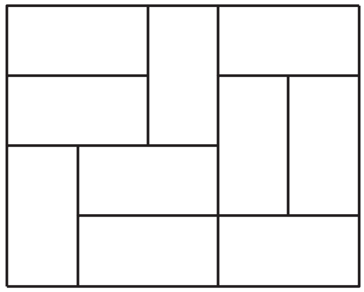
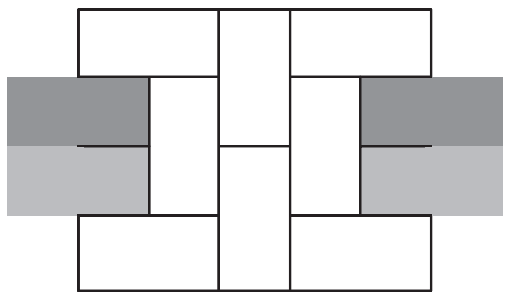
Arranging Tatami Mats

Patch Kessler, December 23, 2008

Traditional Japanese floors are covered with 1×2 *tatami* mats.



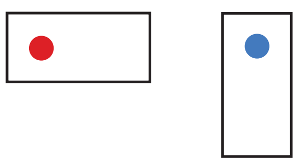
Here we ponder mat configurations on floors that are $m \times n$. Some configurations are symmetrical and some aren't- some seem composed of smaller more basic configurations while others seem irreducible. Allowing mats to extend past a floor boundary and reappear on the opposite side increases the number of possible configurations.



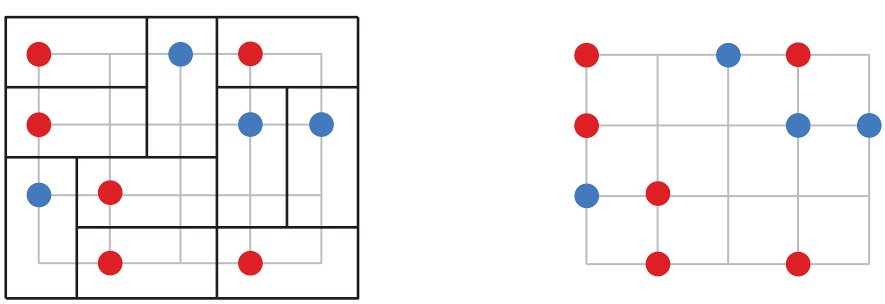
Depending on how this is done, the floor can become a cylinder, a Möbius strip, a torus, or a Klein bottle. We consider none of these cases here- instead we focus on an algorithm for generating all possible configurations on an $m \times n$ floor with inviolable boundaries.

Algorithm

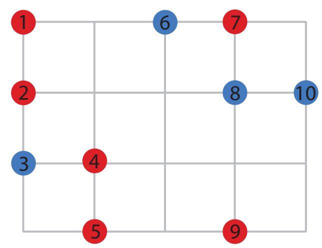
An $m \times n$ floor contains $\frac{mn}{2}$ tatami mats, each of which is either horizontal or vertical.



Every configuration corresponds to a collection of red and blue dots on a grid.

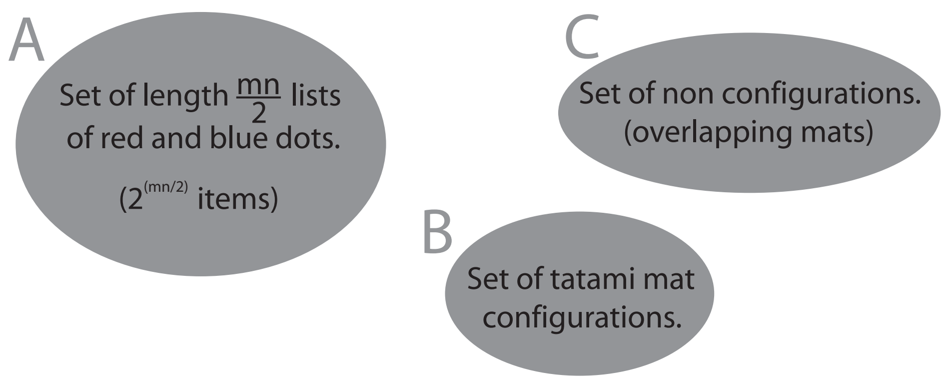


In fact, every configuration corresponds to a *list* (lists have order) of red and blue dots. We order the dots by looking for the *next* dot as far to the left as possible. If multiple dots are equally far to the left, then we pick the one that is as far up as possible.



A configuration can be reconstructed from its corresponding list of dots. Start by placing the first dot in the upper left corner of the grid. The tatami mat corresponding to this dot will cause a total of two grid nodes to be off limits to future dots. Place the n^{th} dot as far to the left, and then as far up as possible, and mark the two nodes covered by its corresponding tatami mat off limits.

Although every configuration corresponds to a list of dots, not every list corresponds to a configuration, (in these cases the construction process leads to overlapping mats, or to mats that cross the floor boundary). There are three sets of objects to deal with.



It is straightforward to create a mapping f from A to $B \cup C$; we do this with Matlab code. The restriction of f to the pre-image of B is bijective, (i.e., the code doesn't return multiple copies of the same configuration). In our code we further discard configurations so that no two differ by a rotation or a reflection (or a combination of these). Here is sample output in the case of 4×4 floors.