Deterministic Self-Stabilising Leader Election for Constant-Memory Programmable Matter

Jérémie Chalopin, LIS, Aix Marseille Univ., jeremie.chalopin@lis-lab.fr
Shantanu Das, LIS, Aix Marseille Univ., shantanu.das@lis-lab.fr
Maria Kokkou, LIS, Aix Marseille Univ., maria.kokkou@lis-lab.fr

Electing a unique leader is central to all distributed systems, including programmable matter. Programmable matter systems comprise of computational entities with constant size memory, called particles. We only consider programmable matter systems to be a set of particles occupying a simply connected subset of vertices of a regular triangular grid. In this context, we present a silent self-stabilising, deterministic, stationary, election algorithm. Our algorithm is the first deterministic self-stabilising algorithm for programmable matter to always stabilise to a configuration with a unique leader, under some fairness conditions [2]. Our result is surprising as it is known that silent self-stabilising algorithms for election in general distributed networks require $\Omega(\log n)$ bits of memory per node, even for ring topologies [1].

First, we present a proof labelling scheme ensuring the existence of a unique leader. Our certificate orients the edges of the network and a configuration is valid when : every edge is oriented, each particle has at most three outgoing edges, outgoing edges appear consecutively around each particle and there are no directed triangles. Then we design a simple algorithm to distributively compute such a certificate. We prove that a configuration with a unique sink is always obtained using this algorithm.

Theorem 1. *Starting from an arbitrary simply connected configuration any Gouda fair execution of our algorithm eventually stabilises to a valid configuration containing a unique sink.*

The proof of correctness of our algorithm relies on the special geometric properties of the triangular grid.

Références

[1] Dolev, S., Gouda, M. & Schneider, M., *Memory requirements for silent stabilization*, Acta Informatica 36, 447–462 (1999). https://doi.org/10.1007/s002360050180

[2] Gouda, M.G., *The Theory of Weak Stabilization*, Lecture Notes in Comp. Sci., vol 2194 (2001). https://doi.org/10.1007/3-540-45438-1_8