

## Estruturas de dados - Trabalho 2

### Game Trees

Prof. Eduardo A. P. Alchieri  
27 de abril de 2017

## 1 Objetivo

O presente projeto tem como objetivo a concretização dos conceitos de árvores. Mais especificamente, o principal objetivo do trabalho é utilizar uma árvore genérica para implementar a inteligência artificial de um jogo de tabuleiro. Estas árvores são chamadas de game trees.

## 2 Descrição do Projeto

O projeto será desenvolvido em grupos de até 2 (dois) alunos, e é constituído de três partes:

1. Relatório (com no mínimo 5 páginas), contendo:
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa
  - Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado
  - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso
2. Código fonte;

### 3. Apresentação oral do projeto.

O relatório deverá ser entregue em uma via impressa e outra digital (arquivo em formato pdf).

## 2.1 Prazo para entrega da versão impressa

A versão impressa do relatório deverá ser entregue no início da aula do dia **20/06/2017**.

## 2.2 Prazo para entrega da versão digital

A versão digital do projeto (relatório + código fonte) deverá ser enviada via email até às 10h00 do dia **20/06/2017**. Email para envio: [alchieri@unb.br](mailto:alchieri@unb.br)

A apresentação oral será feita pelo grupo na data especificada em sala de aula. Cada grupo terá até 20 minutos para apresentar o trabalho, e todos os componentes do grupo devem participar da apresentação.

# 3 Utilizando árvores genéricas (game trees) para implementar inteligência artificial em jogos de tabuleiro

Game Trees ([http://en.wikipedia.org/wiki/Game\\_tree](http://en.wikipedia.org/wiki/Game_tree)) são árvores que representam as possibilidades de jogadas para um jogador a partir de um estado do jogo. Estas árvores são utilizadas para verificar todas as possibilidades de jogadas, i.e., os filhos de um nó representam todas as possibilidades de jogadas a partir daquela situação do jogo (daquele nó). Para determinar a melhor jogada, deve ser desenvolvida uma função de avaliação que retorna um valor representando o quanto bom é um estado do jogo (por exemplo: se está mais próximo da vitória ou da derrota). Esta função deve ser utilizada em determinado nó (será aplicada apenas nas folhas) da árvore para atribuir um valor ao estado do jogo naquele nó. A figura 1 representa uma possível game tree para o jogo da velha. No primeiro nível da árvore encontra-se o estado atual do tabuleiro (raiz), no segundo nível são as jogadas possíveis do jogador 1 (computador, por exemplo), no terceiro nível (não representado na figura) são representadas as possíveis jogadas do jogador 2 (adversário do computador, por exemplo), e assim por diante.

Após isso, aplica-se o método minimax (<http://pt.wikipedia.org/wiki/Minimax>), onde o objetivo é retornar a jogada que maximize as chances de ganhar (como estas árvores implementam a inteligência artificial que o computador irá usar no jogo, queremos maximizar as chances do computador ganhar). O valor retornado por este método representa o menor ganho possível, desde que seja realizada a jogada sugerida.

Como vimos, no segundo nível da árvore (no primeiro está o estado atual do tabuleiro) estão os estados que representam as jogadas do computador, em que o objetivo é maximizar as chances de ganhar. No terceiro nível estão as jogadas que representam as possibilidades

do adversário, e assim por diante alternadamente. A altura da árvore representa quantas jogadas a frente o computador consegue ver (analisar).

É importante perceber que antes de cada jogada, o computador deve criar esta árvore, atribuir os valores aos estados do jogo (apenas para os nós folha) utilizando a função de avaliação, e por fim percorrer esta árvore utilizando o método minimax para escolher a melhor jogada. A jogada escolhida, de acordo com a função de avaliação do tabuleiro, será aquela que representa as maiores chances de ganhar, representando o ganho mínimo.

