TCS_Quiz 1_SH-20[co1]

Total points 8/10

Email address *

ameythakur@ternaengg.ac.in

Name of student *

AMEY THAKUR

Class *

TE-B

Roll No *

50

✓ 1]The concept of FSA is much used in this part of the compiler * 1/1

- (●) lexical analysis ✓
- ( ) parser
- ( ) code generation
- ( ) code optimization

✓ 2]Palindromes can't be recognized by any FSA because * 1/1

- FSA cannot remember arbitrarily large amount of information
- FSA cannot deterministically fix the midpoint
- Even if the mid point is known an FSA cannot find whether the second half of the string matches the first half
- **all of the above** ✓

✓ 3]A language L is accepted by a FSA iff it is * 1/1

- .CFL
- CSL
- recursive
- **regular** ✓

✓ 4]R1 and R2 are regular sets. Which of the following is not true? * 1/1

- **R1 ∩ R2 neet not be regular** ✓
- Σ* − R1 is regular
- R1 ∪ R2 is regular
- is regular

✗ 5]An FSM with * 0/1

- stack is more powerful than an FSM with no stack
- stacks is more powerful than a FSM with 1 stack
- both (a) and (b) ✗
- None of these

✗ 6]If two finite state machines are equivalent, they should have the same number of * 0/1

- states
- edges
- states and edges ✗
- none of these

✓ 7] L = {aP | p > 1 ; } is prime is * 1/1

- regular
- not regular ✓
- accepted by DFA
- accepted by PDA

✓ 8]Which of the following are not regular ? * 1/1

- ○ String of 0's whose length is a perfect square
- ○ Set of all palindromes made up of 0's and 1's
- ○ Strings of 0's, whose length is a prime number
- ◉ All of these ✓

✓ 9]The main difference between a DFSA and an NDFSA is * 1/1

- ○ in DFSA, ε transition may be present
- ○ in NDFSA, ε transitions may be present
- ◉ in DFSA, from any given state, there can't be any alphabet leading to two diferent states ✓
- ○ in NDFSA, from any given state, there can't be any alphabet leading to two diferent states

✓ 10] The major difference between a moore and mealy machine is that * 1/1

- ○ output of the former depends on the present state and present input
- ◉ output of the former depends only on the present state ✓
- ○ output of former depends only on the present input
- ○ all of these