Rightarrow iibwaaLneS \Rightarrow iibwaaneS \Rightarrow iibwaanea$

$S \Rightarrow iSeS \Rightarrow iiSeS \Rightarrow iibLneS \Rightarrow iibSLneS \Rightarrow iibwSLneS \Rightarrow iibwaLneS \Rightarrow iibwaSLneS \Rightarrow iibwaaLneS \Rightarrow iibwaaneS \Rightarrow iibwaanea$

5. [20 points] Draw an NFA with five states which accepts the language described by the regular expression $(a + b)^*a(a + b)(a + b)(a + b)$




6. [20 points] Draw a DFA which accepts the language $L$ over the alphabet $\{a, b, c\}$ consisting of all strings which contain either $aba$ or $caa$ as a substring. (My answer has six states.)

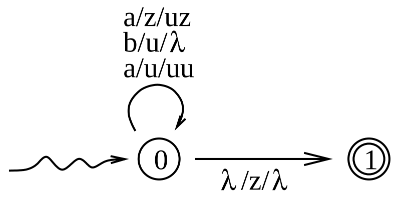



7. [20 points] Find a context-free grammar which generates the language $L = \left\{a^i b^j c^k : i = j \text{ or } i = k\right\}$

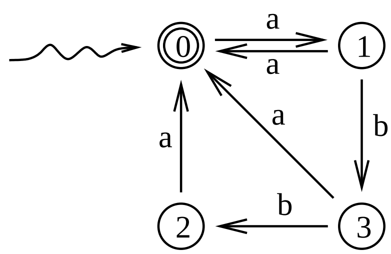
$L$ is the union of two context-free languages which have straightforward grammars.

$S \to S_1 \,|\, S_2$
$S_1 \to S_1 c \,|\, T$
$T \to aTb \,|\, \lambda$
$S_2 \to aS_2 c \,|\, B$
$B \to bB \,|\, \lambda$

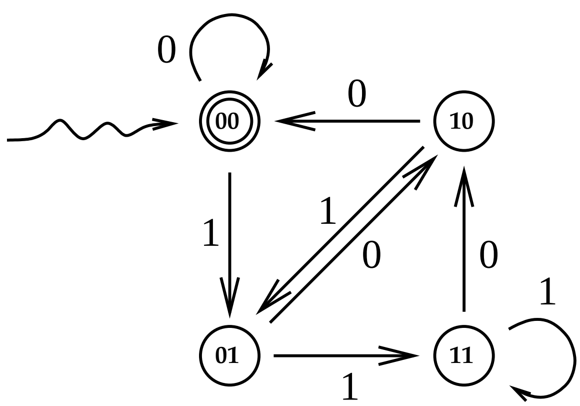
8. [20 points] Draw a state diagram for a PDA that accepts the Dyck language. (For ease of grading, use $a$ and $b$ instead of "[" and "]")




9. [20 points] Draw the state diagram for a DFA that accepts the language described by the regular expression `(a(λ+b+bb)a)`$^*$




10. [20 points] Let $L$ be the language of all binary numerals for positive integers which are multiples of 4. Thus, for example, the binary numerals for 0, 4, 8, 12, 16, 20 ... are in $L$. We allow a binary numeral to have leading zeros; thus (for example) $0011100 \in L$, since it is a binary numeral for 28. Draw a DFA with four states which accepts $L$.




(To think about: what do the labels mean?)

11. [20 points] Prove that every decidable language can be enumerated in canonical order by some machine.

Let $L$ be a decidable language over an alphabet $\Sigma$. let $M$ be a machine that decides $L$. let $w_1, w_2, \ldots$ be all the strings in $\Sigma^*$ in canonical order. The following program decides $L$:

For all $i$ from 1 to $\infty$
    If($M$ accepts $w_i$)
        Write $w_i$

The outer loop will run forever, since the conditional statement can always be evaluated in finite time.

12. [20 points] State the Church-Turing Thesis.

"Every machine is equivalent to some Turing machine."

13. [20 points] Find a $\mathcal{P}$-time reduction of the subset sum problem to the partion problem.

An instance of the subset sum problem is a number $K$ and a set of $n$ objects of positive weight. A solution to that instance is a subset of the objects whose total weight is $K$. Let $S$ be the sum of the weights of the objects. Without loss of generality, $K \leq S$, since otherwsie there can be no solution.

Given that instance, we define an instance of the partition problem to consist of the $n$ objects together with two additional objects, one of weight $K+1$ and the other of weight $S-K+1$. The sum of the weights of thos $n+2$ objects is $2S+2$. That instance has a solution if there is the objects can be partitioned into two equal weight sets, that is, each set must have weight $S+1$.

We now prove that the instance of the subset sum problem has a solution if and only if our instance of the partition problem has a solution. Suppose there is a subset of the set of original weights whose total weight is $K$. Then, that set combined with the one new object of weight $S-K+1$ is a set whose weight is half the total.

Conversely, suppose that the instance of the partition problem has a solution. The two new weights cannot both be in either half, since their total weight is $S+2$. Thus one of the two halves has the new object of weight $S-K+1$. The remaining object in that set must have total weight $K$.

14. [20 points] Let $\Sigma$ be the Boolean alphabet. Here is a "proof" that every language $L$ over $\Sigma$ is decidable.

"For any $n \geq 0$, let $\Sigma^n$ be the set of strings over $\Sigma$ of length $n$, and let $L_n = L \cap \Sigma^n$. $L_n$ is finite, in fact, $|L_n| \leq 2^n$. Thus, $L_n$ is decidable. Let $\mathcal{P}$ be the following program:

Read a string $w \in \Sigma^*$.
    Let $n = |w|$
If $(w \in L_n)$        (Remember: $L_n$ is decidable)
    Write "yes" $(w \in L)$
else
    Write "no" $(w \notin L)$

$\mathcal{P}$ decides $L$. We conclude that every language is decidable."

But, since HALT is undecidable, this proof can't be right. What's wrong with it?

Several students tried to give an answer by writing a proof that the halting problem is undecidable. That is not the point. We know that the halting problem is undecidable. The problem is to find the logical flaw in the "proof" that I gave.

No one gave the correct answer. I will not reveal my answer to this question, since I might want to use it again.