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

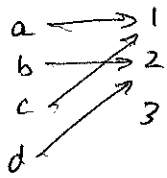
You **MUST** show **CORRECT** work to get full credit.

When in doubt, TINKER.

Total
200

1. A function f maps $\{a, b, c, d\}$ to $\{1, 2, 3\}$ as follows: $f(a) = 1, f(b) = 2, f(c) = 1, f(d) = 2$.

- [A] f is injective (1-to-1) but not bijective.
- [B] f is surjective (onto) but not bijective.
- [C] f is bijective (1-to-1 and onto).
- [D] f is neither injective nor surjective.
- [E] f is not a valid function.



not 1-1 $(a, c \to 1)$
not onto (nothing maps to 3)

2. A set $\mathcal{S}$ contains all the distinct functions which map $\mathbb{N}$ to $\{0\}$. What is the cardinality (size) of $\mathcal{S}$?

- [A] 0.
- [B] 1.
- [C] Bigger than 1 but finite.
- [D] The same as $|\mathbb{N}|$.
- [E] Strictly larger than $|\mathbb{N}|$.

$f(x) = 0 \quad \forall x \in \mathbb{N} \leftarrow$ 1 function

3. A set $\mathcal{S}$ contains all the distinct functions which map $\mathbb{N}$ to $\{2, 3, 4\}$. What is the cardinality (size) of $\mathcal{S}$?

- [A] 0.
- [B] 1.
- [C] Bigger than 1 but finite.
- [D] The same as $|\mathbb{N}|$.
- [E] Larger than $|\mathbb{N}|$.

$f: \mathbb{N} \to \{0,1\}$ is infinite binary strings $\leftarrow$ uncountable.

$\therefore$ $f: \mathbb{N} \to \{2,3,4\}$ is larger

4. What is the cardinality (size) of the set containing all distinct python programs of finite length?

- [A] 0.
- [B] 1.
- [C] Bigger than 1 but finite.
- [D] The same as $|\mathbb{N}|$.
- [E] Larger than $|\mathbb{N}|$.

program $\equiv$ finite binary string
$\uparrow$
countable.

5. Which set is *not* countable, i.e., has a cardinality strictly larger than $|\mathbb{N}|$?

- [A] $\mathbb{Q}$, the rational numbers. $\leftarrow$ countable $\mathbb{Z} \times \mathbb{N}$
- [B] All distinct finite binary strings. $\leftarrow$ countable.
- [C] The set of all possible Turing Machines. $\leftarrow$ finite strings — countable.
- [D] The set containing all distinct functions that map $\{0, 1\}$ to $\mathbb{N}$. $\leftarrow$ $\mathbb{N} \times \mathbb{N}$ — countable
- [E] They are all countable.

6. What is a computing problem?

A A person who knows how to write a program in python.

B A machine that transitions between states.

C A rule for deciding if a string belongs to a set.

D Any set of finite binary strings.

E A Turing Machine.

7. $\mathcal{L}$ is a computing problem. What can we say about the cardinality (size) of $\mathcal{L}$?

A $\mathcal{L}$ must have finite cardinality. ✗ ← can be infinite

B $\mathcal{L}$ must have infinite cardinality. ✗ ← can be finite. $\mathcal{L} \subseteq \Sigma^*$

C $\mathcal{L}$ must be countable. ↑ countable ↑ countable

D $\mathcal{L}$ must be uncountable. ← not possible

E None of the above.

8. $\mathcal{L}_1$ and $\mathcal{L}_2$ are computing problems. Which of the following is *not* a computing problem?

A $\mathcal{L}_1 \bullet \mathcal{L}_2$.

B $\mathcal{L}_1^*$.

C $\mathcal{L}_1 \cap \mathcal{L}_2^*$.

D $\mathcal{L}_1^* \cup \overline{\mathcal{L}_2^*}$.

} regular operations. give new languages.

E They are all computing problems

9. Which of the following strings is in the language described by the regular expression $\{0, 11\}^*$?

A 011111. ✗

B 010010. ✗

C 100100. ✗

D 110011. ← 11 • 0 • 0 • 11 ✓

E None of the strings above are in the language.

even # of 1's in any substring of 1's.

10. Which computing problem *cannot* be solved by a DFA (deterministic finite automata)?

A $\mathcal{L} = \{\text{strings with at least one 1}\}$. ← $*1$

B $\mathcal{L} = \{(01)^{\bullet n} \mid n \geq 0\}$. ← $\{01\}^*$

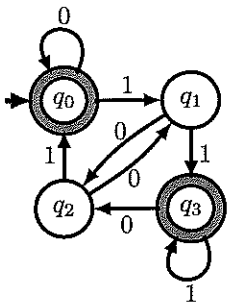
C $\mathcal{L} = \{\text{strings that end with 101}\}$. ← $*101$

D $\mathcal{L} = \{1^{\bullet n} w \mid n \geq 1 \text{ and } w \text{ has } n \text{ or more 1s}\}$. ← $1*1*$

} regular expressions → DFA can solve.

E Each problem can be solved by a DFA.

11. For the DFA on the right, what sequence of states are visited for the input 0110.



A $q_2\ q_0\ q_1\ q_3\ q_2$.

B $q_0\ q_0\ q_0\ q_0\ q_0$.

