x

CSC 456/656 Fall 2022 Answers for Topics for Second Examination

This is the **corrected** version, `Tue Oct 25 12:54:27 PDT 2022` If you find any other errors, send me email **immediately**.

1. **Classes and Operators.** True/False/Open.

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

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

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

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

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

   (vi) **F** The complement of any RE language is RE.

   (vii) **T** The union of any two regular languages is regular.

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

   (ix) **T** The union of any two $\mathcal{P}$–TIME languages is $\mathcal{P}$–TIME

   (x) **T** The union of any two $\mathcal{NP}$ languages is $\mathcal{NP}$.

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

   (xii) **T** The union of any two RE languages is RE.

   (xiii) **T** The intersection of any two regular languages is regular.

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

   (xv) **T** The intersection of any two $\mathcal{NP}$ languages is $\mathcal{NP}$.

   (xvi) **T** The intersection of any two decidable languages is decidable.

   (xvii) **T** The intersection of any two RE languages is RE.

   (xviii) **T** The Kleene closure of any regular language is regular.

   (xix) **T** The Kleene closure of any context-free language is context-free.

   (xx) **T** The Kleene closure of any $\mathcal{P}$–TIME language is $\mathcal{P}$–TIME.

   (xxi) **T** The Kleene closure of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (xxii) **T** The Kleene closure of any decidable language is decidable.

   (xxiii) **T** The Kleene closure of any RE language is RE.

2. **Definitions**

   (i) A language is *regular* if it is accepted by a finite state machine. Equivalently, a language is *regular* if it is described by a regular expression. Equivalently, a language is *regular* if it is generated by a regular grammar.

(ii) A language is *context-free* if it is accepted by a PDA. Equivalently, a language is *context-free* if it is generated by a context-free grammar.

(iii) A language is $\mathcal{P}$–TIME if it can be decided by some machine in time which is polynomial in the length of the input string.

(iv) A language is $\mathcal{NP}$ if it is accepted by a non-deterministic machine, in time which is polynomial in the length of the input string provided the machine makes all the correct guesses. Equivalently, a language is $\mathcal{NP}$ time if it is accepted in polynomial time be a non-deterministic machine if the machine is provided with a polynomial length guide string for each member of the language.

(v) A language L is $\mathcal{NP}$-complete if there is a polynomial time reduction of any given $\mathcal{NP}$ language to L.

(vi) A language is decidable, or recursive, if it is decided by some machine. Equivalently, a language is recursive if there is a machine that enumerates the language in canonical order.

(vii) A language is RE if there is a machine which enumerates the language. Equivalently, a language is RE if there is some machine which **accepts** the language.

(viii) A language is co-RE if its complement is RE.

(ix) A function f is recursive if there is a machine which computes f.

(x) A real number x is recursive if there is a machine which runs forever, writing the decimal expansion of x. Equivalently, x is recursive if there is a machine which, given an integer i, computes the digit in the i^{th} place of the decimal expansion of x. Equivalently, x is recursive if there is a machine which can decide whether a given rational number is less than x.

(xi) If L_1 and L_2 are languages over alphabets Σ_1 and Σ_2, respectively, a reduction of L_1 to L_2 is a function $R : \Sigma_1^* \to \Sigma_2^*$ such that for any $w \in \Sigma_1^*$, $w \in L_1$ if and only if $R(w) \in L_2$.

3. **Countability** True/False/Open.

(i) **T** The set of integers is countable.

(ii) **T** The set of rational numbers is countable.

(iii) **F** The set of real numbers is countable.

(iv) **T** The set of recursive real numbers is countable.

(v) **F** The set of functions from integers to integers is countable.

(vi) **T** The set of recursive functions from integers to integers is countable.

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

(viii) **T** Every recursive language is countable.

(ix) **F** The set of languages over the binary alphabet is countable.

(x) **T** The set of decidable languages over the binary alphabet is countable.

(xi) **T** The set of RE languages over the binary alphabet is countable.

