Turing recognizable Languages

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in

Formerly at department of Computer Science and Engineering
MBM Engineering College, Jodhpur

Monday 10^{th} April, 2017

Enumerating TMs

- We can enumerate Turing machines, by encoding each one of them, say:
- TM-5012847892 = Balancing parenthesis
- TM-5025672893 = Even number of 1s
- TM-5256342939 = Universal TM
- TM-56239892122 = Windows XP
- ...

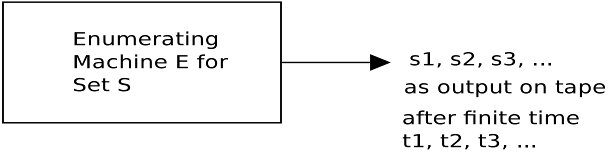
Thus, a TM can be described by a set of 0s and 1s. This set forms a languages,

L = { 000101001010101000, (Turing machine1)
000101011110101000, (Turing machine2)
00011101010100010101000, (Turing machine ...)
... }

L is countable set of infinite number of strings (how?)

Enumerating TMs

- There is one-to-one correspondence between elements of the set of TMs and the natural numbers. Let S be set of strings. An enumeration procedure for S is a TM that generates all strings of S one-by-one, each in finite time. $s_1, s_2, \cdots \in S$. (Hint: algorithm to increment a number)



- If for a set there is a an enumeration procedure for a set, then the set is countable.

  Ex.: Prove that set of all the strings over {a, b, c} is countable

  Put in proper order:

  Produce all strings of length 1

  produce all strings of length 2, and so on

Enumerating TMs

Theorem

Set of All the Turing machines is countable.

Proof.

- Any Turing machine can be encoded in binary strings of 0's and 1's.
- Find an enumeration procedure for the set of TMs.

□

Enumeration of Turing Machines:(Repeat):

1. Generate the next binary string of 1s and 0s in proper order
2. Check if the string describes a Turing machine(an encoding of some TM).
   i if yes: print the string on output tape
   ii if no, ignore it.

Countable and uncountable sets:

- Let a set of strings $S = \{s_1, s_2, \ldots, \}$ is countable. The s_i are generated through enumerating procedure.
- Power set for S is 2^S, is not countable(?).
- Let the elements of power set be: $\{s_1\}, \{s_2, s_3\}, \{s_1, s_3, s_4\}$, etc. We can encode the elements of power set as binary strings of 1s and 0s:

Power set element	Power set	Encoding s_1	s_2	s_3	s_4	...
t_1	$\{s_1\}$	1	0	0	0	...
t_2	$\{s_2, s_3\}$	0	1	1	0	...
t_3	$\{s_1, s_3, s_4\}$	1	0	1	1	...

Power set is uncountable

- Let us assume (for contradiction) that power set is countable. Then we can enumerate its elements
- Take the power set elements whose bits are the complement of diagonal

Power set element	Encoding s_1	s_2	s_3	s_4	...
t_1	**1**	0	0	0	...
t_2	0	**1**	1	0	...
t_3	1	0	**1**	1	...

Power set is uncountable

The complement is: 000 (a binary complement of diagonal). This new element must be some element t_i of power set (since we assume that P(S) is enumerated). However, that is impossible. Hence, we conclude that power set is uncountable.

Countable TM v/s Uncountable Languages:

- For $\Sigma = \{a, b\}$, Σ^* is countable, because Σ^* can be enumerated. $\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$, which maps to {0, 1, 2, 3, ...}
- However, the languages $\{L_1, L_2, \dots\}$ that can be constructed from Σ^*, are subsets of 2^{Σ^*}; are uncountably infinite.
- All the Turing machines $\{M_1, M_2, \dots\}$ can be enumerated (ref. representation of all TMs), which is countably infinite.
- **Conclusion:** There are more languages than TMs, hence for some languages there does not exist TMs. In fact they are not *Turing recognizable*.
- What are those languages?