Swaps of Swaps Grammar

by Sven Nilsen, 2020

In this paper I show how to generate the permutation grammar from swaps of swaps.

An equivalence between equivalences can be encoded as swaps of swaps:

(a ~= b) ~= (c ~= d)

One can also visualize this using a diagram with arrows between objects and arrows.
However, it can be a bit difficult to understand the permutations that are generated.

One important thing to remember:
Swaps of swaps generate a grammar of permutations, not just a single permutation.

Individually, one can write down the swap grammar:

(ab)cd = abcd + bacd	a ~= b
ab(cd) = abcd + abdc	c ~= d

Notice that \`abcd\` occurs as a sentence in each grammar. This is the identity map.
This is an unchanged version of the input.

Instead of using the identity map as input, one can think about swaps of swaps as taking the output from individual swaps as input:

abcd + bacd + abdc

Now, apply every rule of individual swaps again, for every input and collect new outputs:

(ab)cd = abcd + bacd	No new outputs
(ba)cd = bacd + abcd	No new outputs
(ab)dc = abdc + **badc**	\`badc\` is new output
ab(cd) = abcd + abdc	No new outputs
ba(cd) = bacd + badc	\`badc\` already added as new output
ab(dc) = abdc + abcd	No new outputs

This results in the following grammar:

abcd + bacd + abdc + badc

This grammar can be simplified to:

(ab)(cd)