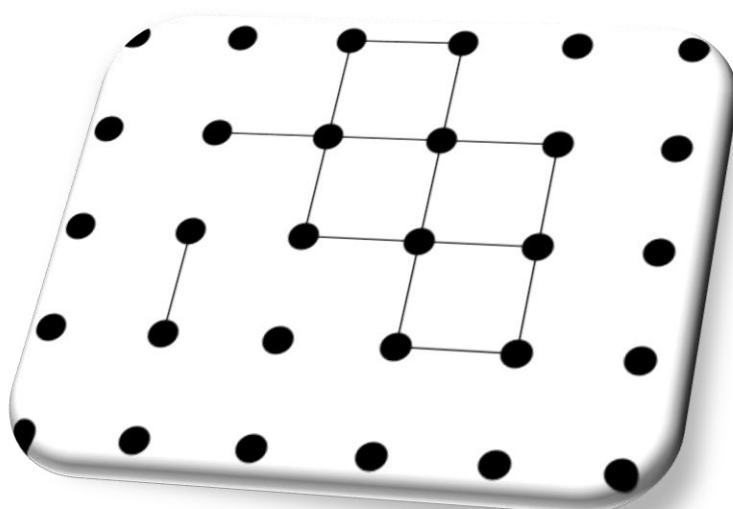


Manual Técnico

Inteligência Artificial 2016/2017



Problema do Puzzle dos Pontos e das Caixas

Professores:

Joaquim Filipe
Cédric Grueau

Turma: 3º-Inf-ES2

140221017 – André Bastos
140221002 – Luís Mestre

Índice

1.	Introdução	3
2.	Objetos que compõem o projeto	4
2.1.	dots-boxes.lisp	4
2.1.1.	Variáveis globais	4
2.1.2.	Escrita das estatísticas	4
2.1.3.	Jogo	4
2.1.4.	Auxiliares ao alfabeto	5
2.1.5.	Operadores	5
2.1.6.	Construtor	6
2.1.7.	Métodos seletores	6
2.2.	alfabeto.lisp	6
2.2.1.	Algoritmo Alfabeto	6
3.	Função de avaliação	7
4.	Testes ao algoritmo	7
4.1.	1º Teste: Tabuleiro 7x7 vazio	7
4.2.	2º Teste: Tabuleiro 3x3 meio cheio	8
4.1.	3º Teste: Tabuleiro 3x3 cheio	8

1. Introdução

Este documento tem a finalidade de informar o utilizador dos aspetos mais técnicos do projeto. Tendo assim como tópicos o algoritmo geral utilizado, a descrição dos objetos que compõem o projeto (dados e procedimentos), identificação das limitações e opções técnicas. Tem também análises críticas de resultados das execuções do programa, análises comparativas do conjunto de execuções do programa para cada problema.

Como este projeto é em conjunto com a UC de Computação Móvel, foi-nos solicitado pelos docentes a separação do projeto em 2 ficheiros (dots-boxes.lisp e alfabet.lisp) diferentes para separar o algoritmo alfabet do problema. Sendo a dots-boxes.lisp destinada aos objetos que fazem parte da implementação do problema e o alfabet.lisp tem a implementação do algoritmo.

Este manual destina-se a utilizadores que compreendam minimamente a linguagem Lisp, visto que é a linguagem que foi sugerida para a implementação do projeto.

2. Objetos que compõem o projeto

2.1. dots-boxes.lisp

2.1.1. Variáveis globais

- ***jogada***
- ***peca-maquina***
- ***cortes-alfa***
- ***cortes-beta***
- ***nos-explorados***

2.1.2. Escrita das estatísticas

Estes vão estar em comentário pois são métodos usados para gerar os testes e nada mais.

- **escreve**
- **escreve-ecra-ficheiro**

2.1.3. Jogo

- **jogar**
- **ver-prof-max**
- **ver-nr-nils**
- **ver-nr-nils-arcos**
- **ver-nr-nils-lista**
- **reset-cortes**
- **peca-jogada-atual**
- **jogador-jogada-atual**
- **ordena-sucessores**
- **numero-caixas-fechadas**
- **limite-numero-caixas-fechadas**
- **tabuleiro-preenchido-p**
- **tira-primeiras**
- **numero-caixas-fechadas-horizontal**

- ver-caixa-fechada
- ver-se-1a-caixa-potencial
- ver-pos-lista-ligadas
- ver-pos-lista-ligadas-aux
- intersecao
- esta-no-conjunto
- posicoes-dos-sucessores
- lista-posicao
- posição
- sucessores-no
- sucessores-no-aux
- numero-caixas-quase-fechadas
- numero-caixas-quase-fechadas-horizontal
- ver-caixa-quase-fechada
- ver-se-1a-caixa-tem-possivel-quase-caixa
- inserir-na-lista

