--Question Starting--
3. Given the finite automaton (FA) recognizing a regular language L, which of the following statements correctly describes the relationship between the DFA and NDFA representations of L?
Statement I: Every DFA can be converted into an equivalent NDFA without changing the language recognized, but the reverse conversion may result in an exponential increase in the number of states.
Statement II: The process of converting an NDFA to an equivalent DFA involves the subset construction algorithm, which systematically explores all possible state subsets to ensure determinism.
In light of the above statements, select the most accurate option:
(1) Both Statement I and Statement II are correct
(2) Both Statement I and Statement II are incorrect
(3) Statement I is correct but Statement II is incorrect
(4) Statement I is incorrect but Statement II is correct
Answer Key: 2
Solution:
? Statement I(Correct): All DFAs are inherently NDFAs (since DFAs are a special case). Converting a DFA to NDFA is straightforward and does not change the language. However, converting NDFA to DFA via subset construction can lead to an exponential increase in states due to the power set of the NDFA's state set.
? Statement II(Correct): The subset construction algorithm involves creating states in the DFA corresponding to subsets of NDFA states, ensuring the resulting automaton is deterministic and recognizes the same language.
Thus, the incorrect part is the suggestion that the reverse conversion does not cause exponential blow-up?it's a well-known worst-case exponential increase.
Hence, Option (2) is the right answer.

--Question Starting--
4. Consider the following statements about system software:
Statement I: During compilation, the process of linking can be performed either statically or dynamically, where static linking embeds all code into a single executable, whereas dynamic linking defers some code loading until runtime, facilitating shared libraries.
Statement II: A macro processor operates before compilation, performing text substitution, while a debugger operates after the program has been compiled and linked, allowing step-by-step execution and variable inspection.
Identify the correct option:
(1) Both Statement I and Statement II are correct
(2) Both Statement I and Statement II are incorrect
(3) Statement I is correct but Statement II is incorrect
(4) Statement I is incorrect but Statement II is correct
Answer Key: 4
Solution:
? Statement I(Correct): Static linking incorporates all necessary code at compile time, resulting in a standalone executable, while dynamic linking postpones some linking to runtime, enabling shared libraries and memory efficiency.
? Statement II(Incorrect): Macros are processed during preprocessing, before compilation, whereas debuggers operate post-compilation, during execution. Debuggers do not perform text substitution but allow inspection of the already compiled binary.
Therefore, Statement I is correct and Statement II is incorrect.
Hence, Option (4) is the right answer.