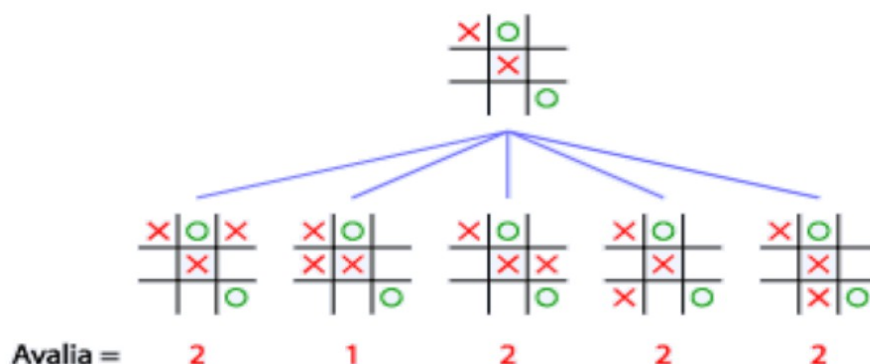


Figura 1: Game tree para o jogo da velha.

O algoritmo minimax é o seguinte:

- Se nó é folha, então retorna o valor da aplicação da função avaliação naquele nó.
- Se nó representa a jogada de um adversário (nível de minimização), aplica-se recursivamente minimax em cada filho e retorna o menor valor encontrado.
- Se nó representa a jogada do jogador (nível de maximização), aplica-se recursivamente minimax em cada filho e retorna o maior valor encontrado.

Neste trabalho, utilizaremos game trees e o algoritmo minimax para implementar a inteligência artificial do jogo mancala (<http://pt.wikipedia.org/wiki/Mancala>). Pode-se utilizar a interface de entrada e saída padrão (console). Como requisito, o jogo deve permitir a configuração do nível de inteligência, i.e., a altura da árvore a ser analisada.

### 3.1 O jogo mancala

A denominação Mancala abrange uma categoria de jogos de tabuleiro de origem africana, com regras ligeiramente diferentes, mas todos baseados em um mesmo esquema de jogo: a

semeadura de grãos em cavidades. O tabuleiro é composto de 14 cavidades, sendo 2 maiores, denominadas Kahalas; as outras 12 cavidades menores são divididas em 2 grupos, sendo 6 pertencentes a cada jogador. A Kahala de cada jogador fica ao lado direito das cavidades menores, e o número de pedras contidas nela define a pontuação do jogador. No início do jogo, são dispostas 4 pedras, em cada uma das cavidades menores. Nessa configuração, as pedras não pertencem a nenhum jogador em especial, mas sim ao tabuleiro. A Figura 3.1 exemplifica a configuração inicial.

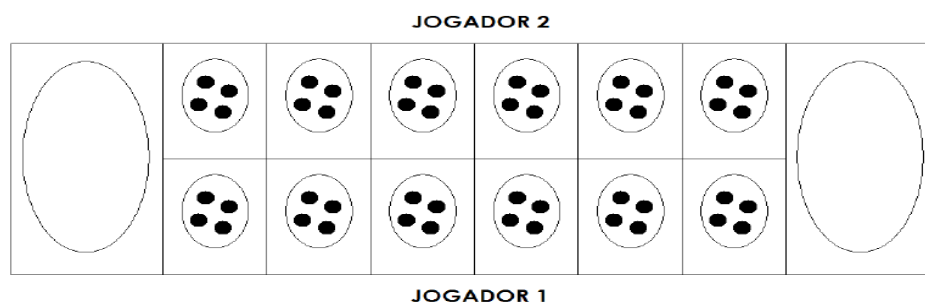


Figura 2: Configuração inicial do jogo mancala.

Uma jogada é realizada quando um jogador escolhe uma de suas 6 cavidades menores com número de pedras diferente de zero, e “semeia” seu conteúdo, em sentido anti-horário, depositando uma pedra em cada cavidade subsequente. A sua própria Kahala deve ser contemplada na semeadura, mas a Kahala adversária deve ser ignorada.

A captura se dá quando a última pedra semeada cai em uma cavidade vazia, pertencente ao jogador que efetuou a semeadura. Neste caso, tanto a pedra em questão como todas as pedras da cavidade adversária oposta a essa são recolhidas para a Kahala do jogador.

O final da partida se dá quando não houver mais pedras no lado do tabuleiro do jogador que deveria realizar a jogada. As peças que sobraram são recolhidas para a Kahala do jogador dono das cavidades onde se encontram. Vence o jogador que possuir o maior número de pedras em sua Kahala, ao final do jogo.

O jogo Mancala é de implementação simples, pois cada um de seus estados pode ser descrito como um vetor de 14 elementos, cada elemento representando o número de pedras em sua respectiva cavidade. Por essa simplicidade, o desenvolvimento da árvore de jogadas é facilitado.