Problem

Show that the Post Correspondence Problem is undecidable over the binary alphabet ∑ = {0,1}.

Step-by-step solution

Step 1 of 1

Post Correspondence Problem is basically concerned with manipulation of string and used to find match.

Conceptually this concept is quite obvious by its statement itself as there are 2 alphabets in the Post Correspondence Problem. PCP is un-decidable over the two alphabets 0 and 1. This can be proved by using contradiction.

Assume the instance of PCP where the size of alphabet is *n* such that $n \le 2^m$, *m* is a constant. M-bit code is generated for the alphabets and then substituted in the tile.

Consider the tile in PCP given below:

$$\{[\frac{aba}{acd}],[\frac{aabd}{acd}],[\frac{a}{bc}],[\frac{acd}{bc}],[\frac{bcd}{abc}]\}$$

If the values of *a*, *b*, *c* and *d* is assigned as 00, 01, 10 and 11, the tiles given above will be written as:

$$\{[\frac{000100}{001011}],[\frac{00000111}{001011}],[\frac{00}{0110}],[\frac{001011}{0110}],[\frac{011011}{000110}]\}$$

The new PCP instance have alphabet size 2 will have solution if the original PCP instance have some solution. In the code, one-for-one substitution takes place. So, if the new instance has a solution, the solution will also exist for the original instance.

It should be taken care of the case when the new tiles provide a solution but the original tiles don't provide any solution. But this will never happens it is because if the tiles are obtained after inserting the values in the original instance.

For unary alphabets, the proof does not work. The codes are applied for *a*, *b*, *c* and *d* as 1, 11, 111 and 1111. If tracing is done backwards, it will be difficult to identify whether 1111 refers to *d*, *bb*, *ac* or *ca*(there will be other possibilities too). So, PCP is un-decidable.

Comment