Autores e Afiliações: identificação do trabalho e do grupo. Nas afiliações colocar a turma e o grupo segundo o seguinte exemplo: FEUP-PLOG, Turma 3MIEIC9, Grupo Xpto_1.

PLOG - Programação em Lógica

Dominos

2º Trabalho laboratorial-Resolução de Problema de Decisão/Optimização usando Programação em Lógica com Restrições

Turma 3MIEIC06 Grupo Dominos_4:
Francisco Miguel Lamares Martins Barbosa- up201404264
Paulo Azevedo Peixoto- up201306002

23 de Dezembro de 2016

Resumo

Com este trabalho pretende-se demonstrar a eficácia da utilização de programação em lógica com restrições para resolver problemas de decisão/otimização. Este problema é baseado no enigma Dominos, que consiste num tabuleiro retangular dividido por celulas, e em cada celula está inserido um numero. O objetivo é inserir peças de domino nesse tabuleiro, sendo que cada peça pode ser inserida apenas uma vez. Através da utilização de predicados disponibilizados pela biblioteca clp(fd) do SICStus Prolog, foi possivel resolver este problema de uma forma bastante eficiente. .

1. Introdução

Descrição dos objetivos e motivação do trabalho, referência sucinta ao problema em análise (idealmente, referência a outros trabalhos sobre o mesmo problema e sua abordagem), e descrição sucinta da estrutura do resto do artigo.

Este projeto, desenvolvido no âmbito da unidade curricular Programação em Lógica, permite aos alunos desenvolver o seu conhecimento na àrea de resolução de problemas de decisão, usando conceitos de programação em lógica com restrições.

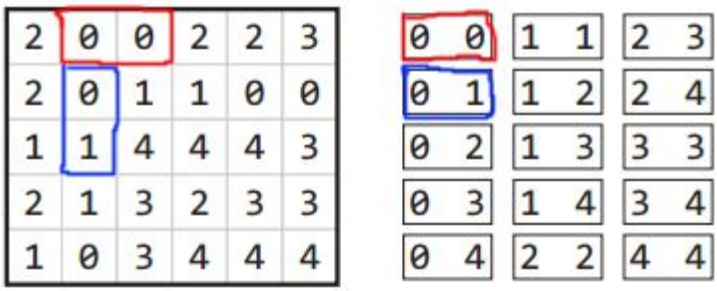
Acabamos por escolher o tema Dominos, pois das opções propostas pareceu a mais interessante e foi a que mais nos identificamos. O problema em si resume-se a "encaixar" peças de Dominó num tabuleiro previamente numerado, no entanto cada peça só pode ser utilizada uma vez e o tabuleiro deve ficar completamente preenchido por peças.

Em suma, este artigo tem como foco dar a conhecer a nossa abordagem ao problema Dominos, encontrando-se explícita ,ao longo do trabalho, a sua exposição e resolução detalhada.

2. Descrição do Problema

O problema em causa denomina-se por Dominos e é composto por um tabuleiro numerado de 1 a X, e por peças de domino (cujos valores tambem se encontram entre 1 e X).

O objetivo do trabalho é encontrar uma resolução que passe por substituir cada peça num determinado sítio do tabuleiro, em que cada peça substitui 2 números(células) do tabuleiro, sendo que cada peça só pode ser usada uma vez. Como se mostra na figura seguinte, a ideia é substituir uma peça no tabuleiro, sendo que esta peça não pode voltar a ser usada. Logo,se a peça vermelha (0,0) substituir as duas casas rodeadas a vermelho no tabuleiro, esta peça não pode ser utilizada novamente.



Assim, o problema fica resolvido quando o tabuleiro fica totalmente substituído por peças e, por consequência, quando não sobram peças.

3. Abordagem

Inicialmente o grupo tentou perceber como estruturar o dominos num problema de restrições. O primeiro passo para tal, foi entender as variáveis de decisão a utilizar no predicado de labeling e a forma mais correta de as restringir.

3.1 Variáveis de Decisão

A solução pretendida é devolvida por uma lista, de S1 até Sn, em que n é o tamanho da lista, usando o tabuleiro ilustrado em cima, Sol=[S1,...,S30]. Cada elemento representa o número da peça,ou seja, na figura acima a peça (0,0) é a peça 1,sendo que a numeração das peças vai de 1 até 15. Assim, no índice 2 (S2) e 3 (S3) do tabuleiro, S2=1 e S3=1. Cada elemento pode portanto tomar valores desde 1 até ao número de peças no tabuleiro, ou seja, número de células/2.

