Turing Machines

Prof. Shrisha Rao
IIIT Bangalore

srao@iiitb.ac.in

2025-09-09

Turing Machines

- We have seen that DFAs/NFAs accept regular languages, and PDAs accept context-free languages. We can thus wonder what type of machine can accept context-sensitive and other languages.
- The difference between finite automata and pushdown automata is only in the stack, which is a limited form of storage. By replacing this with a better form of storage, we get a more powerful type of machine.
- In place of a stack, consider an infinite read-write tape divided into cells, each of which can contain a single symbol.
- A machine of this type is known as a *Turing machine*, named after Alan Turing.

Church-Turing Thesis and Turing Completeness

- A Turing machine (TM) is the ultimate type of computing device that anyone has been able to describe. (There is no known type of computing abstraction beyond it.)
- The *Church-Turing Thesis* (due to Alonzo Church) is the accepted thesis (belief) that a TM is the ultimate computing device; our intuitive notion of "computability" is the same as saying that something is computable by a TM.
- Any other computing abstraction (e.g., two-stack PDA, partial recursive functions, λ calculus) that can be designed is at best as good as a TM.
- A computing abstraction (or even a programming language) is called *Turing complete* if it has the full expressive power of a TM.

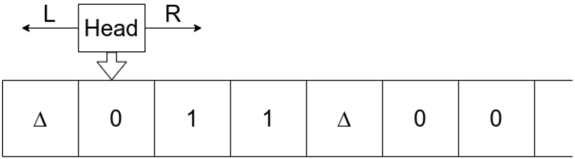
Formal Definition of a Turing Machine

- A Turing machine is a 6-tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$, where $Q, \Sigma, \delta, q_0, F$ are all as before, and Γ is a special *tape alphabet* of symbols written on the tape.
- Instead of a stack push/pop as with a PDA, a TM has a R/W head that can read a symbol from the tape at the current position, write a symbol on to the tape at the current position, or move one cell to the left or right.
- The tape alphabet Γ is a superset of the input alphabet Σ, especially containing the blank tape symbol Δ. (A cell on the tape is blank if it contains Δ.)
- The transition function $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R, S\}$ indicates the state change and possible tape head movement.

Transition Function of a TM

All inputs to a TM are written on the tape, which also is how the TM provides its output. At each step, a TM:

- Reads a symbol from the current tape cell.
- Writes a symbol on to the current tape cell.
- Moves left or right one cell, or stays in the same cell.
- Updates its state.

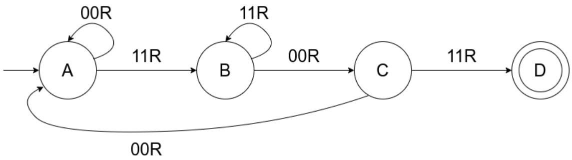


Schematic of a TM tape and head

Drawing a TM

As previously, a TM can be shown by a bubble diagram, but the labels on the arrows have three values:

oldSymbol newSymbol moveDirection



A TM to accept $\{0^n1^n\}$

If 0^k1^k is written on the tape for some k, then the idea is for the TM to successively erase the first 0 and the last 1, and keep doing this. If this ends with the tape blank, the TM accepts, else it rejects.

Exercise: give a pseudocode description of this approach.

Pseudocode for $\{0^n 1^n\}$

(i) If symbol read is 0, then write Δ (erase), else reject.

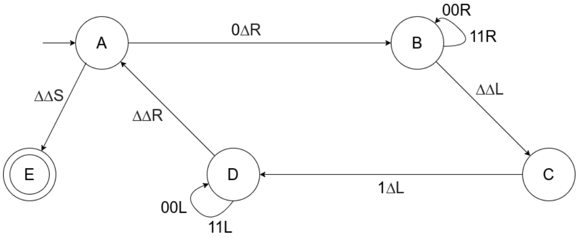
(ii) Move head right until symbol read is Δ.

(iii) Move head one step left. If symbol read is 1, write Δ, else reject.

(iv) Move head left until symbol read is Δ. Move head one step right. If symbol read is Δ then accept.

(v) Goto step (i).

TM FOR $\{0^n1^n\}$



Exercises

Come up with TMs for the following languages:

(1) The language $\{0^n 1^{2n} \mid n \geq 0\}$. Show the operation of the TM for the strings 001111 and 00011.

(2) The language of all balanced brackets over the alphabet (,). Show the operation of the TM for the strings (()()) and ())((.

(3) All strings of the form $\{0^n 1^n 2^n \mid n \geq 0\}$. Show the operation of the TM for $0^2 1^2 2^2$ and $0^2 1^3 2^2$.

For these exercises, come up with the TM logic and pseudocode first, and then only draw the diagram.

Turing Machines as Calculators

- So far we have looked at machines only as *language acceptors*, i.e., as devices that can accept or reject particular languages.
- A TM can also function as a *calculator* that processes numerical inputs and produces an output.
- By the Church-Turing thesis, any computable function that can be processed on any device in any manner whatsoever can be handled by a TM.

Giving Inputs to a Turing Machine

- As before, inputs to a Turing Machine are written on the tape at the outset.
- A TM can only accept natural numbers (zero and the positive integers) as inputs. (NB. This restriction applies in the natural setting to electronic calculators, CPUs, etc., also.)
- In case of multiple inputs to be given (e.g., two numbers), the same are written on the TM tape in sequence with a Δ in between.
- As before, it is presumed that the TM head starts its execution at the leftmost nonblank symbol on the tape.

Unary Alphabet

- To represent natural numbers on the tape, we use a *unary* (not binary) alphabet, with 1 being the only symbol.
- In this alphabet, a single 1 stands for the natural number 0, 11 stands for 1, and a natural number $n > 1$ is represented by 1^{n+1}, i.e., a sequence of $n + 1$ 1s.

The Simplest Possible TM Calculation

- Problem: Create a TM which adds 1 to a number it is given as input.
- Solution: Assume that 1^k is written on the tape, for some $k > 0$. The TM should add a single 1 to the string, to leave 1^{k+1} on the tape.

More Realistic Addition

- Problem: Consider a TM that is given natural numbers m and n. It is to add them to get the natural number $m + n$.
- Solution: Note that m is given in unary alphabet by 1^{m+1} and n is given by 1^{n+1}. Therefore the tape reads $1^{m+1}\Delta 1^{n+1}$ at the outset. The TM should leave the tape with 1^{m+n+1} at the end of its execution.
- It can do this by overwriting the Δ in the middle with a 1, and then deleting *two* 1s from either the beginning or the end of the combined string.
- Example: If starting with $111\Delta 111$ on the tape (representing 2, 2, the numbers to be added), overwrite the Δ with 1 first to get 1111111, then remove two 1s to get 11111, the final answer (representing 4).
- Addition of three of more numbers is handled in the same way, sequentially.

Multiplication by 2 is Addition with Itself

- Once we are able to get TMs to add two or more numbers, we can take on the next step of getting a TM to multiply two natural numbers.
- Problem: Given a number m in unary alphabet on the tape, compute $2m$.
- Solution: The tape reads 1^{m+1} to start with. It should be left with 1^{2m+1}. The best approach is to first repeat the first number m on the tape leaving a blank in between (change the tape to $1^{m+1}\Delta 1^{m+1}$), and then add up as previously.

Exercises

(4) Give a TM that adds 3 to any number given to it as input.

(5) Prove that the function $\lfloor \frac{x}{2} \rfloor$ is Turing computable, by giving a TM that computes it.