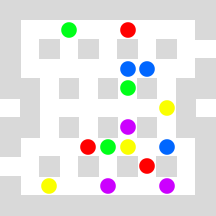
Bird Watching



Disciplina de Programação em Lógica, do 3º Ano (1º Semestre) do Mestrado Integrado em Eng. Informática e Computação da FEUP



Carlos Tiago da Rocha Babo

Felipe de Souza Schmitt

Helder Alexandre Moreira dos Santos

FEUP-PL, Turma 3MIEIC1, Grupo 106

**Resumo/Abstract.**

O trabalho foi realizado no âmbito da disciplina de Programação em Lógica na Faculdade de Engenharia da Universidade do Porto, com o objectivo de aplicar os conhecimentos adquiridos ao longo do semestre, mais propriamente programação de PROLOG com restrições de forma a encontrar uma solução para o puzzle Birdwatching.

Introdução

O objectivo principal para este trabalho foi aplicar os nossos conhecimentos adquiridos em Programação em Lógica. Na escolha do tema achamos que este problema era cativante e interessante, sendo estas as nossas principais motivações na resolução do trabalho.

Este artigo é constituido por três principais partes, em primeiro lugar é apresentado a descrição do problema a resolver. De seguida é demonstrado as abordagens utilizadas para a resolução desse mesmo problema e por fim uma analise dos resultados obtidos e a comparação entre abordagens diferentes.

Descrição do problema

O problema Birdwatching consiste num puzzle em que é necessário encontrar um caminho entre a entrada do tabuleiro e a sua saida em que passe por um e apenas um pássaro de cada cor, não podendo realizar o mesmo caminho mais do que uma vez.

Sendo o objectivo realizar uma solução utilizando a linguagem PROLOG com restrições.

Ficheiros de Dados

Escolhemos vários problemas em que consistiam em vários puzzles em que o objectivo era passar por um passáro de cada cor e também escolhemos puzzles em que o objectivo era passar não um, mas dois pássaros de cada cor.

O primeiro problema que utilizamos para encontrar a resolução através das restrições que definimos encontra se em anexo no ponto 1, junto com a sua solução.

O segundo problema é de mesma dificuldade que o primeiro, alterando a posição dos pássaros no tabuleiro, no ponto 2 do anexo.

O terceiro problema destaca-se dos outros pois neste caso é necessário encontrar um caminho desde a entrada à saida do tabuleiro que passe por apenas dois pássaros de cada cor e encontra-se no ponto 3 do anexo deste artigo.

Variáveis de Decisão

Neste trabalho existe uma variável de decisão, designada por “Caminho” que indica o caminho percorrido para chegar à saida do tabuleiro. Esta variável de decisão tem um dominio de 1 a CaminhoMax+1, em que a variável CaminhoMax é o numero máximo que o caminho pode ter e é calculada pela subtracção das casas a zero e das casas por onde o caminho não passa às 121 casas totais do tabuleiro, sendo no pior caso 60 o número máximo do caminho e 61 o número de casas por onde o caminho não passa.

Restrições

As restrições aplicadas para a solução deste trabalho foram em primeiro lugar as restrições das paredes do tabuleiro em que estão assinaladas com um 0 no tabuleiro, de seguida a restrição foi aplicar a conectividade 4, ou seja num caminho só era possivel dirigir se para cima, baixo, esquerda ou direita, a proxima restrição aplicada é definir as casas que não são do caminho como paredes.

As restrições rígidas aplicadas foram a casa de entrada, a casa adjacente à entrada, a casa de saida e a casa adjacente à saida.

Enquanto as restrições flexiveis aplicadas foram que as posições por onde é possivel passar num caminho podem assumir o valor da sua ordem caso pertençam ao caminho ou o valor da parede caso não o pertençam.

Função de Avaliação

Estratégia de Pesquisa

Na implementação da etiquetagem (“labeling”) utilizamos uma estratégia que é incluir na sua chamada o min e o up, de forma a que a etiquetagem começasse pela variavél mais à esquerda com o dominio esquerdo mais pequeno, ou seja pela entrada do tabuleiro. Enquanto o up é para o dominio ser explorado em ordem ascendente , desta forma processando primeiro os caminhos mais pequenos, uma vez que a etiquetagem destes é mais rápida.

Visualização da Solução

A visualização da solução

Resultados

Em primeiro lugar foi desenvolvida uma resolução para o problema que era hibrida, pois incluia prolog sem e com restrições, calculando os caminhos a percorrer através de prolog sem restrições e posteriormente aplicar as restrições para que a solução apenas tivesse um pássaro de cada cor. Após o desenvolvimento desta solução foi verificada que a solução para os problemas em que o caminho teria que passar por apenas um pássaro eram resolvidos entre 850ms a 900ms. Enquanto na solução dos problemas que o caminho teria que passar obrigatóriamente por dois pássaros eram resolvidos em entre 3 minutos e 40 segundos a 3 minutos e 50 segundos.

Uma vez que para a resolução deste segundo problema consideramos que o seu tempo de execução era demasiado procuramos tentar abordar o problema de outra forma e criar uma resolução utilizando apenas prolog com restrições.

