UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA

LAURO MANOEL LIMA DA GAMA

RESOLUÇÃO DE SUDOKU

Manaus
2014

LAURO MANOEL LIMA DA GAMA

RESOLUÇÃO DE SUDOKU

Implementação e analise de um programa solucionador de jogos matemáticos do tipo sudoku utilizando Matlab.

Professor: Victor Enrique Vermehren Valenzuela

Manaus
2014

LISTA DE ABREVIATURAS E SIGLAS

EST	Escola Superior de Tecnologia
UEA	Universidade do Estado do Amazonas

SUMÁRIO

INTRODUÇÃO

O jogo sudoku é um dos mais populares passatempos matemáticos de todos os tempos. O seu objetivo é o preenchimento com números de uma matriz de 9x9 elementos de forma que cada linha, coluna e submatriz de 3x3 elementos contenha todos os dígitos entre 1 e 9.

O jogo tem origens na França em meados de 1895 e sua forma atual foi introduzida no japão pela editora Nikoli em abril de 1984 com o nome *Suji wa dokushin ni kagiru*. O nome foi posteriormente abreviado para sudoku por Maki Kaji.(1)

A resolução desse jogo pode ser feita por um computador através de recursividade, onde o programa irá sugerir um numero para uma posição vazia do sudoku e verifica se essa escolha é valida. Caso seja valida o programa continuará, caso não seja ele irá tentar outro numero até encontrar um que seja valido para aquela posição. Dessa forma o programa irá preencher todas as posições até que o jogo esteja completo ou seja deduzido que não há soluções validas e o jogo foi formulado de forma incorreta.(2)

1 DESENVOLVIMENTO

O programa desenvolvido em Matlab utiliza os princípios de recursividade para preencher os espaços vazios de uma matriz 9x9. Ao detectar que a matriz não possui espaços vazios o sudoku é considerado resolvido.

O programa completo está descrito abaixo:

```
function S = sudoku(M,S)
if ~exist('S','var')
    S = zeros([size(M),0]);
end
firstId = find(M(:)==0, 1 );
if isempty(firstId)
    S(:,:,size(S,3)+1) = M;
else
    [i,j] = ind2sub([9,9],firstId);
    for k=1:9
        ii = (ceil(i/3)-1)*3+1;
        jj = (ceil(j/3)-1)*3+1;
        mm = M(ii:ii+2,jj:jj+2);
        if sum(M(i,:)==k)==0 && sum(M(:,j)==k)==0 && sum(mm(:)==k)==0
            M(i,j) = k;
            S = sudoku(M,S);
        end
    end
end
```

A primeira linha do programa é a definição da função sudoku. Essa função recebe dois parâmetros M e S. O parâmetro M é a matriz 9x9 que representa o sudoku não resolvido e o parâmetro S é o resultado da função, portanto a função recebe ela mesma como parâmetro.

```
function S = sudoku(M,S)
```

Ao ser iniciada, a função sudoku verifica se a variável S existe. Em caso negativo essa é a primeira vez que a função está sendo executada em um laço recursivo e a variável S deve ser inicializada como uma matriz nula do mesmo tamanho da matriz M.

```
if ~exist('S','var')
    S = zeros([size(M),0]);
end
```

Na linha 5 a função sudoku usa o comando *find*(3) para encontrar o índice da primeira célula da matriz M com valor igual a zero.

```
1 firstId = find(M(:)==0, 1 );
```

Caso não haja elementos iguais a zero o sudoku está completamente preenchido e a solucao do jogo foi encontrada e o resultado é salvo na matriz S.

```
1 if isempty(firstId)
2     S(:,:,size(S,3)+1) = M;
```

Caso haja elementos com valor zero a função irá tentar encontrar todos os valores que podem ser inseridos nessa célula. Esse processo começa utilizando o comando *ind2sub*(3) para encontrar as coordenadas de linha e coluna da célula *M(i,j)*.

```
1 [i,j] = ind2sub([9,9],firstId);
```

De posse desse índice, cria-se um laço para testar os possíveis números validos para a célula *M(i,j)*.

```
1 for k=1:9
2     ii = (ceil(i/3)-1)*3+1;
3     jj = (ceil(j/3)-1)*3+1;
4     mm = M(ii:ii+2,jj:jj+2);
5     ...
6 end
```

No código acima a variável *mm* recebe os índices da submatriz 3x3 em que a célula *M(i,j)* está localizada. A partir desses índices é verificado se o valor k atende os requisitos do sudoku. Caso atenda então a célula *M(i,j)* é preenchida com o valor k e a função sudoku é chamada recursivamente de modo que esse processo se repetirá até que seja encontrada uma solução para o jogo.

```
1 if sum(M(i,:)==k)==0 && sum(M(:,j)==k)==0 && sum(mm(:)==k)==0
2             M(i,j) = k;
3             S = sudoku(M,S);
4     end
```

REFERÊNCIAS

1 PEEG ED, J. *Ed Pegg Jr.'s Math Games: Sudoku Variations*. 2006. Disponível em: <http://www.mathpuzzle.com/MAA/41-Sudoku%20Variations/mathgames_09_05_05.html>. Citado na página 4.

2 WEEKS, M. *Digital Signal Processing using MATLAB and wavelets*. [S.l.: s.n.]. ISBN 0977858200. Citado na página 4.

3 MATHWORKS. 2015. Disponível em: <http://www.mathworks.com/help/matlab/>. Citado na página 6.