Turing Machines and Equivalent Models

Meenakshi D'Souza

International Institute of Information Technology
Bangalore

Term II 2022-23

Outline

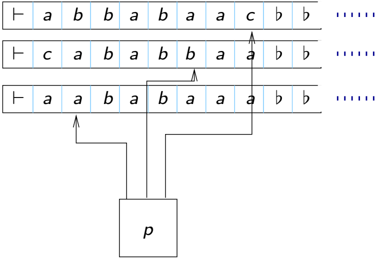
Models equivalent to a Turing Machine

Many different variants of a Turing machine and other models are all *equivalent* to a Turing machine.

- Variants of a Turing machine:
  - Multi-tape Turing machine.
  - Two-way infinite tape Turing machine.
  - Non-deterministic Turing machine.
- Other models:
  - Machines with a two-way, read-only input tape and two stacks.
  - Counter automata: A machine with a two-way, read-only input head and k integer counters.

We show equivalence of the variants of a Turing machine.

TM with multiple tapes



- Has a finite control
- Multiple tapes (say 3), each with its own read-write head.
- Each step: In current state p, read current symbols under the tape heads, say a, b, c: Change state to q, replace current symbols by a', b', c', and move each head left or right.

$$(p, a, b, c) \rightarrow (q, a', b', c', L/R, L/R, L/R).$$

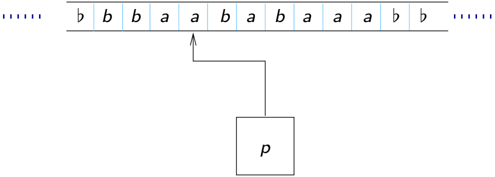
- Transition function: $\delta : Q \times \Gamma^3 \rightarrow Q \times \Gamma^3 \times \{L, R\}^3$.

Equivalence: Multi-tape TM and a single tape TM

Build a single tape TM N that simulates the given multi-tape M:

- The tape of N will have three *tracks*, the tape contents of the three tapes of M will be *merged* to form a tuple of the tape contents of N.
- Mark exactly one symbol in each track to indicate the position of the respective head.
- Simulate one step of M:
  N starts at the left end, scans out until all the three marked symbols are seen, remembering them in the finite control. It then uses the transition of M and based on this information, makes each of the individual transitions at the three marked positions.

TM with two-way infinite tape



- Finite control
- Single two-way infinite tape.
- To simulate, fix a cell and fold it to the right, creating another tape below the tape to the left of the fixed cell. This becomes like a two-track TM.

Non-deterministic TM

The transition function of a non-deterministic Turing Machine (NDTM) M is given by

- $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$

Run of a non-deterministic TM is a tree whose branches correspond to the possible options as determined by the given transition relation δ.

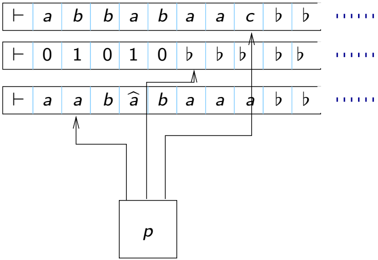
Non-deterministic TM

Theorem: Every non-deterministic Turing machine has an equivalent deterministic Turing machine.

- Simulate M by a deterministic TM M' that accepts the same language.
- M' simulates each tree representing a run of M by doing a BFS on the tree.
- Guaranteed to capture all paths and accept (reject) of the given non-deterministic TM accepts (rejects).

Simulating a non-deterministic TM

- M' uses 3 tapes: for given input, work tape to simulate run of M, another tape to identify the next node in the BFS traversal.
- Configurations are stored in the work tape.
- The tape for BFS traversal identifies the successor node to traverse to, by encoding the nodes in a suitable notation and using lexicographic ordering to identify the successor.



The Church-Turing Thesis

Church-Turing Thesis

The definition of computability based on Turing machines, captures the “right” notion of computability.

Turing computability coincides with several other notions of computability proposed based on different models, in the 1930’s:

- Post systems (Emil Post)
- μ-recursive functions (Gödel, Herbrand)
- λ-calculus (Church, Kleene)
- Combinatory logic (Curry, Schönfinkel)