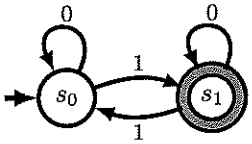
C $q_0\ q_1\ q_3\ q_2$.

D $q_0\ q_0\ q_1\ q_2\ q_3$.

E $q_0\ q_0\ q_1\ q_3\ q_2$.

$q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_3 \xrightarrow{0} q_2$

12. How many 4-bit strings are in the (YES)-set of the DFA on the right. Accept states are double circles (only s_1 in this case).



A 6.

B 7.

C 8.

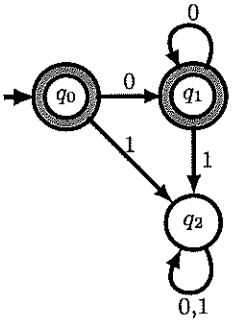
D 9.

E 10.

Accepts strings with an odd # of 1s
∴ 1 or 3

$\binom{4}{1} + \binom{4}{3} = 4+4 = 8$

13. Which is *not* a description of the computing problem solved by the DFA on the right? Accept states are double circles (q_0, q_1 in this case).



A $\{0\}^*$.

B $\{0^{\bullet n} | n \geq 0\}$.

C All strings with no ones, including the empty string ε.

D Strings generated by the CFG: $S \to \varepsilon | 0S$.

E They all describe the computing problem solved by the DFA.

All these are the same and work.

14. Which string *cannot* be generated by the CFG: $S \to 0|1|0S0|1S1$.

A 01110 ✓

B 10001 ✓

C 0110 ✗

D 11111 ✓

E They can all be generated.

0, 1, 0S0 → 000, 010

00S00, 01S10

each time add 2 characters

∴ Palindromes of odd length!

15. Describe the strings generated by the CFG on the right.

$1: S \to A|B$
$2: A \to 0|0A$
$3: B \to 1|1A$

A Nonempty strings containing only 0s.

B Nonempty strings containing only 1s.

C All nonempty strings.

D Nonempty strings which contain either only 0s or only 1s.

E None of the above

$S \to A \to 0^n$

$S \to B \to 1A \to 1 0^n$

$\therefore \{1, 0^n, 10^n\}$

16. Which is a true statement about the computing problem $\mathcal{L} = \{w\#w | w \in \{0,1\}^*\}$.

A A DFA can solve $\mathcal{L}$. A Turing Machine can solve $\mathcal{L}$. ↑ TM in text for this.

B A DFA cannot solve $\mathcal{L}$. A Turing Machine cannot solve $\mathcal{L}$.

C A DFA can solve $\mathcal{L}$. A Turing Machine cannot solve $\mathcal{L}$.

D A DFA cannot solve $\mathcal{L}$. A Turing Machine can solve $\mathcal{L}$.

E None of the above.

17. How do we know there are computing problems which Turing Machines cannot solve?

A Because the Turing Machines are countable and the computing problems are countable.

B Because the Turing Machines are uncountable and the computing problems are countable.

C Because the Turing Machines are countable and the computing problems are uncountable. more problems than TM's.

D Because the Turing Machines are uncountable and the computing problems are uncountable.

E None of the above proves there are computing problems which Turing Machines cannot solve.

18. What is the difference between a decider D for a language $\mathcal{L}$ and a recognizer R for $\mathcal{L}$.

A For $w \in \mathcal{L}$, D halts with (YES) but R may infinite loop. R must also halt with (YES)

B For $w \in \mathcal{L}$, D halts with (YES) and R halts but may say (NO). R must say (YES)

C For $w \notin \mathcal{L}$, D halts with (NO) but R may sometimes go into an infinite loop.

D For $w \notin \mathcal{L}$, D halts with (NO) but R must go into an infinite loop. ← may not infinite loop

E For $w \notin \mathcal{L}$, D halts with (NO) and R halts but may say (YES). ← if halts, R must say (No)

19. What is the Ultimate Debugger which we discussed in class?

A A program that solves Goldbach's conjecture.

B A program that solves the twin-prime conjecture.

C A program that determines if another program will compile under a C++ compiler. ← parser

D A program that translates another program into binary machine-code. ← compiler

E A program that determines if another program when run will halt.

20. Which answer is a valid conclusion about the decidability of the language $\mathcal{L}_B$?

A $\mathcal{L}_A$ is decidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is decidable. $\mathcal{L}_B$ harder than $\mathcal{L}_A$

B $\mathcal{L}_A$ is decidable. A decider for $\mathcal{L}_A$ can be converted to a decider for $\mathcal{L}_B$. So, $\mathcal{L}_B$ is undecidable. $\mathcal{L}_A$ harder than $\mathcal{L}_B$ but $\mathcal{L}_B$ must be decidable

C $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_A$ can be converted to a decider for $\mathcal{L}_B$. So, $\mathcal{L}_B$ is undecidable. $\mathcal{L}_A$ harder than $\mathcal{L}_B$ so $\mathcal{L}_B$ may be decidable

D $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is decidable. $\mathcal{L}_B$ is harder than $\mathcal{L}_A$ ⇒ $\mathcal{L}_B$ is undecidable

E $\mathcal{L}_A$ is undecidable. A decider for $\mathcal{L}_B$ can be converted to a decider for $\mathcal{L}_A$. So, $\mathcal{L}_B$ is undecidable. $\mathcal{L}_B$ is harder than $\mathcal{L}_A$ → undecidable