Após realizar uma solução para o problema utilizando apenas restrições conseguimos observar através dos resultados que para a resolução de um problema em que o caminho tem que passar por um pássaro o tempo de execução é acerca de 1900ms ~2segundos. Ou seja o cerca de o dobro da solução dada pela resolução hibrida. Mas no caso do problema em que o caminho tem que passar por dois pássaros de cada cor o tempo de execução é consideravelmente menor, sendo acerca de 14 segundos, diminuindo assim o tempo de execução para 6,3% do tempo inicial.

Conclusões e Perspectivas de Desenvolvimento

Bibliografia

<http://www2.stetson.edu/~efriedma/birds/>

<http://www.fi.muni.cz/~hanka/sicstus/doc-3.11/pdf/sicstus.pdf>

Anexos

1. Problema 1

[0,0,0,0,0,0,0,0,0,0,0,

0,1,1,2,1,1,3,1,1,1,0,

0,1,0,1,0,1,0,1,0,1,1,

0,1,1,1,1,1,4,4,1,1,0,

0,0,1,0,1,0,2,0,1,0,0,

0,0,1,1,1,1,1,1,5,0,0,

0,0,1,0,1,0,6,0,1,0,0,

0,1,1,1,3,2,5,1,4,1,0,

1,1,0,1,0,1,0,3,0,1,0,

0,1,5,1,1,6,1,1,6,1,0,

0,0,0,0,0,0,0,0,0,0,0].

Onde a solução seria percorrer o seguinte caminho:

[0,0,0,0,0,0,0,0,0,0,0,

0,1,1,2,1,1,3,→,→,↓,0,

0,1,0,1,0,1,0,↑,0,→,→,

0,1,1,1,1,1,4,**4**,←,1,0,

0,0,1,0,1,0,2,0,↑,0,0,

0,0,1,1,→,→,→,→,**5**,0,0,

0,0,1,0,↑,0,6,0,1,0,0,

0,→,→,↓,**3**,**2**,5,1,4,1,0,

→,↑,0,↓,0,↑,0,3,0,1,0,

0,1,5,→,→,**6**,1,1,6,1,0,

0,0,0,0,0,0,0,0,0,0,0].

1. Problema 2

[0,0,0,0,0,0,0,0,0,0,0,

0,1,1,5,1,1,3,4,1,1,0,

0,1,0,1,0,1,0,1,0,1,1,

0,1,1,3,1,1,2,2,1,1,0,

0,0,1,0,5,0,1,0,1,0,0,

0,0,1,1,4,1,2,6,1,0,0,

0,0,6,0,1,0,1,0,1,0,0,

0,4,1,1,1,1,6,1,1,1,0,

1,1,0,1,0,1,0,1,0,1,0,

0,1,1,1,5,1,3,1,1,1,0,

0,0,0,0,0,0,0,0,0,0,0].

E a sua solução:

[0,0,0,0,0,0,0,0,0,0,0,

0,→,→,**5**,→,→,**3**,**4**,1,1,0,

0,↑,0,1,0,1,0,↓,0,→,→,

0,↑,←,3,1,1,2,**2**,→,↑,0,

0,0,↑,0,5,0,1,0,1,0,0,

0,0,↑,1,4,1,2,6,1,0,0,

0,0,**6**,0,1,0,1,0,1,0,0,

0,4,↑,←,1,1,6,1,1,1,0,

→,→,0,↑,0,1,0,1,0,1,0,

0,↓,→,→,5,1,3,1,1,1,0,

0,0,0,0,0,0,0,0,0,0,0].

1. Problema 3

[0,0,0,0,0,0,0,0,0,0,0,

0,6,6,1,4,3,1,2,1,1,0,

0,1,0,4,0,4,0,1,0,1,1,

0,1,1,1,2,1,2,5,5,5,0,

0,0,1,0,1,0,1,0,1,0,0,

0,0,1,1,1,3,4,1,1,0,0,

0,0,1,0,6,0,1,0,1,0,0,

0,1,1,1,1,6,1,1,1,1,0,

1,1,0,1,0,3,0,1,0,1,0,

0,1,1,1,1,1,3,5,1,2,0,

0,0,0,0,0,0,0,0,0,0,0].

Sendo a sua solução o seguinte caminho:

[0,0,0,0,0,0,0,0,0,0,0,

0,6,6,1,4,**3**,→,**2**,→,↓,0,

0,1,0,4,0,**4**,0,1,0,→,→,

0,1,1,1,2,↑,**2**,**5**,**5**,5,0,

0,0,1,0,1,0,1,0,↑,0,0,

0,0,→,→,↓,3,**4**,→,↑,0,0,

0,0,↑,0,**6**,0,↑,0,1,0,0,

0,→,↑,↓,←,**6**,↑,1,1,1,0,

→,↑,0,↓,0,**3**,0,1,0,1,1,

0,1,1,→,→,↑,3,5,1,2,0,

0,0,0,0,0,0,0,0,0,0,0].