4. Other True/False/Open Questions.

(i) **O** If a language is both $\mathcal{NP}$ and co-$\mathcal{NP}$, it must be $\mathcal{P}$–TIME.

(ii) **T** If a language is both RE and co-RE, it must be decidable.

(iii) **F (Hard!)** Let L be any RE language over an alphabet Σ, and let M be a machine that accepts L. For any $w \in L$, let $T(w)$ be the number of steps M takes to accept w. For any integer $n \geq 0$, let $F(n) = \max\{T(w) : w \in L \text{ and } |w| = n\}$. Then F must be a recursive function.

5. List six languages or problems known to be $\mathcal{NP}$-complete.

6. Give a polynomial time reduction of 3SAT to the independent set problem.

7. Give a polynomial time reduction the subset sum problem to partition.

8. Let L be a decidable. Write a program which enumerates L in canonical order.

9. Know what a guide string is.

10. State the pumping lemma for context-free languages.

11. Give an example of a language which is context-sensitive, but not context-free.

12. Let G be the CF grammar given below, with start symbol E, which stands for **expression**. Consider the LALR parser given for G.

    (i) Which entries show that addition and subtraction are left associative and have equal precedence?
    Ans: Rows 3 and 5, columns $+$ and $-$.

    (ii) Which entry shows that negation has precedence over multiplication?
    Ans: Row 9, column $*$.

    (iii) Walk through the computation of the LALR parser if the input string is $x - (x + x)$

    (iv) Walk through the computation of the parser if the input string is $-x * x$

1. $E \to E +_2 E_3$
2. $E \to E -_4 E_5$
3. $E \to E *_6 E_7$
4. $E \to -_8 E_9$
5. $E \to (_{10} E_{11})_{12}$
6. $E \to x_{13}$

	ACTION							GOTO
	x	$+$	$-$	$*$	(	)	\$	E
0	$s13$		$s8$		$s10$			1
1		$s2$	$s4$	$s6$			HALT	
2	$s13$		$s8$		$s10$			3
3		$r1$	$r1$	$s6$		$r1$	$r1$	
4	$s13$		$s8$		$s10$			5
5		$r2$	$r2$	$s6$		$r2$	$r2$	
6	$s13$		$s8$		$s10$			7
7		$r3$	$r3$	$r3$		$r3$	$r3$	
8	$s13$		$s8$		$s10$			9
9		$r4$	$r4$	$r4$		$r4$	$r4$	
10	$s13$		$s8$		$s10$			11
11		$s2$	$s4$	$s6$		$s12$		
12		$r5$	$r5$	$r5$		$r5$	$r5$	
13		$r6$	$r6$	$r6$		$r6$	$r6$	

stack		input	action	output
${}_0$	:	$x-(x+x)\$$		
${}_0x_{13}$	:	$-(x+x)\$$	shift 13	
${}_0E_1$	:	$-(x+x)\$$	reduce 6	6
${}_0E_1{-}_4$	:	$(x+x)\$$	shift 4	6
${}_0E_1\,{-}_4\,({}_{10}$	:	$x+x)\$$	shift 10	6
${}_0E_1\,{-}_4\,({}_{10}x_{13}$	:	$+x)\$$	shift 13	6
${}_0E_1\,{-}_4\,({}_{10}E_5$	:	$+x)\$$	reduce 6	66
${}_0E_1\,{-}_4\,({}_{10}E_5{+}_2$	:	$x)\$$	shift 2	66
${}_0E_1\,{-}_4\,({}_{10}E_5\,{+}_2\,x_{13}$	:	$)\$$	shift 13	66
${}_0E_1\,{-}_4\,({}_{10}E_5\,{+}_2\,E_3$	:	$)\$$	reduce 6	666
${}_0E_1\,{-}_4\,({}_{10}E_5\,{+}_2\,E_3)_{11}$	:	$\$$	shift 11	666
${}_0E_1\,{-}_4\,({}_{10}E_5)_{11}$	:	$\$$	reduce 1	6661
${}_0E_1\,{-}_4\,E_5$	:	$\$$	reduce 5	66615
${}_0E_1$	:	$\$$	reduce 2	666152
${}_0E_1$	:	$\$$	halt	666152

