Permutations

Jacek Karwowski and Amr Sabry

October 15, 2019

1 Introduction

Permutations are pervasive:

- quantum unitary operations; at their heart is a classical permutation
- hott: equivalences are core; when restricted to finite sets we talk about permutations
- Permutation-based cryptography https://permutationbasedcrypto.org/2018/program.html and https://pdfs.semanticscholar.org/a780/6067e6fa3083ac946a9b2effd27d09ed6cde.pdf
- other classical examples: http://www.sfu.ca/~jtmulhol/math302/notes/302notes-May29-2015.pdf
- Paper-folding https://pdfs.semanticscholar.org/dcbf/ef94957a9cc3082db7bcb2e9aed5f8fc588a.pdf and http://news.mit.edu/2017/algorithm-origami-patterns-any-3-D-structure-0622
- origami materials https://journals.aps.org/prx/pdf/10.1103/PhysRevX.7.041070
- DNA http://www.dna.caltech.edu/Papers/surface_CRNs_DNA20.pdf

So how do you program with permutations? We develop a nice programming language in which every expression is a valid permutation; all permutations can expressed; includes sound and complete meta-programs for reasoning about permutations; can express compositions of permutations along sums and products; can transfer algorithms/proofs across several representations of permutations.