Manual Técnico

UC: Inteligência Artificial

Alunos:

- João Fernandes - 202100718
- Rodrigo Santos - 202100722

Docente: Joaquim Filipe

Índice

1. Introdução

No âmbito do primeiro projeto da Unidade Curricular (UC) de Inteligência Artificial (IA), do 1º semestre do 3º ano da Licenciatura em Engenharia Informática, foi solicitado aos alunos que implementassem um programa para resolução do problema Adji-boto, recorrendo um algoritmo de teoria de jogos.

O presente manual técnico tem como objetivo descrever a solução desenvolvida pelos alunos, bem como os detalhes da sua implementação.

2. Algoritmo

De forma a resolver os problemas, os alunos implementaram o algoritmo Negamax, usando uma abordagem funcional, recursiva e não-destrutiva. Os nós são avaliados com base na diferença de peças capturadas entre os dois jogadores. Os cortes realizados pelo algoritmo são sempre do tipo Beta.

```
(defun negamax (node depth jogador &optional (alpha most-negative-fixnum) (beta
most-positive-fixnum) (color 1))
  (if (or (zerop depth) (terminalp node))
      (* color (evaluate node jogador))
      (negamax-recursivo node depth jogador alpha beta color (or (remove nil
(sucessores node jogador)) (list node)))))

(defun negamax-recursivo (node depth jogador alpha beta color children)
  (if (null children)
      alpha
      (let* ((child (car children))
             (score (- (negamax child (1- depth) (alternar-jogador jogador) (-
```

```
beta) (- alpha) (- color)))))
        (incrementar-nos)
        (let ((new-alpha (max alpha score)))
          (if (>= new-alpha beta)
              (progn
                (incrementar-cortes)
                beta)
              (negamax-recursivo node depth jogador new-alpha beta color (cdr
children)))))))
```

3. Tipos Abstratos Utilizados

No presente projeto foi utilizado um tipo abstrato de dados para representar um nó, constituído pelo estado do problema em cada momento, juntamente com a pontuação de cada jogador. Este tipo abstrato possui o seguinte formato: `(((1 2 3 4 5 6)(1 2 3 4 5 6)) (0 0))`.

4. Limitações e Opções Técnicas

A solução desenvolvida pelos alunos possui uma limitação relacionada com memória do programa. Ao fornecer uma profundidade ao algoritmo superior a 4, é atingido o limite de memória stack durante o jogo.

5. Resultados

O algoritmo implementado pelos alunos resolve de forma eficaz e eficiente o jogo proposto. No entanto, poderia apresentar melhores resultados se não existisse a limitação de memória do IDE LispWorks.

6. Análise Estatística

O algoritmo Negamax desenvolvido pelos alunos, possui as seguintes médias numa execução contra um adversário humano:

- Número de nós analisados - 1022.37
- Número de cortes - 232.13
- Tempo gasto - 0.304 segundos

Para cada jogada, consultar o ficheiro log.dat.

7. Conclusão

Ao longo deste projeto, foi possível aplicar na prática os conhecimentos teóricos adquiridos na UC de IA, no que diz respeito aos algoritmos de teoria de jogos.

A implementação da solução para o problema Adji-boto em LISP proporcionou uma valiosa experiência de programação e um aprofundamento da compreensão dos conceitos fundamentais de IA. O desenvolvimento deste programa não só permitiu consolidar os conhecimentos sobre os algoritmos de teoria de jogos, mas também possibilitou a melhoria das competências de resolução de problemas e de pensamento lógico.

Em suma, os alunos implementaram com sucesso uma solução eficaz para o problema proposto, aplicando um algoritmo de teoria de jogos lecionadoo na UC de IA.