Logic Thinking

Turing Machines: 7-tuple definition

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

Outline

- Foundation of logic
  - Propositional Logic
  - Predicative Logic
- Automata and Turing Machines
- Power and Limitation of Computing
  - Mechanical Theorem Proving
  - Church-Turing Hypothesis

Turing Machine

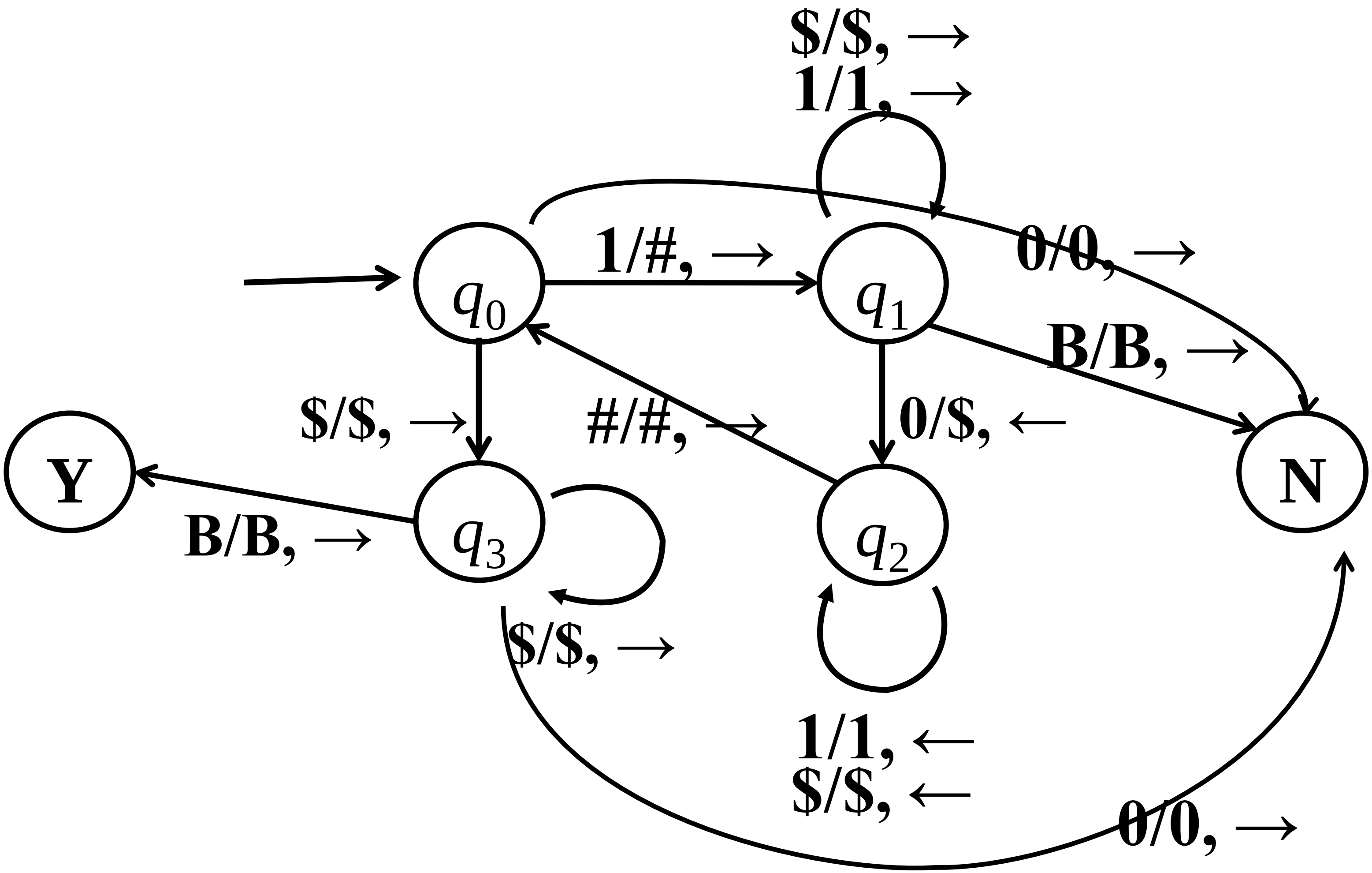
- Turing machine: a 7-tuple $\{Q, \Sigma, \Gamma, \delta, q_0, q_{\text{Accept}}, q_{\text{Reject}}\}$
  - Q: set of states
  - Σ: set of input symbols
  - Γ: set of tape symbols.
    - Special character $B \in \Gamma$
  - δ: transition function

    $$(Q - \{q_{\text{Accept}}, q_{\text{Reject}}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{\rightarrow, \leftarrow\}$$
  - $q_0 \in Q$: the initial state
  - $q_{\text{Accept}} \in Q$: the accept state
  - $q_{\text{Reject}} \in Q$: the reject state

Turing machine example

- Input: a binary string $1^n 0^m$, 111…11000…00
- Goal: decide whether the number of 1s is the same as the number of 0s, that is, whether $m = n$ or not.

