

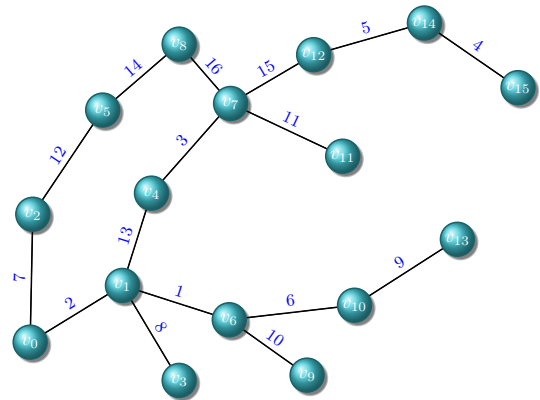
# Bachelor Thesis

## « Complexity of Self-stabilizing Algorithms Considering the Graph's Topology »

### Background

The principle of *self-stabilization* is a very general technique to design a distributed system to tolerate arbitrary transient faults. An algorithm is self-stabilizing if it can start at any possible global configuration and gain consistency in a finite number of steps by itself without any external intervention and remains in a consistent state. The state of a consistent respectively fault-free system is defined by a local predicate based on the states of the nodes.

For evaluating the complexity of self-stabilizing algorithms the maximum number of individual moves needed to reach a legitimate configuration must be determined.



### Description

Many problems in graph theory can be solved by self-stabilizing algorithms, for instance finding maximal independent sets, node colorings, or maximal matchings. However, the maximum number of moves until stabilization may depend on the topology of a graph, i.e. a given algorithm might reach a stable configuration significantly faster on a tree in comparison to a complete graph. The goal of this thesis is to analyze the complexity of several self-stabilizing algorithms with respect to the underlying graph.

For this purpose a few self-stabilizing algorithms have to be selected for analysis. For each of these algorithms the worst-case number of moves have to be determined on several graph classes, e.g. on trees, rings, complete graphs, etc.. Determining the worst-case complexity of the algorithm on a specific topology will be accomplished by means of theoretical analysis and will constitute the main part of this thesis. Since the worst-case scenarios usually do not comply with installations in practice the theoretical results will be compared with results of a self-written simulator.

### Prerequisites

Basic knowledge of Graph Theory and Discrete Mathematics

Contact: Bernd Hauck

hauck@tuhh.de

Phone: +49 40 / 428 78 – 3704

Room: 4.075