2.1.4. Auxiliares ao alfabeto

- alisa-sucessores
- vencedor-p
- avaliar-folha
- avaliar-folha-aux
- solucao
- avaliar-folha-limite
- avaliar-no

2.1.5. Operadores

- operadores
- arco-vertical
- arco-horizontal
- arco-aux
- arco-na-posicao
- arco-vertical-tabuleiro

- arco-horizontal-tabuleiro
- arco-aux-tabuleiro
- arco-na-posicao-tabuleiro

2.1.6. Construtor

- cria-no

2.1.7. Métodos seletores

- get-tabuleiro
- get-profundidade
- get-peca-maquina
- get-arcos-horizontais
- get-arcos-verticais
- get-tabuleiro-no
- get-numero-caixas
- get-numero-caixas-jogador1
- get-numero-caixas-jogador2
- get-elemento-lista

2.2. alfabeto.lisp

2.2.1. Algoritmo Alfabeto

- alfabeto
- jogadorMax
- jogadorMin

3. Função de avaliação

A função avaliar-folha-limite é a função de avaliação usada. Esta função tem em conta o número de caixas fechadas por cada jogador. Se o número for igual então devolve 0. Caso contrário, damos um peso de 1 às caixas fechadas pelo jogador 1 e um peso de 1.2 às caixas fechadas pelo jogador 2. Subtraímos um ao outro dependendo do tipo de jogador que é a máquina.

4. Testes ao algoritmo

Nesta secção o algoritmo vai ser testado.

4.1. 1º Teste: Tabuleiro 7x7 vazio

```
[[[null,null,null,null,null,null,null],[null,null,null,null,null,null,null],[null,null,
null,null,null,null,null],[null,null,null,null,null,null,null],[null,null,null,null,n
ull,null,null],[null,null,null,null,null,null,null],[null,null,null,null,null,null,n
ull],[null,null,null,null,null,null,null],[[null,null,null,null,null,null,null],[null,n
ull,null,null,null,null,null],[null,null,null,null,null,null,null],[null,null,null,n
ull,null,null],[null,null,null,null,null,null,null],[null,null,null,null,null,null,n
ull],[null,null,null,null,null,null,null],[null,null,null,null,null,null,null]]]
```

Resultado:

```
[[[null,null,null,null,null,null,null],[null,null,null,null,null,null,null],[null,null,n
ull,null,null,null,null],[null,null,null,null,null,null,null],[null,null,null,null,n
ull,null],[null,null,null,null,null,null,null],[null,null,null,null,null,null,null],[null,
null,null,null,null,null],[[null,null,null,null,null,null,null],[null,null,null,n
ull,null,null],[null,null,null,null,null,null,null],[null,null,null,null,null,null],
[null,null,null,null,null,null],[null,null,null,null,null,null],[null,null,null,
null,null,null,null],[null,null,null,null,null,null,2]]]
```

Chamada ao alfabeta

Cortes Alfa: 111

Cortes Beta: 110

Nos explorados: 12542

Como no início do jogo todas as jogadas são “iguais” em termos de vantagem, o algoritmo escolhe a última jogada considerada que será uma vertical no final do tabuleiro.

4.2. 2º Teste: Tabuleiro 3x3 meio cheio

Nestes próximos testes fazemos com um tabuleiro mais pequeno, pois da maneira como nos foi solicitado para fazer os sucessores, a gerarem os sucessores dos próximos quando fecham uma caixa existe uma explosão combinatória a meio. Para satisfazer a funcionalidade de quando se fecha uma caixa, joga-se outra vez.

```
[[[2,1,1],[null,1,null],[1,2,2],[null,1,null]],[[2,2,null],[null,null,null],[null,null,null],[null,null,null]]]
```

Resultado:

```
[[[2,1,1],[null,1,null],[1,2,2],[null,1,null]],[[2,2,null],[null,null,null],[null,null,null],[null,null,2]]]
```

Chamada ao alfabeto

Cortes Alfa: 10

Cortes Beta: 169

Nos explorados: 30612

4.1. 3º Teste: Tabuleiro 3x3 cheio

```
[[[2,2,null],[null,null,null],[null,null,null],[null,null,null]],[[2,1,1],[null,1,2],[1,2,2],[1,1,2]]]
```

Resultado:

```
[[[2,2,null],[null,null,null],[null,null,null],[null,null,2]],[[2,1,1],[null,1,2],[1,2,2],[1,1,2]]]
```


Chamada ao alfabeto

Cortes Alfa: 0

Cortes Beta: 0

Nos explorados: 2

5. Limitações

Se o tamanho do tabuleiro for grande (como um 7x7) é possível que o dê erro devido a flata de memória (stackOverflow), isto deve-se ao tamanho do tabuleiro e graças à funcionalidade do jogo de quando se fecha uma caixa joga-se de novo, pois ao termos feito nos sucessores para fechar múltiplas caixas, uma grande sobrecarga pode ser adicionada ao programa.