Hmm... automata theory

In this discussion, DFA = (Q, Σ, δ, q_0, F) and related terms are defined as per Sudkamp [S]. System definitions, especially that of *state* and *state system* are taken from [P].

1. Is a string a tape? No
2. Is a tape a string? No

   A string is a finite sequence of symbols. It is a value.
   A tape is an infinite sequence of cells.

3. Is a language a set of strings? Yes, by definition.

4. Is a language a set of tapes? No

5. Is a string a system? No

   As usually taken, a string has no system boundary (a set of variables that take time-function values).

6. Is a tape a system? Yes

   Each cell of a tape is a read/write buffer (which is a system; i.e., it has a system boundary). A tape is a sequence of cells and has a system boundary consisting of the sequence of the cell system boundaries.

7. Is a DFA (deterministic finite automaton) a system? No

   A DFA has no explicit system boundary.

8. Does a DFA accept or reject individual strings? No

   A DFA has no explicit output variable; thus, as defined, it cannot explicitly assert an acceptance or rejection action. If the DFA's F parameter is interpreted to imply an output variable, this variable would seem to be 2-valued; BUT, the DFA output must be at least 3-valued (corresponding to assertions of busy, accept, and reject).

9. Does a DFA accept or reject a set of strings? No

10. Does a DFA accept or reject individual tapes? No

    The input to a DFA is a string. So, if a DFA accepts anything, it would be a string.

11. Does a DFA accept or reject a set of tapes? No

12. Does a DFA define a language? Yes

    In the same sense that a road map defines the set of all routes from Kalamazoo to Lake Michigan, a DFA defines the set of all strings from q_0 to F. But in no sense does a road map actively answer if any proposed route is acceptable in taking one from Kalamazoo to Lake Michigan; so, no DFA actively answers if a string takes one from q_0 to F.

13. Does a DFA-based string-accepting system have a finite state set? No

    Among other things, the DFA tape (which is explicitly internal to the DFA) has an infinite set of cells, each of which can store any value from $\{B\}\cup\Sigma$. Consequently, the state space of the tape is infinite, and so is the state space of the DFA-based string-accepting system. The position of the tape head, another component of the DFA-based string-accepting system, is also infinite; further, inferring from the multiple explanations in [S], this variable is non-deterministic (e.g., the tape might be stationary with the head moving relative to it, or the head might be stationary with the tape moving relative to it, etc.).

Concluding remarks

- A D(eterministic) F(inite) A(utomaton) is neither deterministic nor finite nor an automaton.
- A DFA is not reusable (in that the DFA has no defined state initialization transitions in δ that would be necessarily executed at the moment of each new string input).
- Traditional automata theory unnecessarily fails to conform in terminology and concept to that of system theory. This makes learning and teaching automata theory harder and closes off opportunities of practical application, such as in
  - formal computer system specification and design [especially real-time systems],
  - software engineering tool development,
  - formal-based testing, and
  - end-user education.

[P] T.F. Piatkowski, *Foundations of State-System Specification: Volume 1 -- Introduction*, 2006.

[S] T.A. Sudkamp, *Languages and Machines: An Introduction to the Theory of Computer Science* (2e), Addison-Wesley, Reading MA, 1997.