Ficha de laboratório Nº 7: Apoio ao 1º Projeto

Inteligência Artificial - Escola Superior de Tecnologia de Setúbal

Prof. Joaquim Filipe
Eng. Filipe Mariano

Nota prévia

Os exercícios a desenvolver no âmbito da presente série de exercícios requerem uma leitura prévia do enunciado do 1º projeto. Se porventura ainda não o leram, devem fazê-lo antes de realizar o laboratório.

Objetivos da ficha

Este laboratório tem como objetivo desenvolver algumas das funções necessárias para o 1º projeto de procura em espaço de estados. Ao longo do mesmo, irá implementar:

- Seletores
- Funções Auxiliares
- Operadores

Os exercícios propostos neste laboratório devem servir de base para a resolução da 1ªfase do projeto. Contudo, não é necessário que a abordagem que cada grupo siga seja exatamente a mesma da que é proposta neste laboratório.

1. Representação do Problema

Gerar um tabuleiro do puzzle *Blokus Uno*

1. Descarregar o ficheiro **laboratorio7.lisp** disponível no Moodle.
2. Abrir o ficheiro no IDE *LispWorks*.
3. Observar a estrutura de comentários presente no inicio do ficheiro para indicar o seu conteúdo e autor.
4. Neste ficheiro está definida a função `tabuleiro-vazio` que retorna um tabuleiro a utilizar no ***Blokus Uno***.
5. O tabuleiro retornados servirá de exemplo para os exercícios propostos neste laboratório. É um tabuleiro de acordo com as regras do enunciado do projeto, de dimensão 14 x 14.
6. Compilar o código do ficheiro e executar as funções no *Listener*.

2. Exercícios

Seletores

1. `linha`: Função que recebe um índice e o tabuleiro e retorna uma lista que representa essa linha do tabuleiro.

```
CL-USER > (linha 0 (tabuleiro-vazio))
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

2. `coluna`: Função que recebe um índice e o tabuleiro e retorna uma lista que representa essa coluna do tabuleiro.

```
CL-USER > (coluna 0 (tabuleiro-vazio))
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

3. `celula`: Função que recebe dois índices (linha e coluna) e o tabuleiro e retorna o valor presente nessa célula do tabuleiro.

```
CL-USER > (celula 0 1 (tabuleiro-vazio))
0
```

Funções auxiliares

4. `casa-vaziap`: Função que recebe dois índices e o tabuleiro e verifica se a casa correspondente a essa posição se encontra vazia, ou seja, igual a 0.

```
CL-USER > (casa-vaziap 0 0 (tabuleiro-vazio))
T
```

5. `verifica-casas-vazias`: Função que recebe o tabuleiro e uma lista de pares de índices linha e coluna e devolve uma lista com `T` ou `NIL` caso se encontrem vazias.

```
CL-USER > (verifica-casas-vazias (tabuleiro-vazio) '((0 0) (1 1)))
(T T)
```

6. `substituir-posicao`: Função que recebe um índice, uma lista e um valor (por *default* o valor é 1) e substitui pelo valor pretendido nessa posição.

```
CL-USER > (substituir-posicao 0 (linha 0 (tabuleiro-vazio)))
(1 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

7. `substituir`: Função que recebe dois índices, o tabuleiro e um valor (por *default* o valor é 1). A função deverá retornar o tabuleiro com a célula substituída pelo valor pretendido. Utilize a função `substituir-posicao` definida anteriormente.

```
CL-USER > (substituir 0 0 (tabuleiro-vazio))
((1 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

```
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0)
 (0 0 0 0 0 0 0 0 0 0 0 0 0 0))
```

Operadores

Para realizar os operadores é necessário recordar os diferentes tipos de peças existentes no puzzle, sendo que uma delas possui duas posições diferentes.

