



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO - CTC  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

INE5416 - Paradigmas de Programação  
Professora: Maicon Rafael Zatelli

Fernanda Larissa Müller<sup>1</sup> - 21202109  
Leonardo Brito<sup>1</sup> - 21200896  
Isac Martins<sup>1</sup> - 21203361

<sup>1</sup> Programa de Graduação em Ciência da Computação, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina.

Relatório Trabalho 3:  
Programação Lógica - Prolog

Florianópolis, 29 de novembro de 2023.

# 1. Análise do problema

Anteriormente foram realizadas duas versões de um resolvidor do jogo Kojun. A primeira em Haskell, onde a solução utilizava de recursão e de elementos específicos do Haskell para a solução do problema, usando backtracking para resolver a situação do board. E a segunda utilizando Scala em uma linguagem que funde POO e Programação Funcional.

Agora, para o desenvolvimento do resolvidor, o objetivo é utilizar a linguagem Prolog e utilizar o conceito de programação de restrições (*constraint programming*).

# 2. Solução adotada

Assim como nos trabalhos anteriores, esse trabalho consiste em um solver para um quebra-cabeça Kojun, onde o objetivo é preencher uma matriz com números, seguindo determinadas regras de região e adjacência, mas agora utilizando a linguagem Prolog e utilizando o conceito de programação de restrições (*constraint programming*).

A Programação por Restrições (CP) é uma abordagem de resolução de problemas que se baseia na declaração de restrições sobre as variáveis de um problema, sem explicitamente definir a ordem em que essas variáveis devem ser atribuídas. O sistema de CP usa técnicas de inferência para explorar e reduzir o espaço de busca, buscando encontrar uma solução para o problema que satisfaça todas as restrições.

Neste trabalho a CP foi implementada usando o módulo clpfd (Constraints Logic Programming over Finite Domains) do SWI-Prolog, como sugerido pelo professor. Esse módulo fornece uma linguagem de programação declarativa para lidar com restrições sobre variáveis inteiras finitas.

Aqui estão algumas partes específicas do código onde utilizamos CP:

- **Declaração de Domínios:**

Esta declaração define que a variável X deve assumir um valor no domínio de 1 a T, onde T é o tamanho da região.

```
/*define o maior valor que os valores de cada regioao podem assumir*/  
valor_maximo_regiao([R,X]) :- regioao_quantidade(R,T), X in 1..T.
```

- **Restrições de vizinhança:**

Essa restrição exige que os valores X1 e X2 sejam diferentes. Isso é usado para garantir que os vizinhos adjacentes em uma linha ou coluna não tenham o mesmo valor.

```
/*verifica se o vizinho a direita eh diferente*/  
vizinhos_diferentes([_,_]).  
vizinhos_diferentes([_,X1],[R2,X2]|T) :-  
X1 #\= X2, append([R2,X2],T,L), vizinhos_diferentes(L).
```

- **Restrição de ordem:**

Essa restrição exige que X1 seja maior que X2. Isso é usado para garantir que o valor acima de outro valor na mesma coluna seja maior, se eles fizerem parte do mesmo grupo.

```
/*verifica se o valor acima de outro eh maior, se fizerem parte do mesmo grupo*/  
superior_maior([[_,_]]).  
superior_maior([[R1,X1],[R2,X2]|T]) :-  
    ((R1 #\= R2);  
    (X1 #> X2)), append([[R2,X2]],T,L), superior_maior(L).
```

### 3. Organização

O trabalho foi realizado ao longo das semanas disponíveis através de reuniões do discord onde nos juntamos para discutir nossas ideias e trabalhar no projeto utilizando a ferramenta de Live Share do Visual Studio Code.

### 4. Dificuldades encontradas

As maiores dificuldades foram encontrar erros no código e adaptar o código ao novo paradigma, uma vez que tem grande diferença com o Haskell, Scala e Prolog. Apesar da criação de um código completamente novo, o conhecimento prévio do problema do Kojun e das estratégias para sua resolução proporcionou uma orientação valiosa. Isso ajudou a delinear as responsabilidades de cada predicado, facilitando a implementação. Contudo, a transição para um paradigma tão distinto apresentou desafios, especialmente na compreensão dos possíveis equívocos em cada predicado.