O Problema da Correspondência de Post

Gabriel de Melo Osório, Henrique Shiguemoto Felizardo, Matheus Augusto Schiavon Parise

Departamento de Informática - Universidade Estadual de Maringá

Paraná, Brasil.

ra107862@uem.br, ra115207@uem.br, ra107115@uem.br

Abstract. *This article seeks to describe an undecidable problem, the problem of post correspondence. During the break down of the document, we will establish a definition of decidable problems and present an instance of the post matching problem. We will also discuss, through a proof taken from the book[1], why it is undecidable.*

Resumo. *O presente artigo busca descrever um problema indecidível, o problema da correspondência de post. Durante o destrinchar do documento, estabeleceremos uma definição de problemas decidíveis e apresentaremos uma instância do problema de correspondência de post. Também discorreremos por meio de uma prova extraída do livro[1] o porquê de ele ser indecidível.*

Introdução

Problema da Parada

O problema da parada consiste em descobrir a existência de um algoritmo para identificar se uma máquina universal qualquer M pára ao ler uma palavra P sobre o alfabeto de entrada, aceitando ou rejeitando-a. O problema da parada é um problema de decisão não decidível, cuja demonstração dá-se pelo princípio da redução ao absurdo.

> *"Dada uma máquina universal M qualquer e uma palavra W qualquer sobre o alfabeto de entrada, existe um algoritmo que verifique se M pára, aceitando ou rejeitando, ao processar a entrada W?"* - MENEZES, P. B

Tese de Church-Turing

A tese em questão foi proposta por Alonzo Chruch e Alan Turing e possui vários enunciados equivalentes, o que iremos usar nesse trabalho é o enunciado que diz respeito à decidibilidade de problemas algorítmicos (Se um problema é decidível, isto é, se um problema possui resposta sim ou não, ou não decidível). Segue o enunciado da Tese de Church-Turing:

> "*Se um problema de decisão tem solução, então existe uma Máquina de Turing que o soluciona*" - Alonzo Church, Alan Turing.

Problemas Decidíveis

Problemas decidíveis são aqueles em que uma Máquina de Turing pode ter uma saída binária (sim ou não, verdadeiro ou falso, 0 ou 1, etc) para todas as instâncias de entrada do problema. Porém, um problema decidível por si só não está expressado em uma linguagem que uma Máquina de Turing possa processar, ou seja, deve haver uma transformação da instância de entrada para o problema decidível para uma palavra de uma linguagem recursiva reconhecível por uma Máquina de Turing.

Exemplos de problemas decídíveis:

Problema de determinar se um número pertencente ao conjunto dos números natural (N) é par:

> *As instâncias de entrada são os números que queremos verificar se é par ou ímpar (não par). Podemos transformar essas entradas em palavras w que pertence à {0,1,2,3,4,5,6,7,8,9}* (w é qualquer número natural). E podemos construir uma Máquina de Turing que reconhece palavras da linguagem Σ = {w / w é par}. Dessa forma, a Máquina pode ser construída para checar se o último símbolo de w pertence ao conjunto {0,2,4,6,8}, se esse for o caso, então a resposta é sim, caso contrário a resposta é não.*

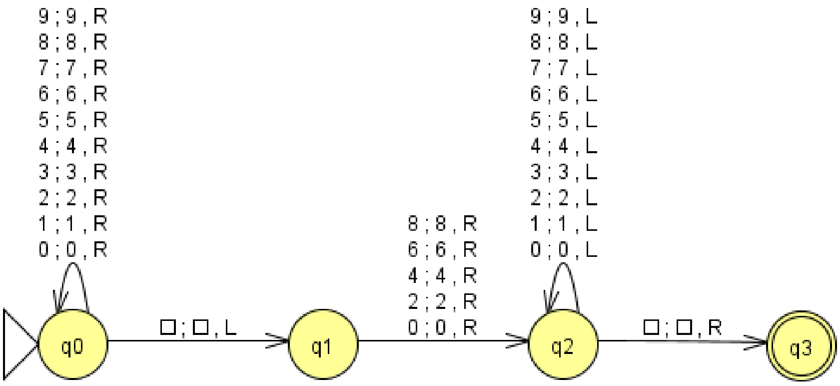


Figura 1 - Máquina de Turing Reconhecedora do Problema Acima, note que nesse exemplo de máquina estamos colocando o cabeçote no início novamente. Feito com JFLAP.

