Recursive Enumerable Languages

• Recursive Enumerable Languages
  A language that is accepted by a TM is said to be recursively enumerable (r.e.). Enumerable means that all strings of a recursive enumerable could be listed by a TM.

• A TM may divide all strings on $\Sigma^*$ into three sets:
  – all accepted strings (strings on which TM halts)
  – all strings on which TM crashes
  – all strings on which TM loops
Recursive Languages

- Recursive Languages
  A language that is accepted by at least one TM that halts or crashes on all strings is called recursive language. That is no string loops on the TM.

- A recursive language is a recursive enumerable language but a recursive enumerable may not be a recursive language
Properties of r.e. and Recursive Languages

• Theorem 1
  – The complement of a recursive language is recursive.

• Theorem 2
  – The union of two recursive languages is recursive.

• Theorem 3
  – The union of two recursive enumerable languages is recursive enumerable.
Properties of r.e. and Recursive Languages

• Theorem 4
  – If a language $L$ and its complement $L'$ are both recursive enumerable, then $L$ (and hence $L'$) is recursive.

• From theorem 1 and 4, we may derive one of the following
  – both $L$ and $L'$ are recursive
  – neither $L$ nor $L'$ is r.e.
  – one of $L$ and $L'$ is r.e., but not recursive; the other is not r.e.
TM Code

- Encode a TM in a string over \( \{0, 1\}^* \)
- Assume \( M = (\{q_1, q_2, \ldots, q_n\}, \{0, 1\}, \{0, 1, \Delta\}, \delta, q_1, \{q_2\}) \)
- States are represented as numbers 1 to n.
- Symbols 0, 1, and \( \Delta \) are represented as 1, 2, and 3 respectively.
- Head moves L and R are represented as 1 and 2 respectively.
TM Code

- Each move $\delta(i, j) = (k, l, m)$ where $i$ and $k = 1, \ldots, n$, $j$ and $l = 1, 2, \text{ or } 3$, and $m = 1 \text{ or } 2$, is encoded as $0^{i}10^{j}10^{k}10^{l}10^{m}$
- A TM is a set of moves and may be represented as $111\text{code}_111\text{code}_211\ldots11\text{code}_t111$
- Each string may be interpreted as the code for at most one TM.
- Not all strings can be interpreted as a TM
- A TM may have many different codes.
Non-Recursive Enumerable Language

- Let $L_d = \{w | w$ doesn’t represent any TM or $w$ is not accepted by the TM represented by $w\}$
- Assume that $L_d$ is r.e. and $M$ is the TM that accepts $L_d$.
- Let $w_m$ is the code representing $M$.
- If $w_m$ is in $L_d$, $w_m$ must be accepted by $M$ because $M$ accepts $L_d$. If $M$ accepts $w_m$, then $w_m$ is not in $L_d$. It is a contradiction.
- If $w_m$ is not in $L_d$, $w_m$ must not be accepted by $M$ because $M$ accepts $L_d$. If $M$ does not accept $w_m$, then $w_m$ is in $L_d$. It is a contradiction. Therefore no TM accepts $L_d$. 
Universal TM

• A universal TM (UTM) is TM that accepts an input string composed of two parts: a TM code representing M followed by an input string w for M.
• The UTM accepts its input string if M accepts w.
• The UTM crashes on its input string if M crashes on w.
• The UTM loops on its input string if M loops on w.
• UTM exists.
Non-Recursive Language

• Let \( L_u = \{w | w \text{ represents a TM } M \text{ and } M \text{ accepts } w \} \)
• \( L_u \) is recursive enumerable.
• \( L_u \) is not recursive.
  – \( L_u \) is complement of \( L_d \).
  – If \( L_u \) is recursive, then \( L_d \) is recursive according to Theorem 1.
• Theorem
  – The complement of a recursive enumerable language may not be recursive enumerable.
Undecidable

- The following properties of r.e. languages are not decidable
  - A TM halts
  - A TM accepts $\varepsilon$.
  - A TM accepts no string (emptiness).
  - A TM accepts a finite number of strings (finiteness)
  - A TM accepts a regular language
  - A TM accepts a context-free language.