QUIZ 3: 60 Minutes

Last Name: Solutions

First Name: ______

RIN: ______

Section: ______

Answer **ALL** questions.
NO COLLABORATION or electronic devices. Any violations result in an F.
NO questions allowed during the test. Interpret and do the best you can.

GOOD LUCK!

Circle at most one answer per question.
10 points for each correct answer

You **MUST** show **CORRECT** work to get full credit.

When in doubt, TINKER.

Total
200

1. How many injective (1-to-1) functions map $\{a, b, c, d\}$ to $\{1, 2, 3\}$?

A

- [A] 0.
- [B] 36.
- [C] 42.
- [D] 81.
- [E] None of the above.

For an injection $A \to B$, $|A| \leq |B|$
here $|A| > |B|$ $\therefore$ not possible

2. How many surjective (onto) functions map $\{a, b, c, d\}$ to $\{1, 2, 3\}$?

B

No work No credit!

- [A] 0.
- [B] 36.
- [C] 42.
- [D] 81.
- [E] None of the above.

There are total 3^4 functions $(3\times3\times3\times3) = 81$ choices for a, b, c, d.
Let A_i be functions not using i, $|A_i| = 2^4 = 16$
$A_i \cap A_j = A_{ij}$ = functions not using i, j $|A_{ij}| = 1^4 = 1$
$A_i \cap A_j \cap A_k = A_{ijk}$ = functions not using $i, j, k = 0$.
By inclusion-exclusion
$|A_1 \cup A_2 \cup A_3| = 3 \cdot |A_1| - 3 \cdot |A_{12}| = 3 \cdot 16 - 3 = 45$
45 functions not surjective $\to 81 - 45 = 36$ surjections

3. An injective function f maps a set $\mathcal{A}$ to $\mathbb{N}$, $f : \mathcal{A} \mapsto \mathbb{N}$. Which is not true?

C

- [A] $\mathcal{A}$ can be finite. ✓
- [B] $\mathcal{A}$ can be infinite. ✓
- [C] $\mathcal{A}$ is a subset of $\mathbb{N}$, $\mathcal{A} \subseteq \mathbb{N}$. ✗ $\mathcal{A}$ can be any domain
- [D] $\mathcal{A}$ can be the set of all possible finite computer programs in python. ✓ countable so true.
- [E] All of the above is true.

4. A computing problem is a language. The cardinality of the set of all possible computing problems is:

D

- [A] Finite. ✗
- [B] Countable. ✗
- [C] Infinite but countable. ✗
- [D] Uncountable.
- [E] None of the above.

UNCOUNTABLE
$\updownarrow$ maps to
infinite binary strings.

5. The language $\mathcal{L} = \{0, 00, 000\} \bullet \{\varepsilon, 1, 11\}$. Which string is not in $\mathcal{L}$?

C

- [A] 0.
- [B] 011.
- [C] 100.
- [D] 001.
- [E] They are all in $\mathcal{L}$.

Start with one of these strings
$\to$ cannot start with 1.

6. For languages $\mathcal{L}_1 = \{1\}^*$ and $\mathcal{L}_2 = \{1\} \bullet \{0,1\}^*$, which is true? ($\{\}^*$ is Kleene star.)

E

- [A] $\mathcal{L}_1 \subseteq \mathcal{L}_2$. × $\varepsilon \in \mathcal{L}_1$, $\varepsilon \notin \mathcal{L}_2$
- [B] $\mathcal{L}_2 \subseteq \mathcal{L}_1$. × $10 \in \mathcal{L}_2$ $10 \notin \mathcal{L}_1$
- [C] $\mathcal{L}_1 = \mathcal{L}_2$. × requires both $\mathcal{L}_1 \subseteq \mathcal{L}_2$ and $\mathcal{L}_2 \subseteq \mathcal{L}_1$.
- [D] The regular expressions describing $\mathcal{L}_1$ and $\mathcal{L}_2$ are not valid regular expressions. × valid
- [E] None of the above are true. ✓