O Problema de Correspondência de Post

O problema de correspondência de Post, também chamado de PCP, é um problema matemático proposto por Emil Post, este é um problema de decisão que pertence a uma classe de problemas algoritmos o qual não existe uma solução genérica, ou seja, não existe algoritmo que resolva esse problema até o momento, logo ele é indecidivel, ou seja, não há Maquina de Turing que dá resposta sim ou não para todas as instâncias possíveis do problema. O problema se trata sobre a pergunta, dado alguns pares de símbolos P_1 até Pn, onde x é o primeiro símbolo e y o segundo, é possível ordenar esses pares de forma que a sequência combinada de todos os x's de X_1 até X_n seja igual a combinação de todos os y's de Y_1 até Y_n ?. Para deixar mais claro vamos utilizar um exemplo, imagine um conjunto de peças de domino em que cada lado do dominó possui um símbolo, este símbolo pode ser um número ou uma string, por conveniência vamos usar as combinações a seguir:

Combinações	1	2	3	4
X	b	a	ca	abc
Y	ca	ab	a	c

Se uma sequência for escolhida, como por exemplo 1-2-3 o resultado será a combinação de todos os X's e Y's na ordem que foi definida da esquerda para direita, X será $X_1 + X_2 + X_3$, ou seja, "b" + "a" + "ca" = "baca", para Y será $Y_1 + Y_2 + Y_3$, ou seja, "ca" + "ab" + "a" = "caaba"

Sequência	1 - 2 - 3
X	baca
Y	caaba

Observe que neste caso há erro, pois a sequência formada por X e Y difere em seus valores, mas se escolhermos uma sequência como 2-1-3-2-4 podemos obter o resultado desejado:

Sequência	2 - 1 - 3 - 2 - 4
X	abcaaabc
Y	abcaaabc

Infelizmente existem casos mais complexos que esse no qual a solução é muito extensa, ou ainda que não tem solução ou até que não é possível determinar uma decisão, no exemplo a seguir a combinação requer uma sequência com no mínimo 206 pares para resolver o problema:

Combinações	1	2	3	4
X	1000	01	1	00
Y	0	0	101	001

Este exemplo não possui solução, pois você é obrigado a começar pela combinação 1, e após a combinação 1 fica inviável escolher a combinação 2, a terceira e última alternativa a combinação 3 é incapaz de suprir a necessidade pois sempre faltará o "1" no final de Y:

Combinações	1	2	3
X	10	011	101
Y	101	11	011

Este próximo exemplo é uma correspondência de post o qual não é possível determinar uma solução, ou seja, indecidivel:

Combinações	1	2	3
X	10	0	001
Y	0	001	1

É possível inferir a partir dos exemplos que, se uma instância do problema de correspondência de Post possui solução, então a mesma instância possui infinitas soluções. Essa conclusão é facilmente provável, pois podemos usar uma das soluções qualquer e concatená-la com ela mesma infinitas vezes. Por exemplo, no exemplo mostrado acima, há mais uma solução "2-1-3-2-4 concatenada com 2-1-3-2-4", e pode-se concatenar infinitas vezes.

Prova da Indecidibilidade do PCP

A prova em questão será baseada na prova do livro [1].

Uma outra maneira de definir o problema de correspondência de Post é definir um sistema conhecido como Sistema de Post, que nada mais é que uma formalização para os Pares de símbolos P_1 até Pn mencionados acima. Segue a definição de Sistema de Post:

> *Um conjunto finito e não-vazio de pares ordenados S é um Sistema de Post definido em Σ se os pares ordenados são compostos de palavras de um alfabeto Σ.*

Ou seja, um Sistema de Post é um conjunto do tipo S = {(x1,y1), (x2,y2), ..., (xn, yn)} tal que n > 1 e xi,yi pertencem à Σ* para i em {1, 2,..., n}.

Uma solução para S é uma sequência não-vazia de pares de números de {1,2,...,n}: i1, i2, ..., ik tal que xi1, xi2, ..., xik = yi1, yi2, ..., yik.

Com essas definições de Sistema de Post e de solução de um Sistema de Post, podemos definir novamente o Problema de Correspondência de Post da seguinte forma:

> *A partir de um algoritmo que analisa Sistemas de Post, determine se há uma solução para esse sistema.*

Que ainda é um problema de decisão, pois um Sistema de Post sempre possui pelo menos uma solução ou não possui nenhuma solução.

A partir dessa nova definição, podemos realizar algumas conclusões sobre os casos em que não há soluções. Como por exemplo, se para todo i em {1,2,...,n} o primeiro símbolo de xi é diferente do primeiro de yi então o Sistema de Post não possui solução. Analogamente, podemos concluir o mesmo para os últimos símbolos de xi e yi. Outro exemplo é o caso em que para todo i em {1,2, ..., n} xi é maior que yi, pois assim sempre a palavra xi1, xi2,...,xik será diferente de yi1, yi2,...,yik. Analogamente, podemos ter a mesma conclusão para o caso em que xi é menor que yi.