stack		input	action	output
${}_0$	:	$-x*x\$$		
${}_0{-}_8$	:	$x*x\$$	shift 8	
${}_0\,{-}_8\,x_{13}$	:	$*x\$$	shift 13	
${}_0\,{-}_8\,E_9$	:	$*x\$$	reduce 6	6
${}_0E_1$	:	$*x\$$	reduce 4	64
${}_0E_1{*}_6$	:	$x\$$	shift 6	64
${}_0E_1\,{*}_6\,x_{13}$	:	$\$$	shift 13	64
${}_0E_1\,{*}_6\,E_7$	:	$\$$	reduce 6	646
${}_0E_1$	:	$\$$	reduce 3	6463
${}_0E_1$	:	$\$$	HALT	6463

13. Let G be the CF grammar given below, with start symbol S, which stands for **statement**. Another variable of the grammar is L, which stands for *list of statements.* Consider the LALR parser given for G.

(i) Walk through the computation of the parser if the input string is $i\{iaeaa\}$.

Note corrected production 6.

1. $S \to i_2 S_3$
2. $S \to i_2 S_3 e_4 S_5$
3. $S \to a_6$
4. $S \to \{_7 L_8\}_9$
5. $L \to L_8 S_{10}$
6. $L \to \lambda$

ACTION / GOTO

	a	i	e	$\{$	$\}$	\$	S	L
0	$s6$	$s2$		$s7$			1	
1						HALT		
2	$s6$	$s2$		$s7$			3	
3	$r1$	$r1$	$s4$	$r1$	$r1$	$r1$		
4	$s6$	$s2$		$s7$			5	
5	$r2$	$r2$	$r2$	$r2$	$r2$	$r2$		
6	$r3$	$r3$	$r3$	$r3$	$r3$	$r3$		
7	$r6$	$r6$		$r6$	$r6$			8
8	$s6$	$s2$		$s7$	$s9$		10	
9	$r4$	$r4$	$r4$	$r4$	$r4$	$r4$		
10	$r5$		$r5$	$r5$	$r5$	$r5$		

stack		input	action	output
$_0$	:	$i\{iaeaa\}\$$		
$_0 i_2$	:	$\{iaeaa\}\$$	shift 2	
$_0 i_2 \{_7$	:	$iaeaa\}\$$	shift 7	
$_0 i_2 \{_7 L_8$	:	$iaeaa\}\$$	reduce 6	6
$_0 i_2 \{_7 L_8 i_2$	:	$aeaa\}\$$	shift 2	6
$_0 i_2 \{_7 L_8 i_2 a_6$	:	$eaa\}\$$	shift 6	6
$_0 i_2 \{_7 L_8 i_2 S_3$	:	$eaa\}\$$	reduce 3	63
$_0 i_2 \{_7 L_8 i_2 S_3 e_4$	:	$aa\}\$$	shift 4	63
$_0 i_2 \{_7 L_8 i_2 S_3 e_4 a_6$	:	$a\}\$$	shift 6	63
$_0 i_2 \{_7 L_8 i_2 S_3 e_4 S_5$	:	$a\}\$$	reduce 3	633
$_0 i_2 \{_7 L_8 S_{10}$	:	$a\}\$$	reduce 2	6332
$_0 i_2 \{_7 L_8$	:	$a\}\$$	reduce 5	63325
$_0 i_2 \{_7 L_8 a_6$	:	$\}\$$	shift 6	63325
$_0 i_2 \{_7 L_8 S_{10}$	:	$\}\$$	reduce 3	633253
$_0 i_2 \{_7 L_8$	:	$\}\$$	reduce 5	6332535
$_0 i_2 \{_7 L_8 \}_9$	:	\$	shift 9	6332535
$_0 i_2 S_3$	:	\$	reduce 4	63325354
$_0 S_1$	:	\$	reduce 1	633253541
$_0 S_1$	:	\$	HALT	633253541