

Proofs for Optimality of Sorting Networks by Logic Programming

Michael Codish¹, Luís Cruz-Filipe², Michael Frank¹, and Peter Schneider-Kamp²

¹ Department of Computer Science, Ben-Gurion University of the Negev, Israel
`{mcodish, frankm}@cs.bgu.ac.il`

² Department of Mathematics and Computer Science, University of Southern Denmark, Denmark
`{lcf, petersk}@imada.sdu.dk`

Abstract. We present a computer-assisted non-existence proof of nine-input sorting networks consisting of 24 comparators, hence showing that the 25-comparator sorting network found by Floyd in 1964 is optimal. As a corollary, we obtain that the 29-comparator network found by Waksman in 1969 is optimal when sorting ten inputs. This closes the two smallest open instances of the optimal size sorting network problem, which have been open since the results of Floyd and Knuth from 1966 proving optimality for sorting networks of up to eight inputs. The entire implementation is written in SWI-Prolog and was run on a cluster of 12 nodes with 12 cores each, able to run a total of 288 concurrent threads, making extensive use of SWI-Prolog's built-in predicate `concurrency/3`. The search space of 2.2×10^{37} comparator networks was exhausted after just under 10 days of computation. This shows the ability of logic programming to provide a scalable parallel implementation while at the same time instilling a high level of trust in the correctness of the proof.