7. Which regular expression describes strings with at least two bits? ($\Sigma = \{0,1\}$.)

- [A] $\Sigma \bullet \Sigma$. ← 2 bits. ×
- [B] Σ^*. ← anything ×
- [C] $\Sigma^* \bullet \Sigma^*$. ← same as Σ^* ×
- [D] $(\Sigma \bullet \Sigma)^*$. ← even # bits ×
- [E] None of the above. ✓

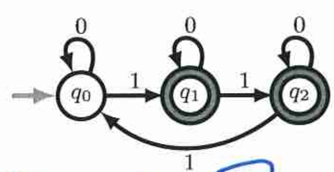
$\Sigma \cdot \Sigma \cdot \Sigma^*$ — 1-bit, 2-bit, anything

E
No work
No credit

8. What is the final resting state for the DFA with input 110010.

A

- [A] q_0.
- [B] q_1.
- [C] q_2.
- [D] This is not a valid DFA.
- [E] None of the above.



$q_0 \to q_1 \to q_2 \to q_2 \to q_2 \to q_0 \to \boxed{q_0}$

9. How many 6 bit strings are in the (YES)-set of the DFA in problem 8.

- [A] 19.
- [B] 22.
- [C] 39.
- [D] 42.
- [E] None of the above.

YES if # 1's is not a multiple of 3

$\therefore \binom{6}{1} + \binom{6}{2} + \binom{6}{4} + \binom{6}{5} = 6 + 15 + 15 + 6 = \underline{42}$.

D
No work
No credit

10. Which is the computing problem solved by the DFA in problem 8

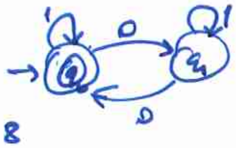
1s is not divisible by 3.

- [A] $\mathcal{L} = \{$strings with a number of 1s divisible by 3$\}$. ×
- [B] $\mathcal{L} = \{$strings with a number of 1s not divisible by 3$\}$. ✓
- [C] $\mathcal{L} = \{$strings with three more 1s than 0s.$\}$. ×
- [D] $\mathcal{L} = \{$strings with three more 0s than 1s$\}$. ×
- [E] None of the above. ×

B
No work/Explain
No credit

11. Which computing problem *cannot* be solved by a DFA (deterministic finite automata)? DFA's solve reg. exp.

$\boxed{A}$ $\mathcal{L}$ = {strings with no 1s}. 0^*

$\boxed{B}$ $\mathcal{L}$ = {strings with no 1s or an even number of 0s}. union Even 0's: → 

$\boxed{C}$ $\mathcal{L}$ = {strings have a number of 1s *not* divisible by 3}. ← prob 8

$\boxed{D}$ $\mathcal{L}$ = {strings which begin and end in different bits}. ← $\{1 * 0\} \cup \{0 * 1\}$

E Explanations needed!

$\boxed{E}$ Each problem above can be solved by a DFA. ✓

12. The main limitation of the DFA which prevents it from solving $\mathcal{L} = \{0^n 1^n | n \geq 0\}$ is:

$\boxed{A}$ The DFA is not a very fast machine so it would take too long. ✗

$\boxed{B}$ The DFA can't have more than one yes-state. ✗

$\boxed{C}$ The input string can be arbitrarily long. ✗

E

$\boxed{D}$ The DFA can go into an infinite loop. ✗

$\boxed{E}$ The DFA cannot remember how many 0s have gone by because it has only finitely many states. ✓

13. Which string *cannot* be generated by the CFG: $S \to \varepsilon \mid 0 \mid 0S$.

$\boxed{A}$ ε. no way to generate 1's.

$\boxed{B}$ 00.

$\boxed{C}$ 000.

D.

$\boxed{D}$ 0001.

$\boxed{E}$ They can all be generated.

14. Which string cannot be generated by the CGF shown?

Strings with # 1's not a multiple of 3.

$$\begin{aligned} 1 &: S \to B1A \mid B1A1B \\ 2 &: A \to \varepsilon \mid B1B1B1B \mid AA \\ 3 &: \underbrace{B \to \varepsilon \mid 0B}_{0\text{'s}} \end{aligned}$$

1's multiple of 3

$\boxed{A}$ 011101

$\boxed{B}$ 110101

$\boxed{C}$ 111100 ← # 1's is a multiple of 3.

D No work/Explanation: No credit.

