Prep Work 11 - Turing Machines

CS 234

Daniel Lee

1. Turing Machines

1. What does a Turing machine have that a DFA does not?

A Turing machine has an infinitely long tape for having an unbounded memory.

2. By convention, we don't write all the transitions that a Turing machine could take, even though it is deterministic. What does the Turing machine do if it encounters a transition that we did not write down?

If there is no rule for a given combination during the computation, then the machine simply halts and rejects.

3. Describe the parts of the 7-tuple defining a Turing machine.

Q: The set of states.
Σ: The input alphabet.
Γ: The tape alphabet ($\Sigma \subseteq \Gamma$ and $B \in \Gamma$).
δ: The transition function $(Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{L, R\})$.
q_0: The start state.
B: The special blank symbol.
F: The set of accepting states.

4. Look at Figure 13.1's Turing machine. Give the trace for the Turing machine on input 01. (See Examples 13.2-4 for examples of how a trace can be written.)

q0_01 ⇒ Bq1_1 ⇒ B1q1_B ⇒ Bq2_1B ⇒ q3_BB ⇒ Bq0_B ⇒ BBq4_B (ACCEPT)

5. In your own words, why can every regular language be accepted by a Turing machine? (You do not need to prove this, just give intuition.)

By the definition of regularity, a regular language is one that has a DFA, an NFA, or a regular expression. Since the Turing machine accepts the language of the DFA, we know that the regular language also has a Turing machine that accepts it. Thus, every regular language can be accepted by a Turing machine.

6. In your own words, describe how a Turing machine can be used to represent a function.

A Turing machine can be used to represent a function by regarding the string on the tape as its input and regarding the modification made by writing while traversing the tape as its output.