* ## Introduction to Undecidability :-

A problem is said to be decidable, if its language is recursive. Otherwise, the problem is undecidable. It means that, a problem is undecidable if there is no algorithm that takes as input an instance of the problem and determines whether the answer to that instance is "yes" or "no."

By a problem, we will understand a set of related statements, each of which must be either true or false.

e.g., we consider the statement "For a context-free grammar G, the language L(G) is ambiguous." For some G this is true, for others it is false, but clearly we must have one or the other. The problem is to decide whether the statement is true for any G we are given.

We say that a problem is decidable if there exists a T.M. that gives the correct answer for every statement in the domain of the problem.

When we state decidability or undecidability results, we must always know what the domain is, because this may affect the conclusion. The problem may be decidable on some domain but not on another. Specifically, a single instance of a problem is always decidable, since the answer is either true or false.

* Rice's Theorem -

A property of Languages is a function

$$P \begin{Bmatrix} \text{set of all} \\ \text{languages} \end{Bmatrix} \rightarrow \{0, 1\}$$

For a language L, if $P(L) = 1$ then we say that L satisfies P, and if $P(L) = 0$ then we say that L does not satisfy P.

P is said to be a non-trivial property of languages of T.M.s if there exists T.M.s M_1 and M_2 such that $P(L(M_1)) = 1$ and $P(L(M_2)) = 0$.

* Post Correspondence Problem -

Ex-(1.) We have some set of elements :-

$$\frac{B}{CA} \quad \frac{A}{AB} \quad \frac{CA}{A} \quad \frac{ABC}{C}$$

We need to find a sequence of sets such that the top and bottom strings are the same.

$$\overset{2}{\frac{A}{AB}} \quad \overset{1}{\frac{B}{CA}} \quad \overset{3}{\frac{CA}{A}} \quad \overset{2}{\frac{A}{AB}} \quad \overset{4}{\frac{ABC}{C}}$$

→ ABC A A ABC
→ ABC A A ABC

Ex-(2.)

$$\frac{1}{111} \quad \frac{10111}{10} \quad \frac{10}{0}$$

We need to find a sequence of sets such that the top and bottom strings are the same.

$$\overset{2}{\frac{10111}{10}} \quad \overset{1}{\frac{1}{111}} \quad \overset{1}{\frac{1}{111}} \quad \overset{3}{\frac{10}{0}}$$

→ 10111
→ 10111

Ex-(3.)

$$\frac{10}{101} \quad \frac{011}{11} \quad \frac{101}{011}$$

We need to find a sequence of sets such that the top and bottom strings are the same.

101 0 10 101 011
101 1 01 101 011 11
X

So, this instance of PCP has no solution.

* Modified PCP (Post Correspondence Problem) :-

We show that PCP is undecidable by showing that if it were decidable, we would have an algorithm for the problem. First, we show that, if PCP were decidable, a modified version of PCP would also be decidable.

The Modified Post's Correspondence Problem (MPCP) is the following:-

Given lists A and B, of k strings each from Σ^*, say

$A = w_1, w_2, \ldots, w_k$ and $B = x_1, x_2, \ldots, x_k$,

does there exist a sequence of integers $i_1, i_2, \ldots, i_r$, such that

$$w_1 w_{i_1} w_{i_2} \ldots w_{i_r} = x_1 x_{i_1} x_{i_2} \ldots x_{i_r} ?$$

The difference between the MPCP and PCP is that in the MPCP, a solution is required to start with the first string on each list.

* **NP- Completeness** - We have seen that a solvable problem is one that admits an algorithm. In this case, a problem is solvable 'in principle'. But, in practice, the algorithm may require a lot of space (memory) and time (time to apply the steps of the algorithm). When the space and time required for implementing the algorithm are reasonable, we can say that the problem is tractable (i.e. solvable in practice).

A decision problem is tractable if there is an algorithm to solve the given problem and the time required is expressed as a polynomial $p(n)$, n being the length of the input string. Usually problems are intractable if the time required for any of the algorithm is at least $f(n)$, where f is an exponential function of n.

* Introduction to Recursive Function Theory -

Primitive Recursive Functions -

(1.) The Zero function $z(x) = 0$, for all $x \in I$.

(2.) The successor function $s(x)$, whose value is the integer next in sequence to x, that is, in the usual notation, $s(x) = x + 1$.

(3.) The projector functions
$$p_k(x_1, x_2) = x_k, \quad k = 1, 2.$$

There are two ways of building more complicated functions from these:-

(1.) Composition, by which we construct
$$f(x, y) = h(g_1(x, y), g_2(x, y))$$
from defined functions g_1, g_2, h.

(2.) Primitive Recursion, by which a function can be defined recursively through
$$f(x, 0) = g_1(x),$$
$$f(x, y+1) = h(g_2(x, y), f(x, y)),$$
from defined functions g_1, g_2, and h.