The Rado Graph and Maps from the lattice to the interval.

12/7 (1

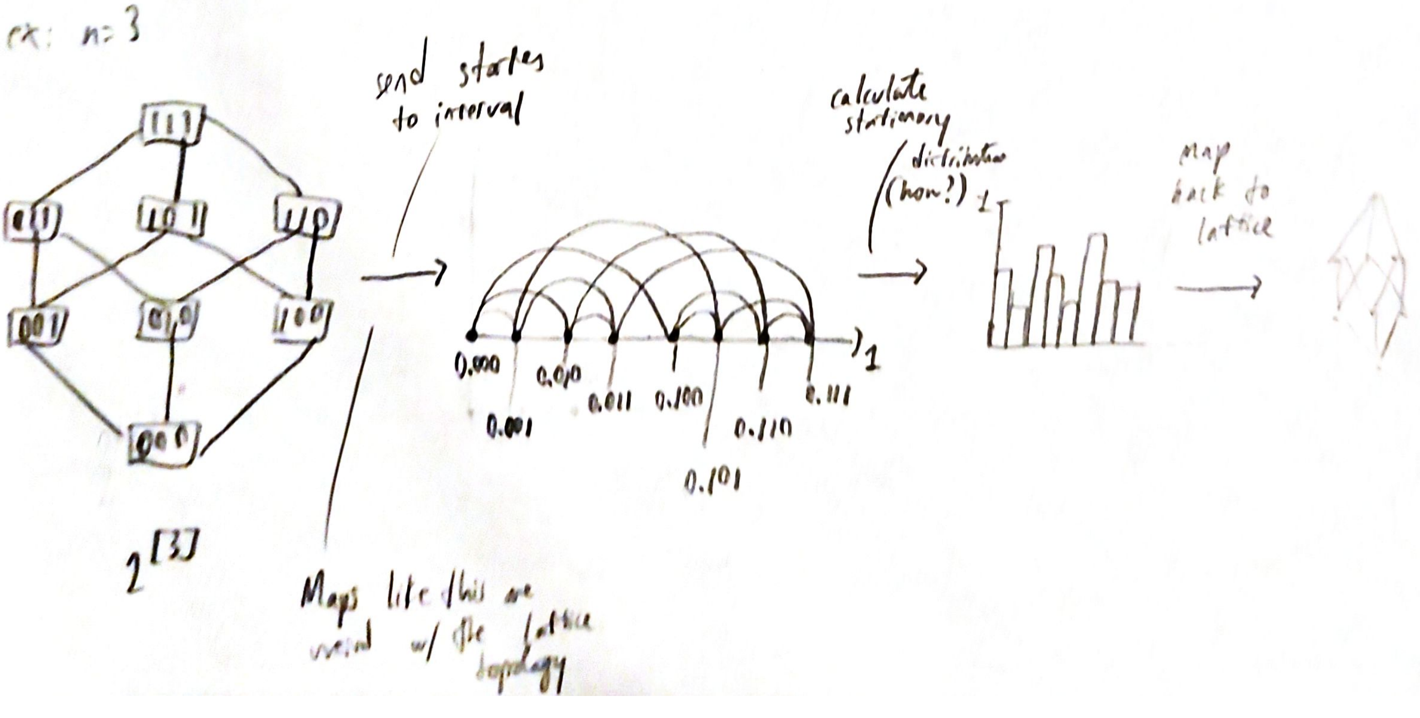
This is not the same problem as the one I've been considering, but I want it written out so I can come back to it the next time I have a desire to.

Let G be the Rado graph (which I believe is represented by graphon $A \equiv 1/2$? (can check this), and let G_n be the induced subgraph of G on $[n]$. Let A_n be the adjacency matrix. Let L be an integer corresponding to an l% load on $[n]$, and let $B = B_L$. Also, let $2^{[n]} = [1]_0^n$, and finally, let $BrAn(A, B, \varepsilon)$ be the ergodified brancher-annihilator process, with $0 < \varepsilon << 1$.

~~BrAn imposes a directed graph on~~ $\cancel{2^{[n]}}$

BrAn is an ergodic Markov chain on $2^{[n]}$, but the statespace is so large that analyzing the stationary distribution is very difficult. What I've been wanting to try is a map such as the following:

ex: $n=3$

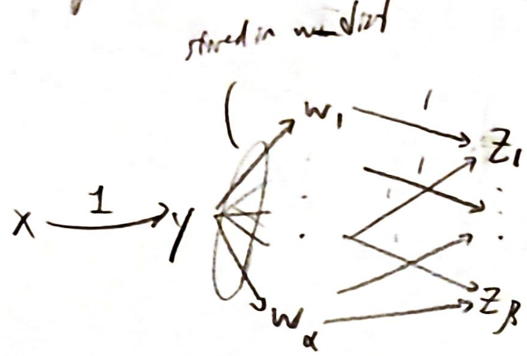


$$2^{[n]} \xrightarrow{A} [n]_0^n \xrightarrow{+b} [n+1]_0^n \xrightarrow{f(\cdot)} 2^{[n]}$$

We've seen before that

Appendix 1

Basically, ~~we~~ we only have brute force to calculate the transition probabilities:




```
For all x ∈ 2^[n]:
    y = Ax;  w_avbl = {},
    for all b in B:
        w = y+b
        w_avbl U= {w}
    w_dict = {w: [] for w in w_avbl}
    for all b in B:
        w = y+b
        w_dict[w].append(IP(b_t = b | x_t = x))
    for w in w_dict.keys():
        w_dict[w] = sum(w_dict[w])
    z_dict = {z: {} for z in 2^[n]}
    for w in w_dict.keys():
        z = f(w)
        z_dict[z] U= {w}
    for z in 2^[n]
        T(x,z) = sum([w_dict[w] for w in z_dict[z]])
```

• Check in the future and perhaps translate into math notation or pseudocode.

Appendix 2 is proof that $\mathbb{P}(\tau > n)$ is a function of π