Prova

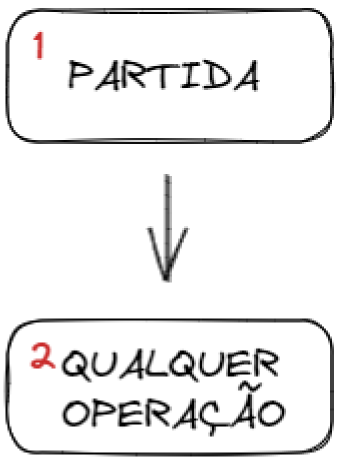
A partir de uma Máquina de Post qualquer sobre o alfabeto Σ e de uma palavra w pertencente à Σ***,** podemos construir um Sistema Normal de Post baseado na sequência de comandos executados por essa Máquina de Post para a entrada w, tal que o Sistema Normal de Post construído tenha solução se e somente se Máquina de Post parar enquanto processa a palavra w.

Para cada ação realizada na fila X na Máquina de Post, um par será construído e adicionado no Sistema Normal de Post.

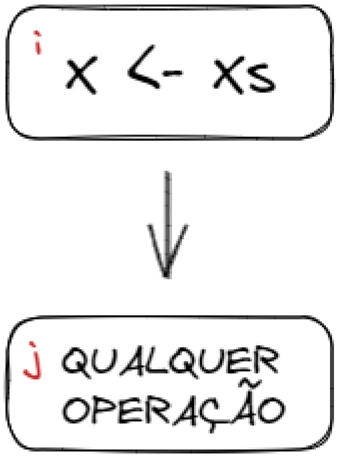
Algumas suposições:

- Suponhamos que w é uma palavra qualquer de Σ*, em que w = a1,a2,a3,...,an.
- Inicialmente a fila X tem a palavra w
- As componentes da Máquina de Post serão enumeradas com números naturais de {1,2,...,m}. Por exemplo, o componente PARTIDA pode ser enumerado com o número 1.

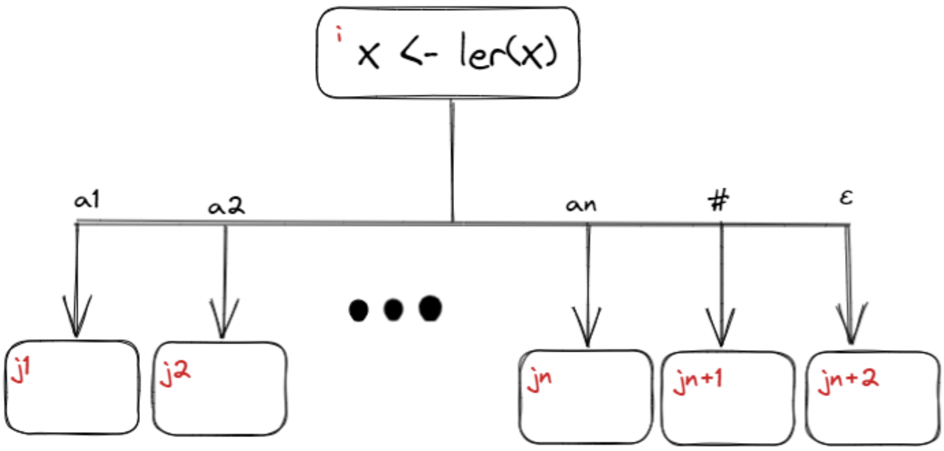
Regras para construção de pares para o Sistema Normal de Post:



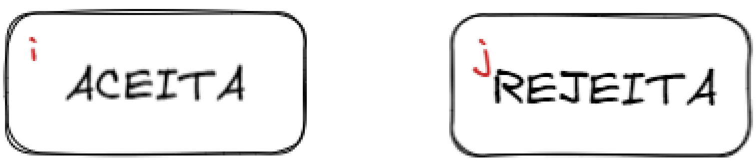
A primeira regra para construção de par é para o começo do diagrama da Máquina de Post, temos que o par construído que será adicionado no Sistema Normal de Post será (1, 1 a1a2...an 2).



A segunda regra se diz respeito à operação de escrita no diagrama da Máquina de Post. Após uma escrita na fila X, o par adicionado ao Sistema Normal de Post será (i, s j).



