Theory of Automata and Formal Language

Lecture-40

Dharmendra Kumar
(Associate Professor)
Department of Computer Science and Engineering United College of Engineering and Research, Prayagraj March 30, 2021

Post Correspondence Problem(PCP)

The PCP problem over an alphabet ∑ is stated as follows:–

Given the following two lists, **X** and **Y** of non-empty strings over ∑,

$$X = (x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$$

$$Y = (y_1, y_2, y_3, \ldots\ldots\ldots, y_n)$$

We can say that there is a Post Correspondence Solution, if for some $(i_1, i_2, \ldots\ldots\ldots\ldots i_k)$, where $1 \le i_j \le n$, the condition

$$x_{i1}\, x_{i2} \ldots\ldots. x_{ik} = y_{i1}\, y_{i2} \ldots\ldots. y_{ik}$$ satisfies.

Ex. Find whether the lists X = (abb, aa, aaa) and Y = (bba, aaa, aa) have a Post Correspondence Solution?

Solution:

Here,

$$\mathbf{x_2x_1x_3 = \text{'aaabbaaa'}}$$

and $\mathbf{y_2y_1y_3 = \text{'aaabbaaa'}}$

We can see that

$$\mathbf{x_2x_1x_3 = y_2y_1y_3}$$

Hence, the solution is **(2, 1,3).** Another solution may be also **(2, 3), (3, 2).**

Ex. Find whether the lists $X = (b, bab^3, ba)$ and $Y = (b^3, ba, a)$ have a Post Correspondence Solution?

Solution:

$$\mathbf{x_2x_1x_1x_3 = bab^3\ b\ bba}$$

and $\mathbf{y_2y_1y_1y_3 = ba\ b^3b^3a}$

Therefore the solution will be (2, 1, 1, 3).

Ex. Find whether the lists $X = (ab, bab, bbaaa)$and $Y = (a, ba, bab)$ have a Post Correspondence Solution?

Solution:

In this case there is no solution of this problem. Because the length of each string in Y is less than corresponding string in X.

That is $|y_i| < |x_i|, \ \forall\, i.$

Modified Post correspondence problem (MPCP)

The modified PCP problem over an alphabet $\sum$ is stated as follows:–

Given the following two lists, **X** and **Y** of non-empty strings over $\sum$,

$$X = (x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$$

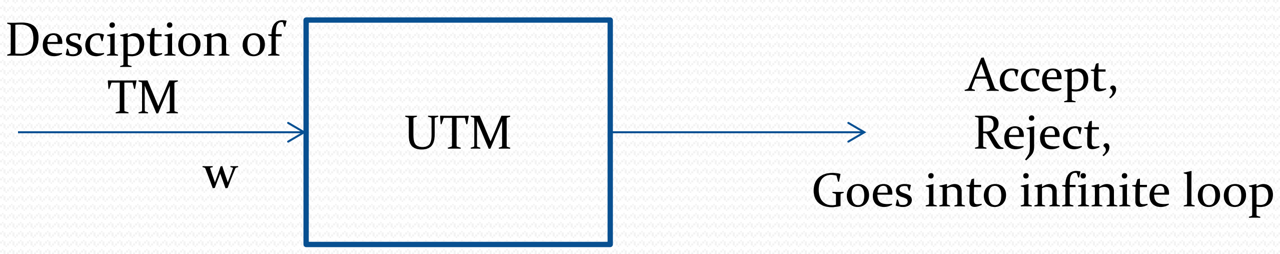
$$Y = (y_1, y_2, y_3, \ldots\ldots\ldots, y_n)$$

We can say that there is a Modified Post Correspondence Solution, if for some $(i_1, i_2, \ldots\ldots\ldots\ldots i_k)$, where $1 \leq i_j \leq n$, the condition $x_1\, x_{i1}\, x_{i2} \ldots\ldots\ldots x_{ik} = y_1\, y_{i1}\, y_{i2} \ldots\ldots\ldots y_{ik}$ satisfies.

Universal Turing machine(UTM)

- A universal Turing machine (UTM) behaves like a general purpose computer. Instead of finite size memory in computer, UTM uses infinite tape.
- UTM is a specified TM that can simulate the behavior any TM.
- UTM is a Turing machine that accepts universal language.
- Universal language is defined as:-

UL= { <M,w> ! M is a Turing machine that accepts input string w.}



Universal Turing machine(UTM)

Input to UTM:

Description of TM

Input string

Action of UTM:

Simulate TM

Behave like TM

UTM as Computer

TM as Program

UTM is a recognizer but not a decider.

UTM takes an encoding of a TM and the input data as its input in its tape and behaves as that TM on the input data.

Church-Turing Thesis

- It states that if there exists an algorithm to solve a problem then there exists a Turing machine to solve that problem and vice-versa.
- It states that a function on the natural numbers can be computed by an algorithm if and only if it is computable by a Turing machine.
- A problem can be solved by an algorithm iff it can be solved by a Turing Machine.
- Algorithm $\longleftrightarrow$ Turing machine

Halting Problem

Statement: Given Turing machine M and input string w, **is it possible to determine whether the machine will ever halt on given input string?**

In another words, the **halting problem** is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running, or continue to run forever.

Halt: the machine will stop or halt at final or non-final state after finite number steps.

No halt: Machine will never stop or halt.

Decidable or Undecidable Problem

- A problem is said to be decidable if there exists an algorithm which can decide the problem in finite amount of time.
- In this type of problems, the output of the algorithm will be yes/no i.e. the answer of decidable problems is yes or no.
- A problem is said to be undecidable if there does not exist an algorithm which can decide the problem in finite amount of time.

Turing decidable and Turing acceptable language

- A language L is said to be Turing decidable if there exists a Turing machine which can accepts all strings belong in to L and rejects all strings which do not belong into L.
- A language L is said to be Turing acceptable if there exists a Turing machine which can accepts all strings belong in to L.

Some undecidable problems

- Halting problem is undecidabe.
- PCP problem is undecidable.
- Modified PCP problem is undecidable.
- For a CFG G, is L(G) ambiguous ?
- For two arbitrary CFG G_1 and G_2,
  deciding $L(G_1) \cap L(G_2) = \phi$ or not, is undecidable.

Some decidable problems

- For a CFG G, is L(G) = ϕ or not, is decidable.
- For a CFG G, finding whether L(G) a finite or not, is decidable.
- For regular language L_1 and L_2, finding whether $L_1 \cup L_2$ is regular, is decidable.
- Membership problem in CFG is decidable.