

# Specyfikacje formalne i programy funkcyjne

## Dokumentacja projektowa - 06/12/2020

Autorzy: Tomasz Główka, Jan Zembowicz

### 1. Algorytm:

*zamień planszę na listę punktów do przeprocesowania (lista współrzędnych cyfr z planszy)  
dla każdego punktu*

*wygeneruj wszystkie możliwe kombinacje brakujących sąsiadów  
dla każdej kombinacji*

*sprawdź czy kombinacja nie zepsuła dotychczasowych punktów*

*jeśli zepsuła to sprawdź następną kombinację, jeśli jest ok to przetwarzaj następny punkt*

### 2. Main.hs

Główny moduł odpowiedzialny jest za implementację funkcji realizujących powyższy pseduo kod. Do rozwiązania użyliśmy algorytmu backtracking. Funkcja **processBoardPosition** dostaje wszystkie punkty, dla których mamy pokolorować sąsiadów. Jeśli znaleźliśmy sąsiadów dla pierwszego punktu z przekazanej listy to przetwarzamy koljeny z wywołując **processBoardPosition** z listą punktów do przeprocesowania pomniejszoną o przeprocesowany punkt. Funkcja **markAndCheckNeighbours** sprawdza, czy jeśli dodamy sąsiadów, przekazanych jako argument, do obenej puli zamalowanych pól to czy nie zepsują oni dotychczas przeprocesowanych punktów (może być tak, że jakiś poprzedni punkt, dla którego już mamy wyznaczonych sąsiadów miałby teraz więcej sąsiadów niż powinien mieć).

### 3. ReadFileModule.hs

Moduł odpowiedzialny za wczytanie pliku wejściowego z planszą do rozwiązania.

### 4. HelperModule.hs

Moduł odpowiedzialny za zestaw funkcji pomocniczych takich jak: funkcja **toInt** do konwersji String na Int, **getAllCombinations** – funkcja do generowania wszystkich kombinacji bez powtórzeń z danej listy, **getFilledBoard** – funkcja do wygenerowania planszy z rozwiązaniem, **printFilledBoard** – funkcja pokazująca planszę w konsoli.

### 5. PositionsModule.hs

Moduł odpowiedzialny za funkcje związane z przygotowaniem potrzebnych struktur do realizacji algorytmu: **getRowPositions** - funkcja przetwarzająca jeden ciąg znaków (jeden wiersz z wczytanej planszy), zwraca listę elementów: (indeks cyfry, cyfra), **getBoardPositions** – zwraca listę współrzędnych cyfr z planszy wraz z samymi cyframi, **getPossibleNeighboursPositions** – zwraca listę współrzędnych wszystkich możliwych sąsiadów dla danego punktu, **getCurrentNeighboursPositions** – zwraca listę aktualnych sąsiadów dla danego punktu, **getPointsToCheck** – zwraca listę punktów, które mogłyby zostać zepsute poprzez dodanie nowych sąsiadów (zamalowanych punktów) dla obecnie procesowanego punktu.