Para os casos de leitura, será adicionado um par para cada caminho que pode ser seguido depois da leitura. Os pares adicionados serão (ia1, j1), (ia2, j2), ..., (ian, jn), (i#, jn+1) e (i ε, jn+2).



Para os casos de parada adicionados os pares (i, ε) (para o exemplo de aceitar acima). No caso do REJEITA acima seria (j, ε).

Ultimamente, para cada símbolo s em Σ U {#} adicionamos mais pares no Sistema Normal de Post. Os pares adicionados serão (s, s).

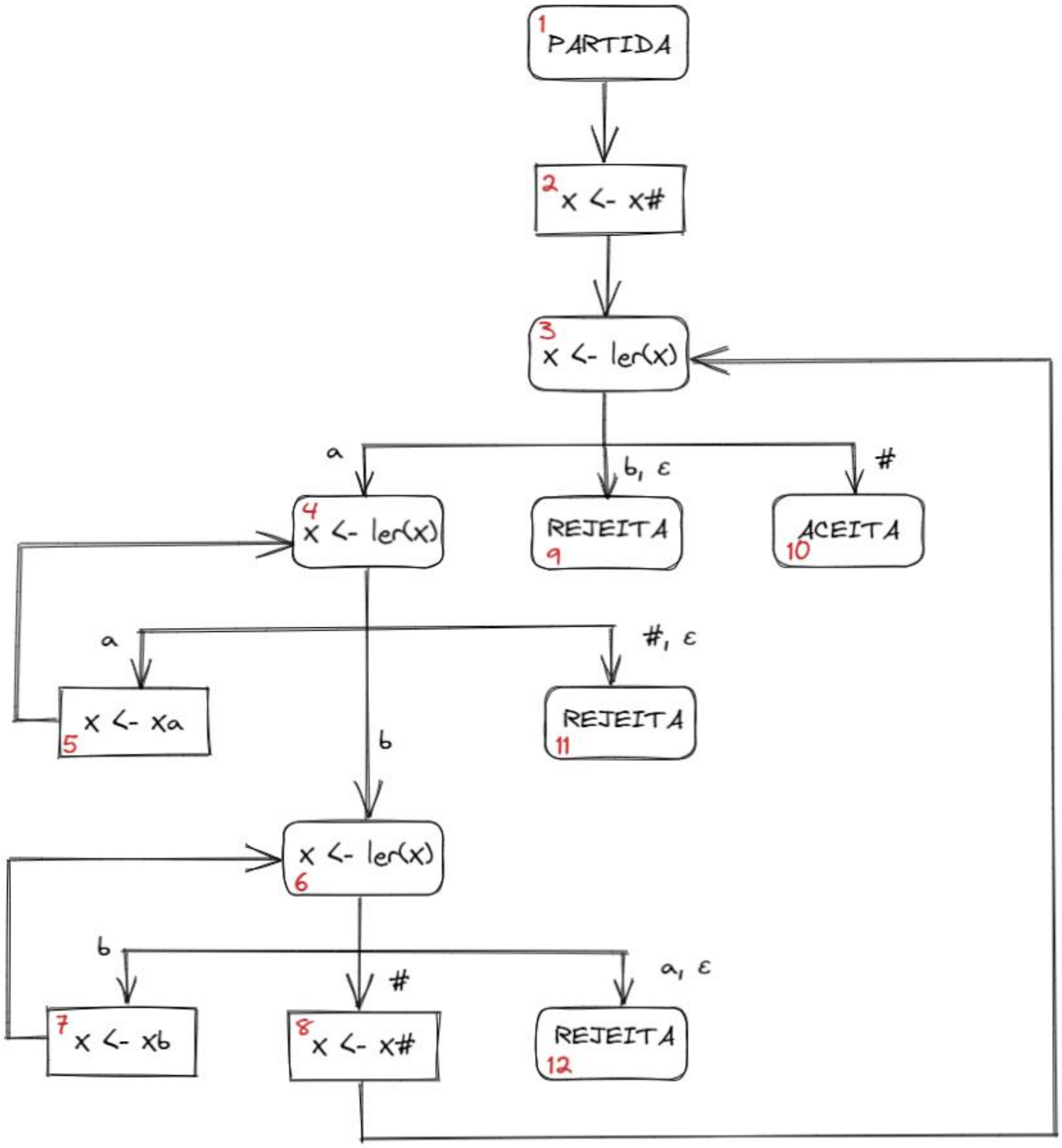
Note que o par adicionado na PARTIDA possui um símbolo a mais no segundo componente, essa diferença de tamanho é compensada no componente da Máquina de Post ACEITA ou REJEITA, dependendo de onde a máquina irá parar. Isso significa que o Sistema Normal de Post tem solução se e somente se a Máquina de Post parar em algum momento para a entrada w. Dessa forma, é possível concluir que o Problema da Parada é redutível ao Problema de Correspondência de Post. E como o Problema da Parada é indecidível, temos que o Problema de Correspondência de Post também é indecidível.

