

**Universidade Federal de Santa Catarina (UFSC)**

Campus Reitor João David Ferreira Lima

Departamento de Informática e Estatística

Bacharelado em Ciência da Computação

**Disciplina INE5416 - Paradigmas de Programação - 2023.2**

Prof. Rafael Santiago

26/09/2023

**Alunos:**

João Pedro Resmer

# Relatório - Atividade I - Paradigmas de Programação

## 1. Proposta:

O trabalho I da disciplina INE5416 no semestre 2023-2 consiste da implementação de um programa em Haskell que solucione um dos três *puzzles*: Kojun, Makaro, Comparison Sudoku. O código desenvolvido pelo aluno resolve Comparison Sudoku.

## 2. Programa:

### 2.1. Algoritmo:

O algoritmo escolhido foi o backtracking, que é uma método que utiliza de recursão para testar possibilidades até que alguma satisfaça as condições que a definiriam como o objetivo ou até a exaustão. Junto ao backtracking foi adotada uma heurística baseada em restrições como forma de acelerar a obtenção de uma resposta. A implementação foi feita com base no pseudocódigo apresentado no curso CS50 - Introduction to AI with Python da Universidade de Harvard:

```
AC(csp):
    queue = all arcs in csp
    while queue non-empty:
        (X, Y) = dequeue(queue)
        if revise(csp, X, Y):
            if size of X.domain == 0:
                return false
            for each Z in X.neighbors - {Y}:
                enqueue(queue, (Z, X))

    revise(csp, X, Y):
        revised = false
        for x in X.domain:
            if no y in Y.domain satifies constraint for (X, Y):
                delete x from X.domain
        revised = true
    return revised
```

### 2.2. Implementação e estruturas de dados:

- A. Para a implementação dos métodos descritos foram usadas 4 principais estruturas de dados:
- B. puzzle: definida como `[[Int]]`, nested list, é usada para representar o tabuleiro inicial que deve ser resolvido;

- C. possibilityTable: definida como `[[[Int]]]`, usando também nested lists, é gerada a partir de um Puzzle e representa as possibilidades possíveis para cada célula de um tabuleiro de Comparison Sudoku. Essa é a estrutura manipulada pelas funções arcConsistency e backtrack que são usadas na solução do problema de satisfação de restrições;
- D. constraint: constraints são triplas (Int, Int, Int) onde as duas primeiras posições são inteiros que definem posições, linearizadas na forma  $i * 9 + j$ , da possibilityTable e a última posição representa a restrição entre as posições  $>$ ,  $<$  ou  $\neq$ ;
- E. Lista de constraints: define todas as restrições presentes em um tabuleiro de Comparison Sudoku usando as triplas, constraints.

### 3. Dificuldades:

Durante o desenvolvimento do código foram encontradas dificuldades relacionadas à natureza da solução escolhida e da sua implementação em uma linguagem do paradigma funcional. Além de escolhas de estruturas de dados e otimização do código.