University of Nevada, Las Vegas Computer Science 456/656 Fall 2026

Answers to Assignment 5: Due Saturday November 7, 2026

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

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

   (b) **O** $\mathcal{P} = \mathcal{NC}$.

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

   (d) **T** Every context-free language is $\mathcal{NC}$.

   (e) **O** CVP, the circuit value probelem, is $\mathcal{NC}$.

   (f) **F** LALR parsing can be used for any context-free grammar.

   (g) **F** LALR parsing cannot be used for any ambiguous grammar.

   (h) **F** The context-free grammar equivalence problem is recursively enumerable.

   (i) **T** If L is any recursively enumerable language and $w \in L$, then there is a proof that $w \in L$.

2. Give a definition of the class $\mathcal{P}$-complete, and then name a language which is $\mathcal{P}$-complete.

   L is $\mathcal{P}$-complete if
   1. L is $\mathcal{P}$
   2. Every P language can be reduced to L using a reduction which can be computed in polylogarithmic time using polynomially many processors. Equivalently, using a reduction which can be computed using polylogarithmic space.

   Name a language which is $\mathcal{P}$-complete. There are many correct answers, but only one such language has been mentioned in class, and that is the circuit value problem.

3. If the set is countable, enter **C**, if uncountable, enter **U**.

   (a) **C** the set of all rational numbers.

   (b) **U** the set of all irrational numbers.

   (c) **C** The set of all finite sets of integers.

   (d) **U** The set of all sets of integers.

   (e) **U** The set of all finite sets of real numbers.

   (f) **U** The set of all sets of real numbers.

   (g) **U** The set of all languages over the binary alphabet.

   (h) **U** The set of all undecidable languages over the binary alphabet.

   (i) **C** the set of all recursive real numbers.

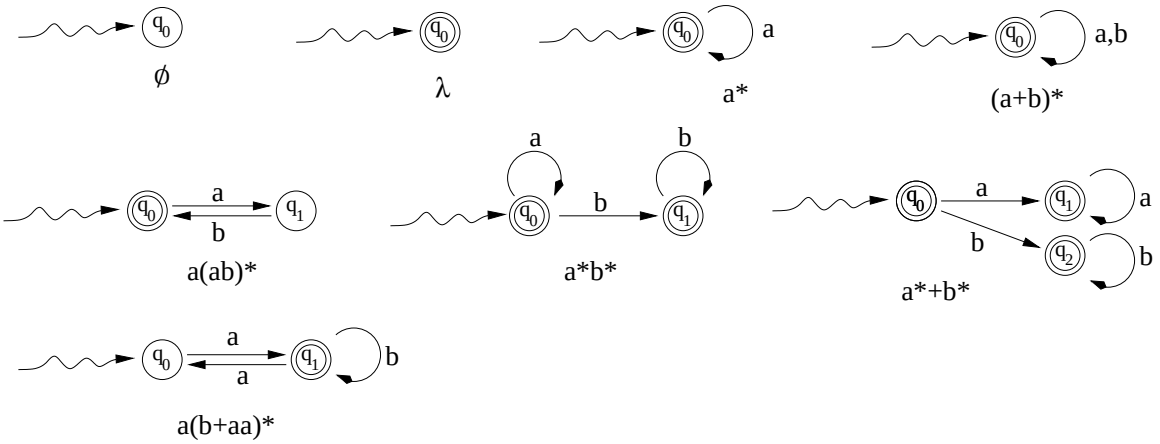
   (j) **C** The set of all recursively enumerable languages over the binary alphabet.

   (k) **C** The set of all regular expressions over the binary alphabet.

4. Let L be the language of all strings over the alphabet $\{a, b\}$ which cootain the substring aba. Write a regular expression for L.

   $(a + b)^* aba(a + b)^*$

5. For each of these regular expressions: $a(b+aa)^*$, $(ab)^*$, a^*b^*, $\emptyset$, λ, $(a+b)^*$, a^*, a^*+b^*, find a corresponding DFA in the figures below.