3.2 Restrições

As funções que verificam as restrições são as seguintes

- **restrainEveryValue -** garante que cada valor referente à peça aparece duas vezes, e apenas 2, no tabuleiro. Isto porque uma peça irá ocupar o espaço de duas células.
- **restrainAdjacentCellsFunctor -** restrição aplicada a todas as células que garante que uma, e apenas uma, das peças adjacentes tem o mesmo valor que a peça em análise.
- **restrainBoard -** faz a ligação entre os valores das peças e os valores do tabuleiro. Sendo que um par de células com o mesmo número de peças tem de verificar a regra Piece(NumPeça, ValorA, ValorB).

3.3 Função de Avaliação

Não foi necessário utilizar nenhuma função de avaliação, visto não se tratar de um problema de otimização.

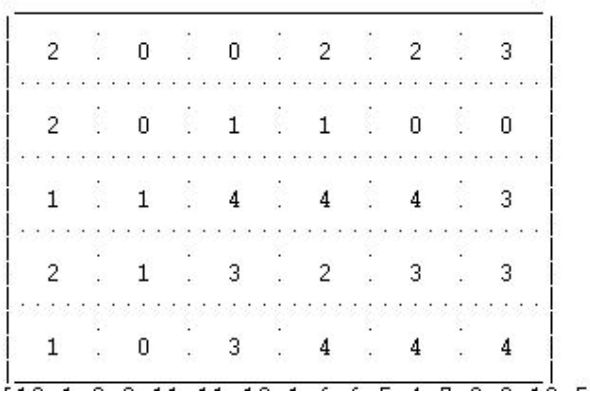
3.4 Estratégia de Pesquisa

Foi utilizada a opção default no labeling visto que todas as variáveis têm o mesmo domínio e não se pretende otimizar nenhum parâmetro.

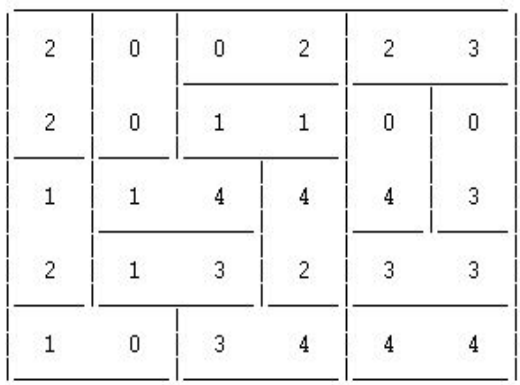
4. Visualização da Solução / Solution Presentation: Explicar os predicados que permitem visualizar a solução em modo de texto.

Os predicados utilzados para a visualização são:

- **printBoard -** Mostra o tabuleiro antes de ser resolvido.



- printSolution - Mostra a resolução com as peças sobre o tabuleiro.



5. Resultados

Os resultados da execução do programa em tabuleiros de diferentes tamanhos são os seguintes:

	Ellapsed Time	Resumptions	Entailments	Prunings	Backtracks	Constraints Created
3x2	37.7	50	21	48	0	21
4x3	246.3	508	190	334	0	78
4x4	18.5	722	186	398	0	136
5x4	33.2	988	210	510	0	210
6x4	60.7	2229	477	980	0	300
6x5	32.9	71941	21386	30164	0	465

Os dados obtidos não são muito conclusivos devido ao tipo de testes realizados, com tabuleiros relativamente pequenos. No entanto é possível observar um crescimento da complexidade da computação com o aumento das dimensões do tabuleiro.

6. Conclusões

Ao longo do projeto, fomo-nos apercebendo das possibilidades que o uso de restrições e labeling nos oferecem, tornando possível a resolução de alguns problemas de uma forma bastante mais simples.

O grupo conseguiu, facilmente perceber e interpretar o problema proposto, no entanto teve alguma dificuldade em perceber qual o melhor método a implementar para a sua resolução, mas acabou por conseguir resolver o problema sem quaisquer limitações.

Assim, todos os objetivos propostos foram cumpridos e a principal conclusão a retirar deste projeto é a de que a linguagem Prolog, usando restrições, é bastante poderosa para resolver uma ampla variedade de questões de decisão/otimização.