Exemplo da Prova

Para facilitar o entendimento da prova, abaixo será representada um exemplo de uso das regras de construção dos pares que serão adicionados no Sistema Normal de Post e a própria Máquina de Post em si. Dessa forma podemos mostrar as enumerações dos componentes da Máquina de Post e os pares que são adicionados no Sistema Normal de Post.

O exemplo em questão é baseado no mesmo exemplo do livro [1].

Consideremos a Máquina de Post para linguagem L = $\{a^n b^n, n >= 0\}$ (Duplo-Balanceamento). Segue a Máquina de Post dessa linguagem com os componentes enumerados de acordo com a prova.



Note que temos 12 componentes (representados pelos retângulos) no diagrama todo. A enumeração dos componentes começa no componente PARTIDA que é enumerado com 1, enquanto que o componente REJEITA mais abaixo do diagrama recebe o número 12. Os números estão em vermelho para facilitar a visão da enumeração, já que é muito importante para a prova do indecidibilidade do PCP.

Considere que nossa palavra inicial é w = ab. Ou seja, X inicialmente é uma fila que contém a palavra ab dentro dela. A partir desse diagrama, podemos criar o nosso Sistema Normal de Post seguindo as regras para construção de pares. Temos que o Sistema Normal de Post possui os seguintes pares:

Pares de S:

1 - (1, 1ab2) (componente de PARTIDA)

2 - (2, #3) (componente 2 de escrita para o componente 3)

3 - (3a, 4) (componente 3 de leitura para o componente 4)

4 - (3b, 9) (componente 3 de leitura para o componente 9)

5 - (3#, 10) (componente 3 de leitura para o componente 10)

6 - (3ε, 9) (componente 3 de leitura para o componente 9)

7 - (4a, 5) (componente 4 de leitura para o componente 5)

8 - (4b, 6) (componente 4 de leitura para o componente 6)

9 - (4#, 11) (componente 4 de leitura para o componente 11)

10 - (4ε, 11) (componente 4 de leitura para o componente 11)

11 - (5, a4) (componente 5 de escrita para o componente 4)

12 - (6a, 12) (componente 6 de leitura para o componente 12)

13 - (6b, 7) (componente 6 de leitura para o componente 7)

14 - (6#, 8) (componente 6 de leitura para o componente 8)

15 - (6ε, 12) (componente 6 de leitura para o componente 12)

16 - (7, b6) (componente 7 de escrita para o componente 6)

17 - (8, #3) (componente 8 de escrita para o componente 3)

18 - (9, ε) (componente REJEITA mais acima do diagrama com número 9)

19 - (10, ε) (componente ACEITA mais acima do diagrama com número 10)

20 - (11, ε) (componente REJEITA no meio do diagrama com número 11)

21 - (12, ε) (componente REJEITA mais abaixo do diagrama com número 12)

22 - (a, a) (par adicionado para o símbolo a)

23 - (b, b) (par adicionado para o símbolo b)

24 - (#, #) (par adicionado para o símbolo #)

Como sabemos que w pertence a linguagem L, quer dizer que a Máquina de Post irá parar em algum momento. E a partir da prova anterior, isso significa que o Sistema construído possui pelo menos uma solução, que implica que temos infinitas soluções.

Note que apenas o primeiro par depende da entrada w, se a entrada w for diferente, a solução para o Sistema Normal de Post construído muda.

Uma observação importante é que não conseguimos a partir da máquina de post saber qual que é a solução por si só, Apenas iremos saber se o Sistema possui pelo menos uma solução ou não, já que PCP é um problema de decisão.

Referências

[1] MENEZES, P. B.; DIVERIO, T. A.; Teoria da Computação: Máquinas Universais e Computabilidade; 3a edição Bookman 2011

[2] TEORIA da Computação - Decidibilidade. Maringá: Uem, 2017. P&B. Disponível em: Teoria da Computação - Decidibilidade

[3] LFA.AULA24.O problema da correspondência de Post. São Carlos: Prof. Dr. Daniel Lucrédio, 2017. P&B. Disponível em: https://www.youtube.com/watch?v=N9EUpXsWdzQ