University of Nevada, Las Vegas Computer Science 456/656 Spring 2021

Answers to Assignment 4: Due October 14 2022

1. Let L be the language of all binary numbers for non-negative integers that are equivalent to 2 modulo 3, where leading zeros are allowed. Give a regular expression for L.

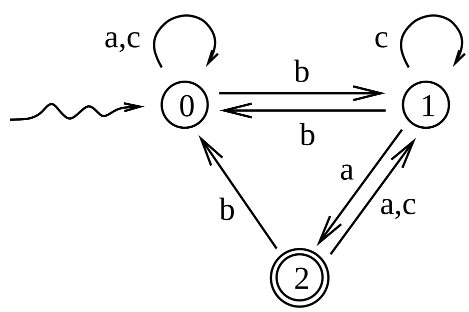
   There are infinitely many correct answers. It is not clear which the simplest is. Here is one I came up with.

   $$0^*1(10^*1)^*0(1+0(10*1)^*0)^*$$

   If I had said "equivalent to 1 mod 3" instead of equivalent to 2, the problem would have been much easier. Here is my answer to that one.

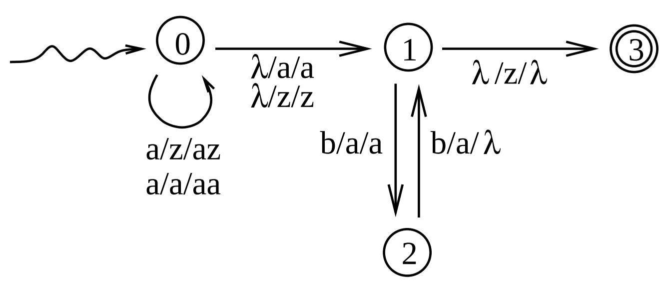
   $$0^*1(10^*1+01^*0)^*$$

2. Let L be the language accepted by the DFA shown below. Give a regular expression for L.



   This problem is much harder than I had realized, and it is nearly impossible to grade. Do not grade it.

3. Let $L = \{a^n b^{2n} : n \geq 0\}$. Draw a PDA that accepts L.



   This PDA reads and pushes a's in State 0, then non-deterministically moves to State 1 when that is finished. It then repeatedly reads b and moves to State 2, then reads one more b and moves back to State 1. When the stack is empty, it moves to State 3. With some small changes, we could draw a DPDA.

4. Correctly state (not prove) the pumping lemma for regular languages, then use the pumping lemma to prove that the language L given in Problem 3 is not regular.

   For any regular language L, there is a positive integer p such that, for any $w \in L$ of length at least p, called a pumping length of L, there exist strings x, y, z such that the following four conditions hold:
   1. $xyz = w$
   2. $|xy| \leq p$
   3. $|y| \geq 1$
   4. For any integer $i \geq 0$, $xy^iz \in L$.

Suppose $L = \left\{a^n b^{2n} \,:\, n \geq 0\right\}$ is regular. L must have a pumping length, p. Let $w = a^p b^{2p}$. There exist x, y, z such that the four conditions hold. Since $|xy| \leq p$, xy is a substring of a^p. Thus $y = a^k$ for some integer $k \geq 1$. Let $i = 0$. Then $xy^i z = xz = a^{p-k} b^{2p} \notin L$, contradiction.

5. (a) Resolve the paradox given in the handout `recenum` The definition of the function f in the handout is correct; the function f does exist.

   The resolution of the paradox is that the enumeration of L_{fnc} given in the proof exists, but cannot be recursive. That is, L_{fnc} is not RE.

   (b) Describe a specific language which is not recursively enumerable. Well, L_{fnc} is an example. Another example you've seen before is the context-free grammar equivalence problem, that is, the set of strings of the form $\langle G_1 \rangle \langle G_2 \rangle$ such that G_1 and G_2 are equivalent context-free grammars. That problem is not RE, although the proof is too long for us to do it in class.

6. Prove that the halting problem is undecidable.

One proof is in the list of handouts, in a document called halt.pdf. There are shorter proofs.

7. Consider the following CF grammar and LALR parser.

1. $S \to i_2 S_3$
2. $S \to i_2 S_3 e_4 S_5$
3. $S \to w_6 S_7$
4. $S \to a_8$

	ACTION					GOTO
	a	i	e	w	\$	S
0	$s8$	$s2$		$s6$		1
1					halt	
2	$s8$	$s2$		$s6$		3
3			$s4$		$r1$	
4	$s8$	$s2$		$s6$		5
5			$r2$		$r2$	
6	$s8$	$s2$		$s6$		7
7			$r3$		$r3$	
8			$r4$		$r4$	

Walk through the computation of this parser where the input string is *iiwaeia*.

stack		input	action	output
${}_0$	:	*iiwaeia*		
${}_0 i_2$	:	*iwaeia*	$s2$	
${}_0 i_2 i_2$	:	*waeia*\$	$s2$	
${}_0 i_2 i_2 w_6$	:	*aeia*\$	$s6$	
${}_0 i_2 i_2 w_6 a_8$	:	*eia*\$	$s8$	
${}_0 i_2 i_2 w_6 S_7$	:	*eia*\$	$r4$	4
${}_0 i_2 i_2 S_3$	:	*eia*\$	$r3$	43
${}_0 i_2 i_2 S_3 e_4$	:	*ia*\$	$s4$	43
${}_0 i_2 i_2 S_3 e_4 i_2$	:	*a*\$	$s2$	43
${}_0 i_2 i_2 S_3 e_4 i_2 a_8$	:	\$	$s8$	43
${}_0 i_2 i_2 S_3 e_4 i_2 S_7$	:	\$	$r4$	434
${}_0 i_2 i_2 S_3 e_4 S_3$	:	\$	$r1$	4341
${}_0 i_2 i_2 S_3 e_4 S_5$	:	\$	$r1$	4341
${}_0 i_2 S_3$	:	\$	$r2$	43412
${}_0 S_1$	:	\$	$r1$	434121
${}_0 S_1$	:	\$	HALT	434121