Grupo Nº \_42\_



**Inteligência Artificial**

1.º Semestre 2014/2015

**Fill-a-Pix**

Relatório de Projecto

Índice

[1 Implementação Tipos e Representação Problema PSR 2](#_Toc404722800)

[1.1 Tipos Abstractos de Informação 2](#_Toc404722801)

[1.2 Representação do problema Fill-a-Pix como PSR 3](#_Toc404722802)

[2 Implementação Procuras e Funções Obrigatórias 4](#_Toc404722803)

[2.1 Fill-a-pix🡪psr 4](#_Toc404722804)

[2.2 Psr->Fill-a-pix 4](#_Toc404722805)

[2.3 Heurística de Grau 4](#_Toc404722806)

[2.4 Heurística MRV 4](#_Toc404722807)

[2.5 Procura-Retrocesso e Inferência 4](#_Toc404722808)

[3 Optimizações, Heurísticas e Técnicas adicionais utilizadas 5](#_Toc404722809)

[3.1 Optimizações especificas para o problema Fill-a-Pix 5](#_Toc404722810)

[3.2 Criação/Combinação de Heurísticas 5](#_Toc404722811)

[3.3 Utilização de técnicas adicionais 5](#_Toc404722812)

[4 Estudo Comparativo 6](#_Toc404722813)

[4.1 Critérios a analisar 6](#_Toc404722814)

[4.2 Testes Efectuados 6](#_Toc404722815)

[4.3 Resultados Obtidos 6](#_Toc404722816)

[4.4 Comparação dos Resultados Obtidos 6](#_Toc404722817)

[4.5 Escolha do resolve-best 6](#_Toc404722818)

# Implementação Tipos e Representação Problema PSR

## Tipos Abstractos de Informação

Neste ponto devem ser abordados, pelo menos, os seguintes tópicos:

•Descrição detalhada das estruturas de dados utilizadas no programa.

•Justificação das escolhas efectuadas em relação a possíveis alternativas e às operações implementadas.

Se implementarem mais do que um tipo psr para o projecto, deverão descrever os vários tipos psr implementados e justificar a sua implementação.

-- começa aqui --

Para implementarmos o tipo Restrição usámos apenas listas porque era um tipo bastante simples só com dois elementos

Para implementarmos o PSR, utilizámos uma lista para guardar os domínios de cada variável e outra lista para as várias restrições do problema. Utilizámos uma hash-table em que a chave é uma variável para guardar um par com a posição do domínio dessa variável na lista de domínios e o segundo elemento do par era o valor atribuído à variável. Utilizámos ainda outra hash-table em que a chave é novamente a variável e o valor guardado é uma lista com as restrições em que essa variável está inserida.

Relativamente ao uso de listas, utilizámos esta estrutura em detrimento, por exemplo de vectores por serem de fácil utilização e manipulação e aplicação de funções como mapcar e reverse.

As hash-tables foram escolhidas para podermos ter um acesso rápido utilizando as variáveis em vez de termos um vector ou lista e depois termos que percorrer a estrutura até a encontrarmos.

## Representação do problema Fill-a-Pix como PSR

Nesta secção deverão descrever como é que decidiram representar o puzzle Fill-a-Pix como um problema de Satisfação de Restrições. Ou seja, deverão indicar as variáveis, os domínios, e as restrições escolhidas para representar um puzzle Fill-a-Pix. Deverão também justificar a escolha efectuada, comparando com possíveis alternativas de representação.

-- começa aqui –

# Implementação Procuras e Funções Obrigatórias

Nesta secção deverão descrever a implementação das várias funções/heurísticas e mecanismos de inferência pedidos explicitamente no enunciado. No entanto, deverão apenas focar-se na descrição de particularidades que não estejam especificadas no documento de esclarecimento. Ou seja, **não queremos ler o pseudocódigo e respectivos comentários** criados por nós. Em particular devem descrever as seguintes funções/algoritmos:

## Fill-a-pix🡪psr

Deverão descrever brevemente o algoritmo de conversão de um tabuleiro Fill-a-Pix para o tipo PSR. Devido à sua complexidade, é importante descrever o processo de criação e teste de restrições.

## Psr->Fill-a-pix

Aqui deverão descrever o algoritmo oposto, i.e. a conversão de um PSR resolvido para um tabuleiro Fill-a-Pix.

## Heurística de Grau

Deverão descrever brevemente como é que implementaram a heurística de grau.

## Heurística MRV

Deverão descrever brevemente como é que implementaram a heurística MRV.

## Procura-Retrocesso e Inferência

Nesta secção deverão descrever algum detalhe que considerem relevante na implementação das procuras por retrocesso com os vários tipos de inferência. Mais uma vez, aqui não é pretendido que façam copy-paste do pseudocódigo, mas sim que descrevam algum pormenor que achem importante. Por exemplo, o que usaram para representar uma inferência, e como é que essa é aplicada/removida ao psr.

# Optimizações, Heurísticas e Técnicas adicionais utilizadas

Neste capítulo deverão descrever com algum detalhe 3 tipos de estratégias usadas para tornar o vosso algoritmo de procura mais eficiente.

## Optimizações especificas para o problema Fill-a-Pix

Nesta secção deverão descrever optimizações efectuadas nas funções de modo a tornar a procura por retrocesso mais eficiente para um puzzle especifico do Fill-a-Pix. Podem também descrever outros tipos de optimizações efectuadas ao pseudocódigo original fornecido. Por exemplo, se não houver a restrição de que as coisas têm que ser feitas por uma determinada ordem, é possível criar versões ligeiramente mais eficientes da função vizinhos-não-atribuidos, da função psr-variaveis-não-atribuidas, etc.

## Criação/Combinação de Heurísticas

Se criarem alguma heurística adicional para a escolha de variável/valor, ou se usarem alguma combinação de heurísticas deverão descrevê-las nesta secção.

## Utilização de técnicas adicionais

Nesta secção final deverão descrever outras técnicas adicionais utilizadas para tentar tornar o algoritmo de procura mais eficiente. Alguns exemplos (que podem funcionar ou não) são:

* Conflict-directed backjumping.
* Execução algoritmo MR3 antes de iniciar a procura.
* Utilização de restrições globais para testes de consistência e inferência.

# Estudo Comparativo

Pretende-se estudar a eficácia dos vários algoritmos e variantes implementados para a resolução de um problema Fill-a-Pix.

## Critérios a analisar

Descrever quais os critérios a ser usados para comparar as várias variantes e justificar a sua escolha.

## Testes Efectuados

Descrever os problemas utilizados para efectuar o estudo. Justificar a sua *relevância* para os critérios a comparar.

## Resultados Obtidos

Enumerar os resultados obtidos, sob a forma mais adequada (tabela e gráficos, se necessário).

## Comparação dos Resultados Obtidos

Analisar, caso a caso, os resultados, focando a diferença de desempenho das várias variantes entre si. A análise não deve ser somente uma descrição dos resultados obtidos, mas sim uma análise dos factores que possam ter conduzido a esses resultados.

## Escolha do resolve-best

Com base nos resultados obtidos na comparação anterior, devem descrever e justificar quais as variantes escolhidas para implementar a vossa função resolve-best.