

All Sorts of Permutations

Jan Christiansen

Fachhochschule Flensburg

Fachbereich 3

jan.christiansen@fh-flensburg.de

Nikita Danilenko

CAU Kiel

Institut für Informatik

nda@informatik.uni-kiel.de

Sandra Dylus

CAU Kiel

Institut für Informatik

sad@informatik.uni-kiel.de

The combination of non-determinism and sorting is mostly associated with permutation sort, a sorting algorithm that is not very useful for sorting and has an awful running time.

In this presentation we look at the combination of non-determinism and sorting in a different light: given a sorting function, we apply it to a non-deterministic predicate in order to gain a function that enumerates permutations of the input list. We get to the bottom of necessary properties of the sorting algorithms and predicates in play. Hereby, the following interesting questions arise. Does the resulting function enumerate *all* permutations of the input list? Does the resulting function enumerate *only* all permutations and no other bogus results? Does the resulting function enumerate *exactly* all permutations and no duplicates?

Although improbable, to the best of our knowledge this connection has been first noted in the context of functional logic programming on the mailing list of the functional logic programming language Curry in a thread by Fischer and Christiansen [2009].

References

Sebastian Fischer and Jan Christiansen. Curry mailing list, July 2009. URL <http://www.informatik.uni-kiel.de/%7Emh/curry/listarchive/0781.html>.