

# Hoo-Doo Solver

Daniel Mendona and José Pedro Moreira

FEUP-PLOG, Turma 3MIEIC9, Grupo 123

**Abstract.** Este projecto consiste na implementao de um *solver* para o jogo de tabuleiro *Hoo-Doo*. O solver funciona para uma dimenso arbitrria do tabuleiro. A implementao foi feita usando Prolog, mas concretamente a plataforma *Sicstus Prolog* tendo sido usados para tal os mdulos desta mesma ferramenta para Programao em Lgica com Restries sobre domnios finitos.

## 1 Introduo

A realizao deste trabalho teve como um dos objectivos, perceber a importncia e utilidade da Programa em Lgica com restries. Outro dos objectivos foi o uso e conhecimento de bibliotecas do SICStus, que auxiliam na resoluo dos problemas existentes e o alcance dos seus predicados, que, por vezes, mesmo no estando directamente relacionados com um problema em concreto, sendo interpretados de formas alternativas se tornam bastante teis, quer na simplificao do problema, quer na procura da sua resoluo. O Hoo-Doo, dependendo da sua dimenso, pode ter vrias, uma ou at nenhuma resoluo se no forem usadas pegs transparentes(peg uma pea de cor, ser explicado em detalhe na descrio do jogo). Quando foi lanado o jogo de tabuleiro Hoo-doo, os seus criadores ofereciam 1000\$ primeira pessoa que conseguisse resolver um tabuleiro de 8x8 sem recurso a pegs transparentes, e, os elementos do grupo acharam que seria interessante e at certo ponto divertido verificar como a implementao deste jogo em Prolog, usando restries seria uma mais valia para vencer o prmio que era na altura oferecido. A pouca informao encontrada sobre este jogo de tabuleiro tambm despertou curiosidade. ...

## 2 Descrio

O jogo Hoo-Doo foi criado pela empresa(?) Tryne Games, lanado na dcada de 50. Hoo-Doo um jogo de tabuleiro, para um nico jogador, normalmente quadrado e que tem pelo menos tantas cores quanto o nmero de colunas no tabuleiro, e o numero de pegs(peg a designao de uma pea no jogo Hoo-Doo) de cada cor tambm o numero de colunas do tabuleiro. Os tamanhos de tabuleiro mais frequentes so os de 4x4, 6x6 e 8x8. O jogo tem como incio um tabuleiro vazio, e o objectivo preencher todas as posies do tabuleiro com as pegs disponveis, sem nunca repetir peas da mesma cor quer na mesma linha, coluna, ou qualquer uma das diagonais. Para auxiliar na resoluo do tabuleiro, existem as denominadas pegs transparentes, cuja sua caracterstica preencher uma posio sem lhe atribuir

uma cor. O uso de pegs transparentes é absolutamente necessário para a resolução de tabuleiros com determinados tamanhos, cuja resolução é possível apenas com o uso de pegs transparentes (como exemplo temos um tabuleiro de 6x6, que é impossível resolver mesmo com duas pegs transparentes). Ser sempre considerada a melhor resolução, aquela que usar menos pegs transparentes, sendo portanto esse, o parâmetro de otimização.

### 3 Visualização

Existem seis predicados utilizados para a construção visual do tabuleiro em modo de texto. O primeiro predicado a ser executado é o *print\_tab(+board)* que recebe como argumento um tabuleiro representado por uma lista de listas. Este predicado calcula o comprimento da lista, que determina o número de linhas, colunas e respectivos índices a serem imprimidos, e de seguida passa-os como argumentos para as funções auxiliares que controlam a impressão, descritas em baixo.

- *print\_tab\_aux(+Board, ?LineI, ?ColumnI)*: coordena a utilização dos seguintes predicados para a construção visual do tabuleiro.
- *tab\_map(+Symb)*: Imprime o número ou correspondente no tabuleiro.
- *print\_line(+Line)*: imprime uma linha do tabuleiro, fazendo uso do *tab\_map(+Symb)* para a impressão numérica.
- *print\_empty\_line(+Length)*: imprime uma linha horizontal.
- *print\_column\_index(+ASCIICode, +Index)*: imprime o índice das colunas.