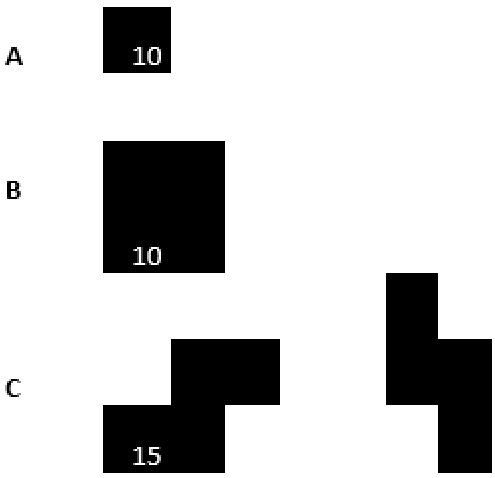

Figura 1: Indicação dos diferentes tipos de peças e que passarão a ser denominados de `peca-a`, `peca-b`, `peca-c-1` e `peca-c-2`.

8. `peca-casas-ocupadas`: Função que recebe dois índices e um tipo de peça (peca-a, peca-b, peca-c-1 ou peca-c-2) e retorna uma lista com os pares de índices correspondentes às posições em que irá ser colocada a peça. Para todas as peças vamos assumir que os índices passados como argumento para a função serão referentes à casa do canto superior esquerdo. No caso do esse-lado será a primeira casa a contar da esquerda o ponto de referência.

```
CL-USER > (peca-casas-ocupadas 1 1 'peca-a)
((1 1))

CL-USER > (peca-casas-ocupadas 1 1 'peca-b)
((1 1) (1 2) (2 1) (2 2))

CL-USER > (peca-casas-ocupadas 1 1 'peca-c-1)
((1 1) (2 1) (2 2) (3 2))
```

```
CL-USER > (peca-casas-ocupadas 1 1 'peca-c-2)
((1 1) (1 2) (0 2) (0 3))
```

9. peca-a: Função que recebe dois índices e o tabuleiro e coloca um quadrado de 1x1 no tabuleiro. A função deverá retornar NIL caso a casa pretendida esteja preenchida ou caso a peça esteja a ser colocada fora dos limites do tabuleiro.

```
CL-USER > (peca-a 1 1 (tabuleiro-vazio))
((0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 1 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0))
```

10. peca-b: Função que recebe dois índices e o tabuleiro e coloca o quadrado 2x2 no tabuleiro tendo como ponto de referência o índice passado como argumento. A função deverá retornar NIL caso alguma das casas a serem preenchidas não esteja vazia ou caso a peça esteja a ser colocada fora dos limites do tabuleiro.

```
CL-USER > (peca-b 1 1 (tabuleiro-vazio))
((0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 1 1 0 0 0 0 0 0 0 0 0 0 0)
(0 1 1 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0))
```

11. peca-c-1: Função que recebe dois índices e o tabuleiro e coloca a peça esse na posição de lado no tabuleiro tendo como ponto de referência o índice passado como argumento. A função deverá retornar

`NIL` caso alguma das casas a serem preenchidas não esteja vazia ou caso a peça esteja a ser colocada fora dos limites do tabuleiro.

```
CL-USER > (peca-c-1 1 1 (tabuleiro-vazio))
((0 0 1 1 0 0 0 0 0 0 0 0 0 0)
(0 1 1 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0))
```

12. `peca-c-2`: Função que recebe dois índices e o tabuleiro e coloca a peça esse na posição para cima no tabuleiro tendo como ponto de referência o índice passado como argumento. A função deverá retornar `NIL` caso alguma das casas a serem preenchidas não esteja vazia ou caso a peça esteja a ser colocada fora dos limites do tabuleiro.

```
CL-USER > (peca-c-2 1 1 (tabuleiro-vazio))
((0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 1 0 0 0 0 0 0 0 0 0 0 0 0)
(0 1 1 0 0 0 0 0 0 0 0 0 0 0)
(0 0 1 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0)
(0 0 0 0 0 0 0 0 0 0 0 0 0 0))
```

Nota: Nas funções de colocação das peças presentes neste laboratório ainda não está a ser contemplada a verificação se a peça está a ser colocada nalgum vértice de uma casa preenchida anteriormente, pelo que mais tarde terão de incorporar esse tipo de verificação nas funções já desenvolvidas ou numa função construída para gerir a colocação das várias peças.