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

   Let $E = C_1 * C_2 * \cdots * C_k$ be a Boolean expression in 3-CNF form. Each C_i is a clause which is the disjunction of three terms, each of which is either a variable or the negation of a variable. The reduction maps E onto a graph G, which $3k$ vertices, each labeled by one of the $3k$ terms of E. Each clause C_i corresponds to the 3-clique Q_i whose vertices are labeled by the three terms of C_i. The remaining edges of G are between cliques. These edges connect any two vertices of G which have contradictory labels. We show that E is satisfiable if and only if G has an independent set of size k.

   Suppose E has a satisfying assignment. Pick one term of each clause which is true under the assignment, we call that the *satisfying* term of that clause. Let I be the set of vertices labeled by the certifying terms, k vertices altogether. Any two members of I cannot be connected by a clique edge, since they lie in different cliques. They also cannot be connected by an edge between cliques, since the labels of the ends of that edge cannot both be true under the assignment. Thus G has an independent set of size k.

   Conversely, suppose G has an independent set I of size k. There must No two of these vertices can lie in the same clique, hence there must be one in each clique. Let v_i be the sole member of $I \cap Q_i$. Now pick an assignment such that the label of each member of I is assigned true. This is possible because no two labels of members of I can contradict. If there is a variable whose value has not been assigned, assign that variable the value true. (Actually, that assignment is arbitrary.) This assignment satisfies E, since each clause has a certifying term.

7. Prove that every decidable language can be enumerated in canonical order by some machine.

Proof. Let L be any decidable language over an alphabet Σ. If L is finite, there exists a list of its members, which can be sorted and then written in canonical order. If L is infinite, let $w_1, \ldots$ be the canonical enumeration of Σ^*, which can be computed by a machine. The following program enumerates L in canonical order:

```
for i = 1 to ∞
  if(w_i ∈ L) Comment: since L is decidable, this condition can be evaluated
    write w_i
```

□

8. Prove that every recursively enumerable language is accepted by some machine.

Proof. Let L be any recursively enumerable language. If L is finite, then the result is trivial. Otherwise, let $w_1, \ldots$ be an enumeration of L computed by some machine. The following program accepts L:

```
read w
for i = 1 to ∞
  if(w_i = w)
    halt and accept.
```

□

9. The halting problem is defined as follows. Given a machine M and a string w, does M halt if its input is w? We will, later in this course, prove that the halting problem is undecidable. **However:**

   Here is a "proof" that the halting problem is decidable! Explain why this "proof" is invalid.

   Every machine is equivalent to some C++ program, and each such program has finite length. Let $\mathcal{M}$ be the set of all ordered pairs (M, w) such that M is a machine and w is a string. We define the length of such a pair (M, w) to be the length of the C++ program equivalent to M, plus the length of w. There can be only finitely many members of $\mathcal{H}$ of length n. Let $\mathcal{H}$ be the set of pairs (M, w) such that M halts with input w, a subset of $\mathcal{M}$, hence finite.

   For any n, let $T(n)$ be the maximum number of steps it takes M to halt with input w for any $(M, w) \in \mathcal{H}$.

   Given $(M, w) \in \mathcal{M}$, we can determine whether M halts with input w, the following algorithm gives us the answer.

   Let n be the length of (M, w). Run M with input w for $T(n)+1$ steps or until it halts, whichever comes first. If it halts, the answer is yes, but if it runs $T(n) + 1$ steps, the answer is no. Thus, the halting problem is decidable.

   Answer: The proof assumes that the value $T(n)$ can be computed. But there is no proof given that T is computable, and in fact, it is not.

10. Read the handouts lalrhandout1.pdf and lalrhanout2.pdf.

Consider the following annotated CF grammar G, where E is the start symbol, and the action and goto tables of an LALR parser for G.