$\boxed{D}$ 011100 ←

$\boxed{E}$ They can all be generated.

15. What is the difference between a Turing machine decider and a Turing machine recognizer?

$\boxed{A}$ Both are the same thing. ✗

$\boxed{B}$ A decider cannot write to the tape, a recognizer can. ✗

$\boxed{C}$ A decider can write to the tape, a recognizer cannot. ✗

E

$\boxed{D}$ A decider has a finite number of states, a recognizer can have infinitely many states. ✗

$\boxed{E}$ A decider must always halt, saying (YES) or (NO). A recognizer may not halt. ✓

16. Consider the computing problem $\mathcal{L} = \{w\#w | w \in \{0,1\}^*\}$ (# is punctuation). Which claim is true?

- [A] A DFA can solve $\mathcal{L}$.
- [B] A DFA with a top-access stack can solve $\mathcal{L}$.
- [C] A Turing machine decider can solve $\mathcal{L}$.
- [D] A Turing machine decider cannot solve $\mathcal{L}$.
- [E] None of the above.

repetition
NOT DFA
NOT CFG ⇒ DFA + stack.
TM can solve.

C

17. The theory of computing and the Church-Turing thesis define computing problems and algorithms as:

- [A] A computing problem is a string. An algorithm is a recognizer. NOPE
- [B] A computing problem is a set of finite binary strings. An algorithm is a recognizer. YES NO.
- [C] A computing problem is a Turing Machine. An algorithm is a decider.
- [D] A computing problem is a a set of finite binary strings. An algorithm is a person. YES ?
- [E] A computing problem is a a set of finite binary strings. An algorithm is a decider. ✓ ✓

E

18. The Ultimate Debugger, which we discussed in class solves, what problem?

- [A] $\mathcal{L} = \{\langle M\rangle\#w | M \text{ halts on input } w\}$. ✓
- [B] $\mathcal{L} = \{\langle M\rangle\#w | M \text{ does not halt on input } w\}$. ✓
- [C] $\mathcal{L} = \{\langle M\rangle | M \text{ halts and says yes on some input}\}$. ✗
- [D] $\mathcal{L} = \{\langle M\rangle | M \text{ halts and says no on some input}\}$. ✗
- [E] None of the above.

These are equivalent problems solved by U-D.

A or B accepted.

19. Any decider for problem $\mathcal{L}_A$ can be used as a subroutine to solve problem $\mathcal{L}_B$. Which conclusion is not true?

- [A] $\mathcal{L}_A$ is decidable. We conclude $\mathcal{L}_B$ must be decidable. ← by definition So $\mathcal{L}_B < \mathcal{L}_A$
- [B] $\mathcal{L}_A$ is undecidable. We conclude $\mathcal{L}_B$ could be decidable. $\mathcal{L}_B < \mathcal{L}_A$ so $\mathcal{L}_B$ could be decidable (easier)
- [C] $\mathcal{L}_B$ is decidable. We conclude $\mathcal{L}_A$ could be undecidable. $\mathcal{L}_B < \mathcal{L}_A$ so $\mathcal{L}_A$ could be undecidable (harder)
- [D] $\mathcal{L}_B$ is undecidable. We conclude $\mathcal{L}_A$ must be undecidable. $\mathcal{L}_B < \mathcal{L}_A$ ∴ $\mathcal{L}_A$ must be undecidable.
- [E] All of the above are true.

ALL TRUE

E

Explanations Needed!

20. Let $\mathcal{M}$ be the set of all possible Turing Machines. Which statement is not true?

- [A] Every Turing Machine in $\mathcal{M}$ can be uniquely encoded into a finite binary string. ✓
- [B] All Turing Machines in $\mathcal{M}$ can be listed: $\{\langle M_1\rangle, \langle M_2\rangle, \langle M_3\rangle, \langle M_4\rangle, \ldots, \}$. ✓ from A, countable ∴ listable ∴ ✓
- [C] $\mathcal{M}$ is countable. ✓
- [D] There is an injection from $\mathcal{M}$ to $\mathbb{N}$. Countable means injection from $\mathcal{M} \to \mathbb{N}$.
- [E] All of the above are true.

ALL TRUE

E