Turing Machines – Lecture 13
James Marshall

The Turing Machine

Like a finite automaton but:

1. **1.** A TM can read from *and* write to the input tape
2. **2.** The read-write head can move right *or* left
3. **3.** The tape is infinite
4. **4.** Accept and reject states take effect immediately

Definition

A TM is a 7-tuple $(Q,\Sigma,\Gamma,\delta,q_0,q_{accept},q_{reject})$ where Q,Σ,Γ are finite sets and

1. Q is the set of *states*
2. Σ is the *input alphabet*, which must exclude the blank symbol
3. Γ is the *tape alphabet*, where
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ is the transition function
5. $q_0 \in Q$ is the start state
6. $q_{accept} \in Q$ is the accept state
7. $q_{reject} \in Q$ is the reject state, where $q_{reject} \neq q_{accept}$

Definition

A language is *Turing-recognisable* if some TM recognises it, defined as reaching q_{accept} if the input string is a member of the language.

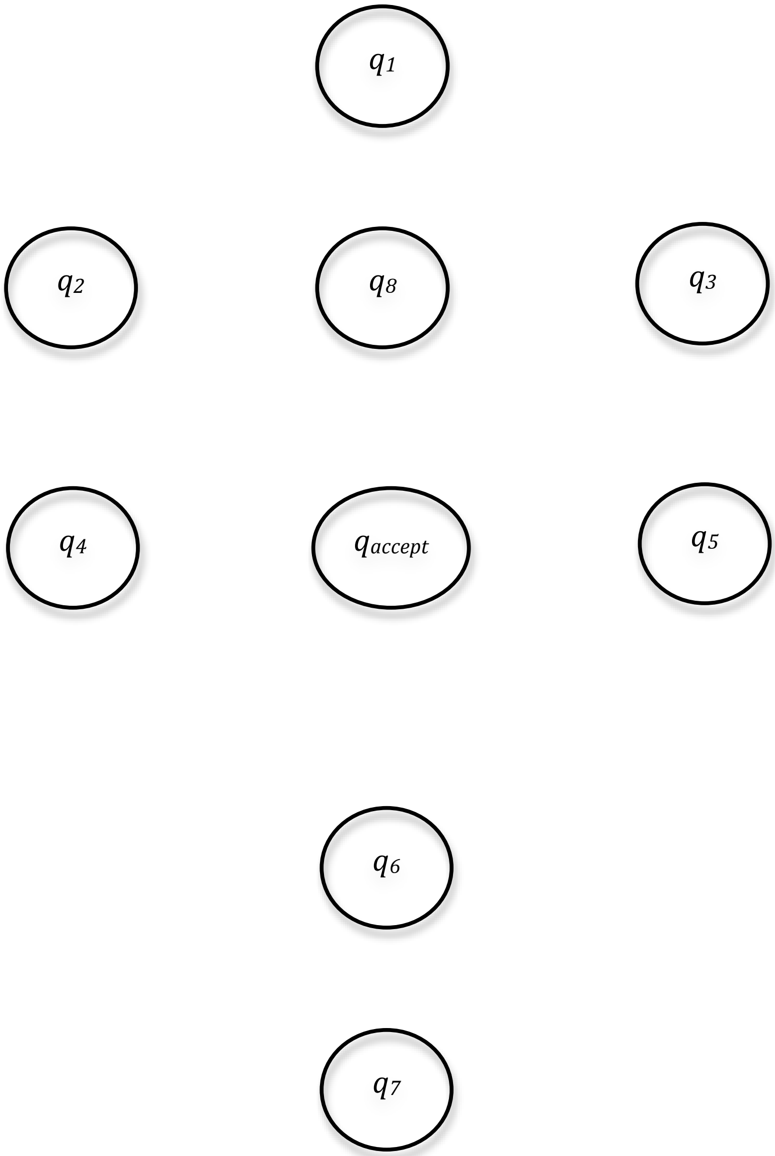
Definition

A language is *Turing-decidable* if some TM decides it, defined as reaching q_{accept} if the input string is a member of the language, otherwise reaching q_{reject}

Example

Design a TM to decide $B = \left\{ w\#w \mid w \in \{0,1\}^* \right\}$

- $\Sigma =$
- $\Gamma =$



Church-Turing Thesis

"*Every effectively calculable function is a computable function*"