**Resolução de um Puzzle usando**

**Programação em Lógica com Restrições:**

***Grape***

Miguel Neves e José Miguel Maçães

Turma 2, Grupo Grape\_5

Faculdade de Engenharia da Universidade do Porto

Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

**Resumo** Este trabalho tem como objetivo a resolução de um problema de otimização baseado no puzzle *Grape*, fazendo uso do SICStus Prolog. A finalidade do *Grape* é o preenchimento de células em filas sucessivas, cada fila com menos uma célula que a anterior, fazendo com que as células contenham as somas das células contíguas da fila anterior, havendo células “coloridas”, tal que duas células devem conter o mesmo número, se forem da mesma cor.

Para além disso deverão ser gerados puzzles aleatoriamente. Assim, utilizando a linguagem Prolog, foi possível a resolução deste mesmo puzzle.

**Keywords:** Redistribuição de Público, SICStus, Prolog, FEUP

# Introdução

O projeto está enquadrado na unidade curricular de Programação em Lógica de 3º ano do curso Mestrado Integrado em Engenharia Informática e de Computação. Durante o seu desenvolvimento, foi necessário implementar uma possível resolução para um puzzle em Prolog com restrições, bem como gerar problemas aleatórios para resolver. O problema escolhido foi o puzzle *Grape*.

O esquema do puzzle escolhido é em forma de triângulo invertido, havendo linhas sucessivas, cada uma com menos uma célula que a outra, sendo a última formada por apenas um elemento. O objetivo é preencher cada célula com um número inteiro, respeitando as respetivas somas da linha anterior. Algumas células têm cores, tais que todas as células da mesma cor devem ser preenchidas com o mesmo número.

Este artigo tem a seguinte estrutura:

* **Descrição do Puzzle:** descrição detalhada do puzzle em questão.
* **Abordagem:** apresentação da representação do problema e da sua resolução, de acordo com as seguintes subsecções.
  + **Variáveis:** escolha das variáveis do problema e definição do domínio.
  + **Restrições:** descrição das restrições do problema e a sua implementação em Prolog.
  + **Função de Avaliação:** !!!!!!!!!!!!!!
  + **Estratégia de Pesquisa:** estratégia de *labeling* implementada.
* **Visualização da Solução:** explicação dos predicados que permitem apresentar a solução de forma visualmente acessível.
* **Geração Aleatória de Puzzle:** predicados de geração automática do puzzle.
* **Resultados:** análise dos resultados obtidos para exemplos de puzzles.
* **Conclusões:** conclusões retiradas deste projeto.
* **Anexo:** código fonte, ficheiros de dados e resultados, entre outros.

# Descrição do Puzzle

*Grape* é um puzzle em forma de triângulo invertido, composto linhas de células, cada uma com menos uma célula que a anterior, terminando numa linha formada por apenas uma célula. O objetivo é preencher cada célula com um número inteiro, sendo que a primeira linha deve conter apenas números entre 1 e 9 e as restantes filas são completadas com base na fila anterior: um elemento de uma linha é igual à soma dos dois elementos contíguos acima desse mesmo elemento. Algumas células têm cores, tais que todas as células da mesma cor devem ser preenchidas com o mesmo número.

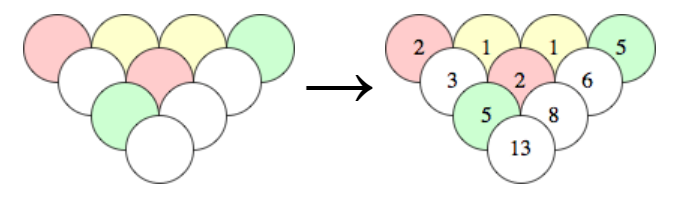
Exemplo:

Figura 1: Exemplo do puzzle

# Abordagem

Para a resolução deste problema foi criada um estado inicial de variáveis para as células a preencher, sendo as células da mesma cor representadas pela mesma variável.

Assim, se um problema for da forma do exemplo da figura 1, seria representado por:

[A,B,B,C]

[D,A,E]

[C,F]

[G]

## Variáveis

Para resolver este problema, as variáveis estão numa só lista, seguindo a ordem das filas, ou seja, as variáveis para resolver o exemplo da figura 1 seriam uma lista [A,B,B,C,D,A,E,C,F,G].

O seu domínio será inicialmente definido entre 1 e o valor máximo possível tendo em conta o tamanho da primeira linha, seguindo a fórmula Máximo = 9 \* 2^(TamanhoPrimeiraLinha-1).

## Restrições

As variáveis têm que ser todas distintas**.**Apesar de as cores implicarem que haverá células preenchidas com o mesmo número, o facto de estas células serem representadas pela mesma variável já garante esses casos.

**Cada elemento da primeira linha está entre 1 e 9.**Esta restrição é feita percorrendo a lista de variáveis e adicionando esta restrição a cada elemento.

**Cada elemento é a soma dos elementos contíguos na linha anterior.**Esta restrição é feita percorrendo a lista de variáveis e adicionando restrições a cada elemento relativas aos elementos anteriores.

## Função de Avaliação

!!!!!!!!!!!

## Estratégia de Pesquisa

# Geração Aleatória de Puzzle

# Visualização da Solução

# Resultados

# Conclusões

# Anexo