# **Implementing Dancing Links in Haskell**

# Project Plan

Epaphras Manaen Ndayisenga

**Abstract**. Many problems like the exact cover problem require backtracking to solve. To solve such problems however require a lot of computing, since each subset must be considered. That usually leads to wasting a lot of computation power and time, since it can be shown that not each element is necessary in the solving of such problems. Donald Knuth [1], [3] proposes an algorithm to efficiently solve such problems using a data type he calls The Dancing Links that only considers potential elements that are need to find a solution to a given problem. This algorithm can be used to solve problems such as Sudoku, Exact cover problem, N queens and many more, which are all NP-Complete. While the naïve solution to those problems require to scan all elements present, this approach only takes in the necessary elements to solve the problem. Although the time complexity of this implementation doesn’t improve in worst case, it has been shown that this algorithm can improve the real time complexity, since the worst case hardly ever happen in real world. In this thesis we implement the dancing links in Haskell programming language and test them using the Algorithm X suggested by Donald Knuth. The implementation will be presented via solving the exact cover and Sudoku problems. This thesis is divided in two parts: The first part deals with the documentations and Section two which presents the implementation codes of the dancing links and associated solved problems like exact cover problem and Sudoku.

**Keywords:** Exact Cover Problem, time-complexity, linked list, Sudoku

# Time Distribution of the Project execution

This project is distributed over a span of 4 months. Each activity has a set milestone that is to be achieved.

**08.11.2018 – 25.03.2019: Time period to developing a software system for the project**

***08.11.2018*** : ***Milestone***. ***Start Project***. Meet the Professor and get the Thesis topic

**09.11.2018 - 23.11.2018**: Project planning and research

***23.10.2018: Milestone***. Finish project planning and start Requirement analysis

***26*.11.2018-01.12.2018**: Requirement analysis and design

***01.12.2018: Milestone***. Finish Requirement analysis and Design

**03.12.2018 – 31.01.2019:** Coding and Testing

***01.02.019: Milestone***. Finish coding and testing

**4.02.2019 - 20.03.2019: Time period to writing the thesis on the project**

**Gantt chart for the project can be found in another file (bachelor project plan.png)**

# Challenges in this project

The main focus in this thesis is to implement the Dancing Links in Haskell programming language. The dancing links rely on using links or pointers to different elements. Pointers are unfortunately hardly ever used in strongly static typing functional programming language such as Haskell. The main challenge lies within defining new data types that can handle the problem, without relying to pointers. Testing the reliability of the developed codes will be done using the Algorithm X [1] to solve problems such as the Exact Cover Problem. This is however an extra work, which in itself requires much more research and understanding

# Techniques to solving the challenges

Since many of the needed functionalities are not pre-defined in Haskell, we will implement all the necessary data types ourselves. Luckily the algorithm necessary are already present, so the work is mainly centered in developing a correct data type and translate the algorithm to Haskell and that saves some time. There are however not many reference papers, since the present ones come usually from the same author.

# Reference

1. Donald Knuth. “Dancing Links (incomplete Draft)”. The art of Computer Programming. Volume 4 PRE-FASCICLE 5c. Stanford University. ADDISON-WESLEY October 29, 2018