The problem I discussed during our meeting today would, more precisely, be something like this.

Given a function F from $\mathbb{N}^d \to X$, define the n-neighbourhood of a point $x \in \mathbb{N}^d$ to be $f_{x,n} : [-n,n]^d \to X$, $f_{x,n}(y) = f(x+y)$ (where the interval is a subset of $\mathbb{N}$.

Given a prior probability distribution on X and a weighting function $w : \mathbb{N}^d \times (X^{[-n,n]^d}) \to \mathbb{R}$, define the associated probability distribution on $X^{\mathbb{N}^d}$ by setting its marginal distribution on any finite set of points P equal to the limit as the finite region $R \subset \mathbb{N}^d$ goes to $\mathbb{N}^d$ of the marginal distribution on P of the distribution obtained by multiplying the prior (given by the product distribution of $|R|$ copies of X) by the posterior (given by the product of $w(x, \text{the } n\text{-neighbourhood of } x)$ for every x such that the neighbourhood is contained in R) and re-normalising, assuming the limit exists (which it sometimes doesn't).

The problem, then, is to provide an algorithm that, given n, the prior distribution on X, the weighting function w, a tolerance value $\epsilon > 0$, a point x and a trace/source of randomness, will output an element of X in such a way that the joint distribution obtained by evaluating the algorithm with the same trace for all points in a region P has an error of at most $\epsilon|P|$ (in the sense that the information of the algorithm's output distribution relative to the correct distribution on P is at most $\epsilon|P|$).

There are a lot of omitted details in terms of how everything is encoded, but I don't expect that to matter all that much. Also, I'm reasonably confident that this problem doesn't have a full solution (there's probably some sort of way of encoding halting problems in this form), but it might well have a partial solution given some sort of restriction on w.

The way of defining the global distribution in terms of local weights that I used here is standard in statistical mechanics. A solution for $n = 1$ would imply a solution for arbitrary n.

Example One application of such an algorithm would be sampling a state of the Ising model. In that case, set $d = 2, n = 1$, $X = \{up, down\}$, the prior distribution on X giving each of the 2 states $1/2$ chance, and

$$w(f) = \exp\left(\frac{(f(-1,0) = f(0,0) \, ? \, 1 : -1) + (f(0,-1) = f(0,0) \, ? \, 1 : -1)}{T}\right)$$

where T is the temperature parameter. This means that fields where the state at adjacent points is equal are more likely. For temperatures above some critical temperature (I can't remember the value, but it doesn't really matter), the true distribution of results has patches of each state, with the patches getting larger as T decreases.

There is therefore a correlation between the states at different points, that falls off exponentially with distance. (The behaviour below the critical temperature is more strange and discontinuous, and that case should possibly be

excluded from consideration). The algorithm has to approximate this distribution, but it can't generate the entire infinite grid, so it has to generate finite portions at a time. When generating (separately) the values for two different points, it is given the coordinates of each point, but it's given the same source of randomness for both of them. The correct distribution for each point individually is just 1/2 chance of each state, but it has to (for sufficiently low ϵ) use the shared randomness to generate correctly correlated results.