Post's Correspondence Problem (PCP) Is Undecidable

Presented By Bharath Bhushan Reddy Goulla

Contents :

Introduction:

The Post’s Correspondence Problem is an undecidable decision problem that was introduced by “ Emil Leon Post ” in 1946.

Post’s Correspondence Problem (PCP) involves strings rather than TM. Our goal is to prove this problem about strings to be undecidable, and then use its Un-decidability to prove other problems undecidable by reducing PCP to those.

➢What Is Post's Correspondence Problem (PCP) ?

A PCP consists of two lists of string over some alphabet Σ; the two lists must be of equal length.

Generally $A=w_1, w_2, w_3, \ldots\ldots.w_k$ and $B= x_1, x_2, x_3, \ldots\ldots x_k$ for some integer k.

For each i , the pair (w_i , x_i) is said to be a corresponding pair.

We say this instances of PCP has a solution, if there is a sequence of one or more integers $i_1, i_2, \ldots\ldots., i_m$ that, when interpreted as indexes for strings in the A and B lists, yield the same string.

Contd.

That is, $w_{i1}\, w_{i2} \ldots\ldots.\, w_{im} = x_{i1}\, x_{i2} \ldots\ldots.\, x_{im}$. We say the sequence $i_1, i_2, \ldots\ldots., i_m$ is a solution to this instance of PCP, if so.

Instances Of PCP !

1) Consider two lists A and B

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

List Order :

2

w2	10111
x2	10

Instances Of PCP ! (Contd.)

Consider two lists A and B

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

List Order :

2 1

w2 w1	10111	1
x2 x1	10	111

➤Instances Of PCP ! (Contd.)

Consider two lists A and B

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

List Order :

2 1 1

w2 w1 w1	10111	1	1
x2 x1 x1	10	111	111

➢Instances Of PCP ! (Contd.)

Consider two lists A and B

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

List Order :

2 1 1 3

w2 w1 w1 w3	10111	1	1	10
x2 x1 x1 x3	10	111	111	0

Both list order arrangement makes them similar sequence.

Contd.

- The above instance of PCP have a solution in the order of indexes 2,1,1,3.
- The solution can also be expressed in terms of 2,1,1,3,2,1,1,3,2,1,1,3 and so on.

Another Instance of PCP :

2) Consider Lists A and B

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011

Contd.

2) Consider Lists A and B

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011

List Order :

1

W1	10
X1	101

- We cannot start with 2 and 3 because the starting doesn't match with each other.
- So the only choice in the list is 1.

Contd.

Consider Lists A and B

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011

List Order :

1 3

W1 w3	10	101
X1 x3	101	011

Contd.

Consider Lists A and B

	List A	List B
i	w_i	x_i
1	10	101
2	011	11
3	101	011

Partial Solution

List Order :

1 3 3

W1 w3 w3	10	101	101
X1 x3 x3	101	011	011

- Always at the end we see an extra 1 in x list sequence and this make us repeatedly to go for index 3.
- These continues which results in **Partial Solution** or no solution.

Introduction Of Modified PCP !

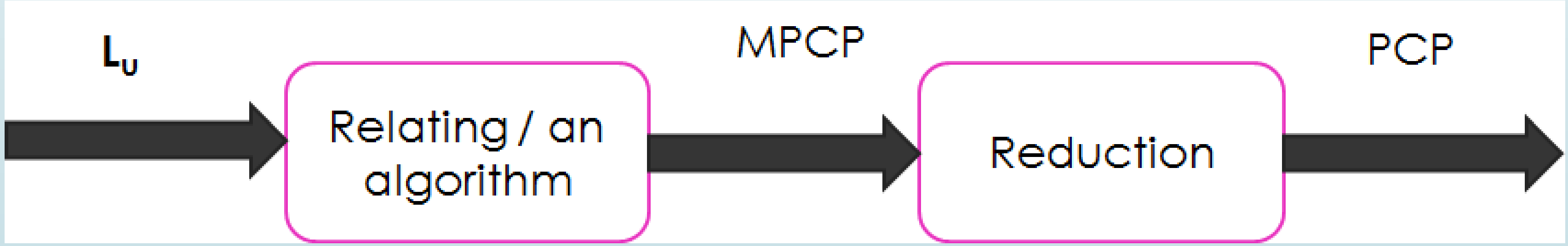
In MPCP, there is the additional requirement on a solution that the first pair on the A and B lists must be the first pair in the solution.

More formally, an instance of MPCP is two lists $A=w_1, w_2, w_3, \ldots\ldots. w_k$ and $B= x_1, x_2, x_3, \ldots\ldots x_k$ for some integer k.

And a solution is a list of 0 or more integers $i_1, i_2, \ldots, i_m$ Such that $\mathbf{w_1} w_{i1}\, w_{i2} \ldots\ldots w_{im} = \mathbf{x_1}\, x_{i1}\, x_{i2} \ldots\ldots x_{im}$.

➢ Why MPCP ?

- We use MPCP as a medium for proving PCP is Un-decidable.
- First we reduce L_U (A_{TM}) to MPCP instance.
- We Reduce MPCP instance to PCP instance.
- As from reductions from L_U to PCP as MPCP in middle It become easy to prove that PCP is Undecidable.



➢Reducing MPCP To PCP !

Rules for conversion of MPCP to PCP:

1. Add a symbol (*) after every string in the **list(w_i)** of MPCP.
2. Add a symbol (*) before every string in the **list(x_i)** of MPCP.
3. We need to observe that we will have k+2 indexes after conversion of MPCP to PCP.
   1. For instance if we have from 1 in MPCP then we start with 0 in PCP such that **list(y_i)** will have a symbol before 1 index **list(y_i)** and **list(z_i)** will have same as of 1 index value in **list(z_i)**
   2. Extra Stings in lists are initial and final terminals for identification, i.e.; $ and *$ { **list(y_i)** and **list(z_i)** } respectively.

