Jogos com informação completa determinísticos

João Matos (32409), João Rouxinol (44451) e André Rato (45517)

15 de maio de 2022

1 Nim

1.1 O problema

1.1.1 Estados

Para a representação do problema, os estados seguem todos a estrutura "`e(A, B, C, D)`", representando as filas do jogo.

1.1.2 Terminal

O predicado terminal é: `terminal(e(0, 0, 0, 0))`.

1.1.3 Função de utilidade

Como, no jogo do Nim, não existem empates, a função de utilidade apenas por tomar os valores 1 e -1.

```
% Utility function (1->win, 0->draw, -1->lose)
valor(B, 1) :- lines(B, o).
valor(B, 1) :- columns(B, o).
valor(B, 1) :- diagonals(B, o).

valor(B, -1) :- lines(B, o).
valor(B, -1) :- columns(B, o).
valor(B, -1) :- diagonals(B, o).

valor(_, 0).
```

Figura 1: Função de utilidade

1.2 Algoritmos

1.2.1 Minmax

Embora consideremos que a nossa representação do problema está correta, não foi possível determinar a melhor jogada com o algoritmo Minmax dado pela professora.

1.2.2 Alfa-beta

Não tendo a certeza de que o algoritmo Alfa-beta esteja 100% bem implementado, não foi possível determinar a melhor jogada. Independentemente disso, podemos afirmar que, como o algoritmo Minmax visita mais nós que o Alfa-beta (este limita o primeiro), o algoritmo Alfa-beta seria mais rápido.

2 Jogo do galo

2.1 O problema

2.1.1 Estados

Para a representação do problema, os estados são representados por uma lista de listas, onde cada lista interior representa uma linha do tabulero.

2.1.2 Terminal

Um estado é considerado terminal se nele existe uma linha, uma coluna ou uma diagonal preenchida com os símbolos "x"ou "o".

2.1.3 Função de utilidade

```
% Utility function (1->win, 0->draw, -1->lose)
valor(B, 1) :- lines(B, o).
valor(B, 1) :- columns(B, o).
valor(B, 1) :- diagonals(B, o).

valor(B, -1) :- lines(B, o).
valor(B, -1) :- columns(B, o).
valor(B, -1) :- diagonals(B, o).

valor(_, 0).
```

Figura 2: Função de utilidade