FOCS Quiz 3
Wang Xinshi

(1). B

There exists a constant $c \in \mathbb{N}$ such that $1 + \frac{1}{\sqrt{2}} + ... + \frac{1}{\sqrt{n}} > c\sqrt{n}$. Thus the c is in the outer parenthese with an there exists statement.

(2). E

There are 3^4 ways of mapping range to domain. We want to exclude situation where domain does not map to all elements in range. Case one would be only 2 elements in Range is used. Thus there are $\binom{3}{2} \times 2^4$ ways of mapping 2 elements in the range into domain. Case two would be 1 element in the Domain is used. There are $\binom{3}{1} \times 1^4$(since there are only 2 available). Thus in total we have 3^4 - $\binom{3}{2} \times 2^4$(since there are only 2 available) - $\binom{3}{1} \times 2^4 = 81 - 3 \times 16 - 6 = 27$

(3). C

An example given in class is $0 \cup \mathbb{N}$. You can map this element $n \in 0 \cup \mathbb{N}$ into $n + 1 \in \mathbb{N}$.

(4). D

It is uncountable as we discussed in class. You can prove it use canotr-diagonalization as we did in class. Therefore it is uncountable.

(5). C

The string must start with 0. Therefore C is not in $\mathcal{L}$.

(6). E

$\mathcal{L}_1 = \{1, 11, 111, 1111...\}$. $\mathcal{L}_2 = \{1, 10, 101, ...\}$. As we can see, there is no overlap between those two sets.

(7). E

In order for the Regex to contain at least 2 bits, it cannot contain a $*$. Therefore the only option left is A and E. A contains exactly two bits, thus choose E since we need at least 2 bits.

(8). A

We move step by step, and end up with q_0.

(9). D

There are in total 2^6 ways of forming a binary string. We discover 1 one, 2 ones, 4 ones, and 5 ones. Therefore the cardinality of yes-set = 2^6 - $\binom{6}{1}$ - $\binom{6}{3}$ - $\binom{6}{6} = 64 - 1 - \frac{6}{3 \times 2 \times 1} - 1 = 64 - 2 - 20 = 42$.

(10). B

As we found in 9, only 1 one, 2 ones, 4 ones, and 5 ones.Thus the turing machine does not work at 3 and 6, which is divisible by 3.

(11). E

None of these problems require meomory, therefore a DFA could solve all of them.

(12). E

This is the exact same thing we discussed in Class. DFA could not memorize its previous states.

(13). D

There is no 1 in the prodiction rules. Thus there could not be a 1 in the Language. Thus D is wrong.

(14). D

If S choose $B1A$ in the first place, there will be at least one 1 or four 1s consecutively. If S choose $B1A1B$ in the first place, there will be at least two 1s or five 1s consecutively. Therefore the case of having three consecutively and thus D is wrong.

(15). E

This is the definiton of deciders and recognizers. A decider must always halt and say yes or no. A recognizer might trap into an infinite loop.

(16). C
This is the first Turing Machine we met in the first TM class. We even built it. Thus a turing machine could solve the problem. A DFA could not solve it since it has no memory.

(17). E
An algorithm is a Turing machine that always halts, which in this case is a decider. A computing problem in essence is a set of finite binary strings as the definition says.

(18). A
The ultimate debugger should tell us whether a program will end or not. Thus M must halt.

(19). E
Since $\mathcal{L}_A$ decides $\mathcal{L}_B$, we know $\mathcal{L}_A$ is harder than $\mathcal{L}_B$. If $\mathcal{L}_A$ is decidable, $\mathcal{L}_B$ must be decidable since $\mathcal{L}_A$ is harder. If $\mathcal{L}_A$ is undecidable, $\mathcal{L}_B$ could be decidable since $\mathcal{L}_A$ is harder. If $\mathcal{L}_B$ is decidable, $\mathcal{L}_A$ may be undecidable since $\mathcal{L}_A$ is harder. If $\mathcal{L}_B$ is undecidable, $\mathcal{L}_A$ must be undecidable since $\mathcal{L}_A$ is harder.

(20). E
A. It is true. A turing machine can be encoded to a finite binary string, like a yes set.
B. It is obvious that you can find a turing machine and list it. You can definitely list them.
C. It is finite since you can list all of turing machines. D. Since it is finte, the cardinality is smaller than $|\mathbb{N}|$.