State-transition diagram



State-transition function

- $(q_0,1)\rightarrow(q_1,\#, R)$, $(q_0,\$)\rightarrow(q_3,\$, R)$
- $(q_1,1)\rightarrow(q_1,1, R)$, $(q_1,\$)\rightarrow(q_1,\$, R)$, $(q_1,0)\rightarrow (q_2,\$, L)$, $(q_1,B)\rightarrow (q_{reject},B, R)$
- $(q_2,1)\rightarrow(q_2,1, L)$, $(q_2,\$)\rightarrow(q_2,\$, L)$, $(q_2,\#)\rightarrow (q_0,\#, R)$
- $(q_3,\$)\rightarrow(q_2,\$, R)$, $(q_3,0)\rightarrow(q_{reject},0, R)$, $(q_3,B)\rightarrow(q_{accept},B, R)$

In-class exercise

- Draw a complete state-transition table for the Turing machine we just discussed

Thinking problem

- The previous Turing machine needs about $2n^2$ steps
- Is it possible to use fewer steps?

In-lab exercise

- Palindromes can be recognized by a Turing machine

Key points of Turing machines

- Key point 1: a mathematical model of computing
  - Not a real model of computing
    - E.g., fetching a word from a 16-GB memory on a laptop computer takes just one instruction, but needs a lot of operations on a Turing machine
- Key point 2: the state-transition function
  - Needs clever design
- Key point 3: the set of states is a finite set
  - The size does not depend on the size of input

Notable details of Turing machines

- $\boldsymbol{B} \notin \boldsymbol{\Sigma}$ **but** $\boldsymbol{B} \in \boldsymbol{\Gamma}$
  - The input blank symbol *B* belongs to the tape alphabet but does not belong to the input alphabet
  - The tape alphabet contains all input symbols *and* the blank symbol
  - Don't confuse the blank symbol B with the capital letter B (0x42)
  - When the input string needs to contain B (0x42), change the blank symbol to β
- **The palindromes recognition TM example**
  - The input alphabet: $\boldsymbol{\Sigma} = \{\mathbf{0}, \mathbf{1}\}$
  - The tape alphabet: $\boldsymbol{\Gamma} = \{\mathbf{0}, \mathbf{1}, \boldsymbol{B}\}$

Notable details of Turing machines

- **No stop in the middle**
  - The Turing machine stops (halts) only when it enters a final state, either q_{Accept} or q_{Reject}
  - The transition function δ is a mathematical *function*, which means that δ is defined for every element of its domain
    - That is, for every non-final state *s* and tape symbol *t*, $\delta(s,t)$ is always defined, and there is always a next state to transition to

Notable details of Turing machines

- **There are $(|\boldsymbol{Q}| - \mathbf{2}) \times |\boldsymbol{\Gamma}|$ transitions**
  where $|Q|$ is number of elements of set *Q*
  - The value 2 is for the two final states q_{Accept} and q_{Reject}
  - the calculation only works if both accept state and reject state exist in the Turing machine
- **The palindromes recognition TM example**
  - Example 27 in Textbook
  - There are 15 transitions
    - Number of non-final states $\times$ number of tape symbols
    - $Q = \{q_0, q_{\text{Accept}}, q_{\text{Reject}}; q_{\text{Seen0}}, q_{\text{Seen1}}, q_{\text{Want0}}, q_{\text{Want1}}, q_{\text{Back}}, q_{\text{BackErase}}\}$
      - Number of non-final states = $|Q| - 2 = 5$
    - $\Gamma = \{0, 1, B\}$, number of tape symbols = 3
    - $(|Q| - 2) \times |\Gamma| = 5 \times 3 = 15$

Notable details of Turing machines

- **Explicit versus implicit input/output**
  - In the initial configuration, the input string must explicitly appear in the tape between two blanks
  - The output is often defined as the string between the head-pointed square and the first blank right of it
  - Sometimes, we more carefully and explicitly define the output, e.g., in Example 27
- **The palindromes recognition TM example**
  - Both input and output are explicitly shown on tape
    - Input string 01 is not a palindrome; output is 0
      - Initial tape configuration: …B01B…; final tape configuration: …B0BB…
    - Input string 101 is a palindrome; output is 1
      - Initial tape configuration: ...B101B…; final tape configuration: …BB1BB…