1. $E \to E +_2 E_3$
2. $E \to E *_4 E_5$
3. $E \to (_6 E_7)_8$
4. $E \to x_9$

	x	$+$	$*$	$($	$)$	\$	
0	$s9$			$s6$			1
1		$s2$	$s4$	$s6$		HALT	
2	$s9$			$s6$			3
3		$r1$	$s4$		$r1$	$r1$	
4	$s9$			$s6$			5
5		$r2$	$r2$		$r2$	$r2$	
6	$s9$			$s6$			7
7		$s2$	$s4$		$s8$		
8		$r3$	$r3$		$r3$	$r3$	
9		$r4$	$r4$		$r4$	$r4$	

(a) There are two entries in the action table which together guarantee that multiplication has precedence over addition. There is one entry which guarantees that addition is left associative, and one which guarantees that multiplication is left associative. Identify each of those four entries, by row and column.

The rule that multiplication has precedence over addition is actually two rules. The first rule is for the expression $x * x + x$. If the top stack state is 5, and the lookahead is $+$, the action is r2 instead of s2. The second rule is for the expression $x + x * x$. If the top stack state is 3 and the lookahead symbol is $*$, the action is s4 instead of r1.

Left associativity of addition means that, in the case of $x + x + x$, the first addition is executed first. If the top stack state is 3 and the lookahead is $+$, the action is r1 instead of s2.

Left associativity of multiplication means that, in the case of $x * x * x$, the first multiplication is executed first. If the top stack state is 5 and the lookahead is $*$, the action is r2 instead of s4.

(b) Walk through a computation of the parser with the input string $x * x + x * ((x + x * x) + x)$.

stack	instream	output	action
$\$_0$	$x * x + x * ((x + x * x) + x)\$$		
$\$_0 x_9$	$* x + x * ((x + x * x) + x)\$$		$s9$
$\$_0 E_1$	$* x + x * ((x + x * x) + x)\$$	4	$r4$
$\$_0 E_1 *_4$	$x + x * ((x + x * x) + x)\$$	4	$s4$
$\$_0 E_1 *_4 x_9$	$+ x * ((x + x * x) + x)\$$	4	$s9$
$\$_0 E_1 *_4 E_5$	$+ x * ((x + x * x) + x)\$$	44	$r4$
$\$_0 E_1$	$+ x * ((x + x * x) + x)\$$	442	$r2$
$\$_0 E_1 +_2$	$x * ((x + x * x) + x)\$$	442	$s2$
$\$_0 E_1 +_2 x_9$	$*((x + x * x) + x)\$$	442	$s9$
$\$_0 E_1 +_2 E_3$	$*((x + x * x) + x)\$$	4424	$r4$
$\$_0 E_1 +_2 E_3 *_4$	$((x + x * x) + x)\$$	4424	$s4$
$\$_0 E_1 +_2 E_3 *_4 (_6$	$(x + x * x) + x)\$$	4424	$s6$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6$	$x + x * x) + x)\$$	4424	$s6$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 x_9$	$+ x * x) + x)\$$	4424	$s9$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7$	$+ x * x) + x)\$$	44244	$r2$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2$	$x * x) + x)\$$	44244	$s4$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 x_9$	$* x) + x)\$$	44244	$s9$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 E_8$	$* x) + x)\$$	442444	$r4$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 E_8 *_4$	$x) + x)\$$	442444	$s4$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 E_8 *_4 x_9$	$) + x)\$$	442444	$s9$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 E_8 *_4 E_5$	$) + x)\$$	4424444	$r4$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7 +_2 E_8$	$) + x)\$$	44244442	$r2$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7$	$) + x)\$$	442444421	$r1$
$\$_0 E_1 +_2 E_3 *_4 (_6 (_6 E_7)_8$	$+ x)\$$	442444421	$s8$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7$	$+ x)\$$	4424444213	$r3$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7 +_2$	$x)\$$	4424444213	$s2$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7 +_2 x_9$	$)\$$	4424444213	$s9$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7 +_2 E_3$	$)\$$	44244442134	$r4$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7$	$)\$$	442444421341	$r1$
$\$_0 E_1 +_2 E_3 *_4 (_6 E_7)_8$	$\$$	442444421341	$s8$
$\$_0 E_1 +_2 E_3 *_4 E_5$	$\$$	4424444213413	$r3$
$\$_0 E_1 +_2 E_3$	$\$$	44244442131342	$r2$
$\$_0 E_1$	$\$$	442444421313421	$r1$

HALT