Generating linear extensions of the power set

Alexis Poindron

February 22, 2024

Two well-known and simple linear extensions of the power set are the following:

(Left) For $k = 1..n$, add states of cardinality k in any order.

Example:

$$L = [[],\ \underbrace{[1],[2],[3],[4]}_{\text{with permutations}},\ \underbrace{[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]}_{\text{with permutations}},\ \underbrace{[1,2,3],[1,2,4],[1,3,4],[2,3,4]}_{\text{with permutations}},\ [1,2,3,4]]$$

(Middle) The "hypercube duplication procedure". Start from $L = [[]]$. For $k = 1..n$, duplicate the states of L by adding k.

Example:

(i) $L = [[]]$

(ii) $L = [[], [1]]$

(iii) $L = [[], [1], [2], [1,2]]$

(iv) $L = [[], [1], [2], [1,2], [3], [1,3], [2,3], [1,2,3]]$

(v) $L = [[], [1], [2], [1,2], [3], [1,3], [2,3], [1,2,3], [4], [1,4], [2,4], [1,2,4], [3,4], [1,3,4], [2,3,4], [1,2,3,4]]$.

The "hypercube duplication" linear extension is appreciated for the following advantage: in order to add a new agent, we do not need to reshuffle the list: it suffices to duplicate the list by adding the new agent to all states. On the contrary, the "cardinality linear extension", the left one, requires maximal reshuffling when adding an agent. The two aforementionned linear extensions are regarded as opposite[1]. Algorithm 1 is based on the previous observation.

Our conjecture is that, up to a relabelling of nodes either on the middle linear extension, or on the resulting linear extension, Algorithm 1 produces a random linear extension uniformly over the set of linear extensions of the power set. An example is given in Figure 1.

[1] Just like the majority 0-1 capacity ($f(x) = 1 \Leftrightarrow |x| \geq n/2$) is the "opposite" of the singleton winner game capacity ($f(x) = 1 \Leftrightarrow x_i = 1$); the majority capacity has also a maximal entropy for a balanced 0-1 capacity (we call balanced a 0-1 capacity which has the same number of 0-values and 1-values), in the sense that it requires a maximal number of questions to identify its antichain.

Algorithm 1 Linear extension of the power set

Generate the cardinality linear extension and the hypercube duplicate linear extension
In the cardinality linear extension, operate random and uniform permutation within each group of identical cardinality
In the hypercube duplication linear extension, operate a random relabelling of nodes
Start from one of the linear extensions - say the "cardinality" one
Draw a random sequence of heads and tails (H head=switch; T tail=stay)
Navigate between the two linear extensions according to the random draws. Go upwards by picking the first state that has not been attributed yet.

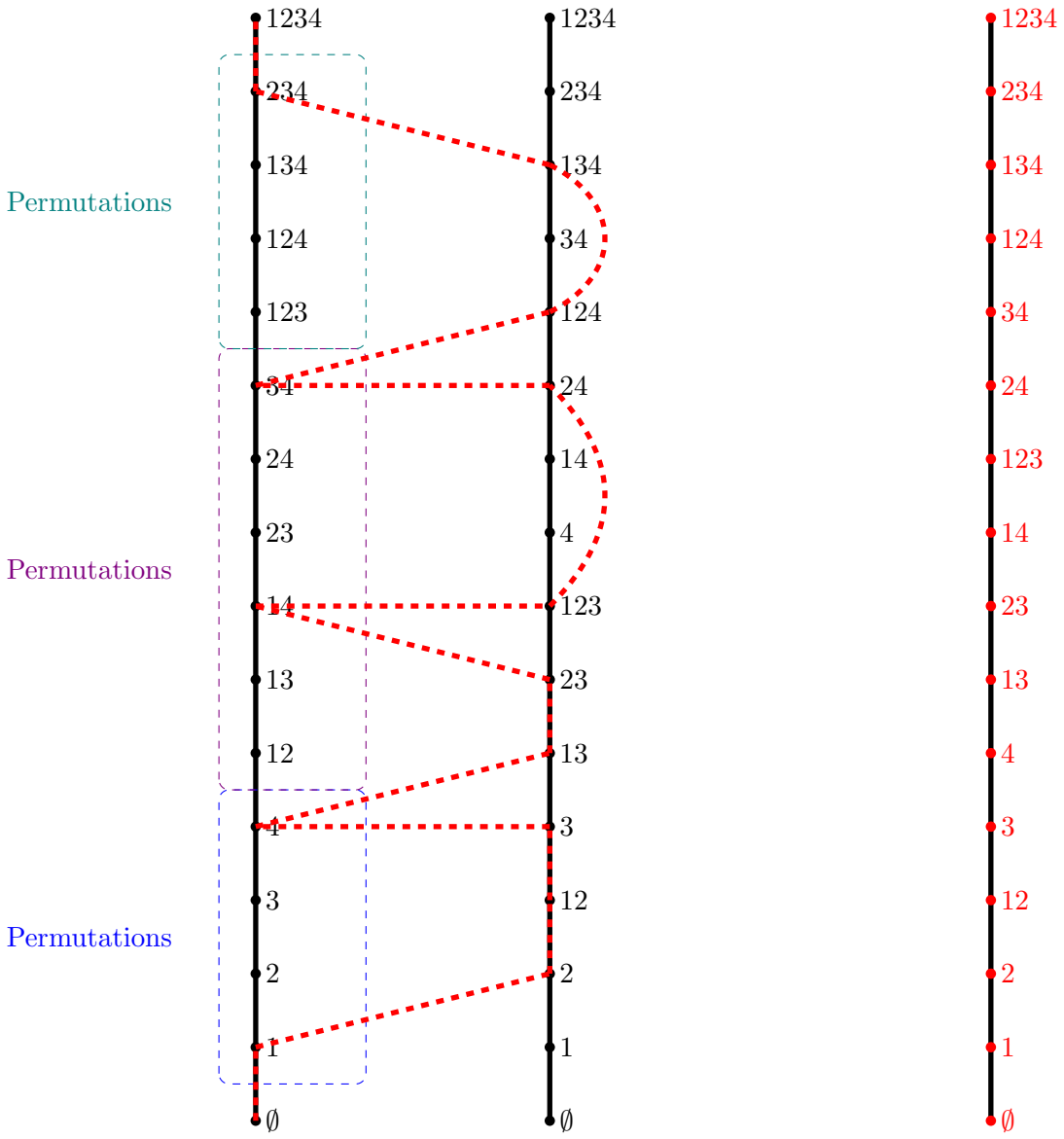

Figure 1: Left: the "cardinality" linear extension. Middle: "hypercube duplication" linear extension. Right: resulting linear extension, where the red walk corresponds to HTHHTTHTTHTTHTH in Algorithm 1.