Geometry of Post's Correspondence Problem

Mostafa Ghandehari

TR # 322
GEOMETRY OF POST'S CORRESPONDENCE PROBLEM
Mostafa Ghandehari

Abstract. Orthogonality of vectors with integer coordinates in an \( n \)-dimensional Euclidean space is used to show that Post's correspondence problem is solvable for words over a one-symbol alphabet. We also use orthogonality to discover a match for an instance of Post's correspondence problem with three symbols.

Key Words. Solvability, Post's correspondence problem
AMS(MOS) subject classification 68

Introduction. A well-known unsolvable problem is Post's correspondence problem (named after the American mathematician Emil L. Post, 1897-1954), formulated as follows: A correspondence system is a finite set \( P \) of ordered pairs of nonempty strings. A match of \( P \) is any string \( w \) such that for some \( n > 0 \) and some (not necessarily distinct) pairs \( (u_1, v_1), (u_2, v_2) \ldots (u_n, v_n) \in P \),

\[ w = u_1u_2\ldots u_n = v_1v_2\ldots v_n. \]

Post's correspondence problem is to determine, given a correspondence system, whether that system has a match. Post's correspondence problem is unsolvable. That is, given a correspondence system there is no algorithm to decide if there is a match for the system. The original proof of unsolvability of Post's correspondence problem appeared in [5]. For the definition of solvability in terms of Turing machines, as well as a proof of unsolvability of Post's correspondence problem, see [3] or [4]. A discussion of Post's correspondence problem is contained in [2].

Another famous unsolvable problem is the halting problem, which is that of deciding whether or not an arbitrary Turing machine presented with an arbitrary input string will ever halt. Many problems can be shown to be unsolvable by showing that they are
equivalent to the halting problem or Post's correspondence problem. For example, the problem of deciding whether or not a grammar is ambiguous (that is, whether or not there is a string in the language generated by the grammar for which there are two distinct parse trees) is unsolvable, since this problem is equivalent to Post's correspondence problem.

To describe the geometric approach proposed in this note consider the example given below.

Example. Find a match for the following instance of Post's correspondence problem

\[ P = \{ (b, ca), (a, ab), (ca, a), (abc, c) \} \] .

Discussion. The matching string is \( w = abcaabc \) where one copy of the first pair, two copies of the second pair, one copy of the third pair, and one copy of the fourth pair are used. The above answer can be guessed geometrically as follows:
Assume \( j_i \) is the number of copies of \( i \)th pair used in the match where \( i = 1, 2, 3, 4 \). It is necessary that the number of a's that result from using strings from the first components equal the number of a's that result when juxtaposing the corresponding strings from the second components. Similarly for the b's and the c's. This gives the three equations:

\[ j_1(0-1) + j_2(1-1) + j_3(1-1) + j_4(1-0) = 0 \]
\[ j_1(1-0) + j_2(0-1) + j_3(0-0) + j_4(1-0) = 0 \]
\[ j_1(0-1) + j_2(0-1) + j_3(0-0) + j_4(1-0) = 0 \]

Thus the vector \( \mathbf{j} = (j_1, j_2, j_3, j_4) \) is orthogonal to the three vectors \( \mathbf{\vec{u}} = (-1, 0, 0, 1), \mathbf{\vec{v}} = (1, -1, 0, 1) \) and \( \mathbf{\vec{y}} = (-1, 0, 1, 0) \). We let \( \mathbf{\hat{u}}, \mathbf{\hat{v}}, \mathbf{\hat{z}}, \mathbf{\hat{w}} \) be standard unit vectors in \( \mathbb{R}^4 \) and formally expand the following determinant using the first row to obtain
the generalized cross product of three vectors \( a, \hat{a} \) and \( b \) in \( \mathbb{R}^4 \).

\[
\begin{bmatrix}
\hat{e}_1 & \hat{e}_2 & \hat{e}_3 & \hat{e}_4 \\
-1 & 0 & 0 & 1 \\
1 & -1 & 0 & 1 \\
-1 & 0 & 1 & 0 \\
\end{bmatrix}
\]

The resulting vector is \(-\hat{e}_1 - 2\hat{e}_2 - \hat{e}_3 - \hat{e}_4\). This suggests we choose \( \mathbf{j} = (1, 2, 1, 1) \) which is orthogonal to \( \hat{a}, \hat{b}, \hat{c} \). This suggests one copy of the first pair, two copies of the second pair, one copy of the third pair, and one copy of the fourth pair of strings of alphabets in the instance of Post's correspondence problem given above. We then need to try different possibilities to come up with a match. Hence, this approach is a way of discovering a match, assuming there is one, rather than an algorithm. A discussion of the generalization of cross product of vectors in terms of determinants can be found in [1]. Orthogonality can be used to solve the following exercise.

**Exercise.** Post's correspondence is solvable for a one letter alphabet.

**Solution.** Let \( \mathbf{P} = \{(a_1^{n_1}, a_1^{m_1}), (a_2^{n_2}, a_2^{m_2}), ..., (a_k^{n_k}, a_k^{m_k})\} \). Assume \( j_i \) copies of the pair \( (a_i^{n_i}, a_i^{m_i}) \) is used. Let \( a_i = n_i - m_i \) and \( \vec{a} = (a_1, a_2, ..., a_k) \). Then we require the dot product \( \vec{j} \cdot \vec{a} = 0 \).

Since the components of \( \vec{j} \) are non-negative integers, the vector \( \vec{a} \) must lie in an orthant with some negative and some non-negative components. Otherwise \( \vec{a} \) will not be in the orthogonal subspace of \( \mathbb{R}^k \) orthogonal to \( \vec{j} \). Thus, to decide if there is a match, we can proceed as follows.
Calculate the vector $\tilde{\alpha}$. If $\tilde{\alpha}$ is in the orthant with all non-negative or all negative components then there is no match. Otherwise there is a match. To come up with the match, without loss of generality assume $\alpha_1, \alpha_2, \ldots, \alpha_r$ are non-negative and $\alpha_{r+1}, \ldots, \alpha_k$ are negative. Choose $j_1 = j_2 = \ldots = j_r = -(\alpha_{r+1} + \ldots + \alpha_k)$ and $j_{r+1} = \ldots = j_k = (\alpha_1 + \ldots + \alpha_r)$.

Then $j$ is orthogonal to $\tilde{\alpha}$ and all $j_i$ are non-negative integers as desired.

REFERENCES


Mostafa Ghandehari
Department of Mathematics
University of Texas at Arlington
Box 19408
Arlington, TX 76019