

# What is Meant by a Turing Machine Accepting a Language

But first let's talk about a Universal Turing Machine:

Let  $P$  be a program in  $S$  that computes a partial function of one variable. We say that  $P$  computes

$$\psi_P^{(1)}(x)$$

As you have seen, we can now conceive of a partial function  $\Phi(x, z)$  with  $z = \#(P)$  such that

$$\Phi(x, z) = \psi_P^{(1)}(x) \rightarrow z = \#(P)$$

Since we already know that  $\Phi(x, z)$  is computed by the universal program  $U$ , in  $S$ , there must obviously exist a Turing Machine  $M$  that starts in the configuration

$$B \uparrow x B z$$

and that computes  $\Phi(x, z)$ . If necessary, we can construct  $M$  by first converting  $U$  into a Post-Turing program (by Theorem 5.11) and then translating the Post-Turing program into a Turing Machine by Theorem 1.1 of Chap. 6. This  $M$  will compute any partially computable function of 1 variable. Note that  $M$  stores its data and the program in a single "memory".

- Given a TM with alphabet  $A = \{s_1, \dots, s_n\}$ , a word  $u \in A^*$  is said to be accepted by  $M$  if when  $M$  begins with the configuration

$$B \uparrow u$$

it eventually halts.

- The set of all words  $u \in A^*$  that  $M$  accepts is the language accepted by  $M$ .
- Earlier we said that a set is r.e. if membership in the set is determined by whether or not a function is defined for that element value (Lecture 9).
- Since the halting of a TM implies that the function computed by the TM is defined for that input, ~~function~~ we can say that the language accepted by a TM is recursively enumerable. In fact, we can make the following assertion:
- A language is accepted by a TM iff the language is r.e.

- Direct implication of the above: If  $L$  is accepted by  $M$ , testing for word membership in  $L$  is an unsolvable problem.
- Therefore, an r.e.  $L$  must also be recursive if testing for word membership is required.

- $L \subseteq A^*$ . Now consider a larger alphabet  $\tilde{A}$ . That is  $A \subseteq \tilde{A}$ . The  $L$  is an r.e. language of  $A$  iff  $L$  is an r.e. language on  $\tilde{A}$ .

- Same as above for  $L, A, \tilde{A}$ .  $L$  is a recursive language on  $A$  if and only if  $L$  is a recursive language on  $\tilde{A}$ .

$$\text{recursive } L \Rightarrow L \text{ and } A^* - L \text{ both r.e.}$$

$$\tilde{A}^* - L = (\tilde{A}^* - A^*) \cup (A^* - L) \text{ is also r.e.}$$

# Nondeterministic Turing Machines:

- is an arbitrary finite set of quadruples.
- theoretically speaking, every deterministic TM is also a nondeterministic TM.
- may contain multiple quadruples that start with the same  $q_i, s_j$  pair.
- this results in choice of action if the machine is in state  $q_i$  with the scanhead looking at symbol  $s_j$  and there exists more than one quadruple ~~for~~ that begins with  $q_i, s_j$ .
- Let  $A = \{s_1, \dots, s_n\}$  and  $u \in A^*$ , a nondeterministic TM  $M$  is said to accept  $u$  provided there exists a sequence of <sup>tape</sup> configurations  $\gamma_1, \gamma_2, \dots, \gamma_m$  such that  $\gamma_1$  is the configuration

$\gamma_1 \xrightarrow{B} \gamma_2 \xrightarrow{u} \gamma_3 \xrightarrow{\dots} \gamma_m$

and  $\gamma_m$  is terminal (in the sense that there does not exist a quadruple that begins with the state and the symbol being scanned particular to  $\gamma_m$ ), and with the transitions between the configurations indicated by

$\gamma_1 \xrightarrow{} \gamma_2 \xrightarrow{} \gamma_3 \xrightarrow{} \dots \xrightarrow{} \gamma_m$  transition allowed by  $M$

- We say the sequence  $\gamma_1, \gamma_2, \dots, \gamma_m$  is an accepting computation by  $M$  for  $u$ .
- The language  $L$  accepted by a nondeterministic TM  $M$  is the set of all  $u \in A^*$  for which there exists an accepting computation.
- But note that for a nondeterministic TM, even when there exists an accepting computation for a string  $u$ , for the same string it may be possible to construct an infinite sequence of configurations

$\gamma_1 \xrightarrow{} \gamma_2 \xrightarrow{} \dots$

- For every r.e.  $L$ , there is a nondeterministic TM that accepts  $L$ . (True because, by definition, a language is r.e. if it is accepted by a deterministic TM and, by definition, a deterministic TM is a nondeterministic TM.)
- The converse of the above is also true, but we need the notions of processes and grammars to prove that.

Summary of what we have done so far:

The following assertions are equivalent:

- $f$  is partially computable (in  $S$ )
- $f$  is partially computable in  $S_n$  for all  $n \geq 1$
- $f$  is partially computable by a Post-Turing program
- $f$  is partially computable by a Turing Machine
- $f$  is partially computable by a quintuple Turing Machine