MPCP	PCP
list(w_i)	**list(y_i)**
list(x_i)	**list(z_i)**

Example of Reducing MPCP to PCP.

Consider two lists A and B

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Consider this as a MPCP lists

Corresponding PCP instance

	List C	List D
i	y_i	z_i
0	*1*	*1*1*1
1	1*	*1*1*1
2	1*0*1*1*1*	*1*0
3	1*0*	*0
4	$	*$

Contd.

- Above is the example that we can change any instance of MPCP to PCP by reduction following rules.
- After Converting or reducing MPCP to PCP
  - The lists will be in the form of

$$y_0\ y_{i1}\ y_{i2}\ \ldots\ldots y_{im} y_{k+1} = z_0\ z_{i1}\ z_{i2}\ \ldots\ldots z_{im} z_{k+1}$$

So it fallows that

$$w_{i1}\ w_{i2}\ \ldots\ldots\ w_{im} = x_{i1}\ x_{i2}\ \ldots\ldots\ x_{im}$$

PCP is Undecidable !!

Statements

- PCP is decidable iff MPCP is decidable.
- MPCP have a solution iff PCP have a solution.
- MPCP is decidable iff L_U is recursive, where L_U is a language <M,w> M is Turing machine accepts W string => A_{TM} is decidable. (Assumption)

Proof:

First Reducing L_U to MPCP. That is, given a pair (M, w), we construct an instance (A,B) of MPCP such that TM M accepts input w iff (A,B) has a solution.

Let $M=(Q, \Sigma,\Gamma,\delta,q_0,B,F)$ be a TM . Let w in Σ^* be an input string.

Construction of MPCP as fallows.

List A	List B	
#	$\#q_0w\#$	First Pair
X	X	Tape Symbols and seperator
#	#	
qX	Yp	if δ(q,X)= (p,Y,R) — Simulate
ZqX	pZY	if δ(q,X)= (p,Y,L); Z is any type symbol — Simulate
q#	Yp#	if δ(q,B)= (p,Y,R) — Simulate
Zq#	pZY#	if δ(q,B)= (p,Y,L); Z is any tape symbol — Simulate

Proof : (Contd.)

- If the ID at the end of the B string has an accepting state .
- Thus, if q is an accepting state, then for all tape symbols X and Y, there are pairs:

List A	List B
XqY	Q
Xq	Q
qY	q

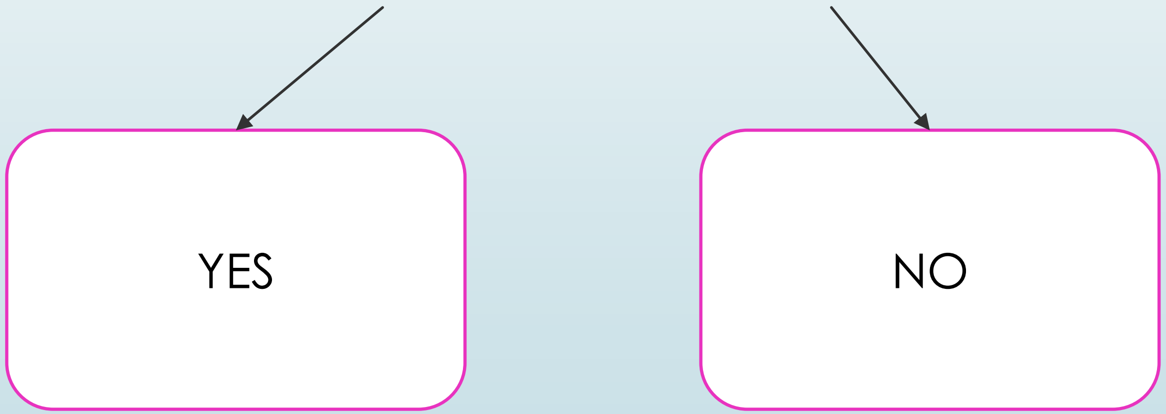
- Finally, once the accepting state has consumed all tape symbols, it stands alone as the last ID on the B string.

List A	List B
q##	#

} Acceptance State

Contd.

- From given <M, w> we constructed an instance of MPCP => If M accepts W we will have solution if not there is no solution.
- Suppose, MPCP is decidable, then what you can do is ???



M Accepts w

M does not accept w

THESE IS CONTRIDICTION

Conclusion

- If M does not accept w you will not go to final state you may halt in that case consuming of symbols will not be possible and you will not get equal strings.
- If it gets into a loop that will be a partial solution that means no final state it goes on and on . You will not get equal strings.
- So, M accepts w iff the instance of MPCP has a solution.

Conclusion (Contd.)

- As long as M does not enter final state it remains in partial solution. The string in list B is always greater than A String. Thus if there is a solution at some point M must enter acceptance state, i,.e.; M accepts w. (Contradiction)
- As we already know that A_{TM} is Undecidable i.e.; MPCP is also Undecidable.
- As MPCP is Undecidable PCP is also Undecidable.

References:

- http://en.wikipedia.org/wiki/Post_correspondence_problem
- *Introduction to Automata Theory, Languages, and Computation*- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman
- http://www.slideshare.net/ThamerAlamery/theory-of-computation-presentation-final

Video References:

- https://www.youtube.com/watch?v=YSr5zmVqZLI
- https://www.youtube.com/watch?v=AEkYcf9nQpg
- https://www.youtube.com/watch?v=qRhQpvFJmf8

Thank You

Bharath Bhushan Reddy Goulla