# Kodowanie i generowanie układów statystycznych za pomocą algorytmów probabilistycznych.

**Jarosław Duda**

Instytut Fizyki
Wydział Fizyki, Astronomii i Informatyki Stosowanej
Uniwersytet Jagielloński

Praca magisterska
opiekun: prof. dr hab. Karol Życzkowski
Kraków 2006

# Spis treści



# 1 Wstęp

W teorii informacji, gdy mamy pewne informacje o układzie, pozostałe parametry ($p$) przyjmuje się korzystając z *zasady maksymalnej niewiedzy* - maksymalizujemy entropię (która charakteryzuje wkład elementów danego typu):

$$\#\rho(p) \propto \exp(N * H_p) \tag{1}$$

czyli przy przybliżeniu naszego nieskończonego układu układem skończonym o wielkości $N$, wkład stanów o danych parametrach $p : \rho(p)$ zachowuje się proporcjonalnie do eksponenty z iloczynu wielkości układu i średniej entropii $H_p$ dla tych parametrów. Czyli przy przejściu $N \to \infty$ istotne są tylko układy z parametrami maksymalizującymi entropię.

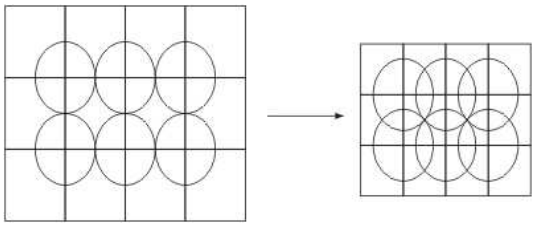
W niniejszej pracy będziemy się zajmować modelami zachowywującymi wysoką symetrię przestrzeni. Okaże się, że dzięki temu wystarczy ograniczyć się do elementów, które też będą (statystycznie) zachowywały te symetrie - pozostałe zostaną przez nie zdominowane - nazwę takie ograniczenie *opisem statystycznym p*. Następnie ograniczymy się, jak podpowiada zasada maksymalnej niewiedzy, do elementów dla których ów opis statystyczny będzie maksymalizował entropię $H_p$ -*opis typowy*, pozostałe są nieistotne.

Wspomniana entropia jest to średnia pojemność informacyjna, znaleziony opis statystyczny pozwoli nam skonstruować algorytm probabilistyczny do zapisywania dowolnych informacji w tym modelu z daną ilością bitów na węzeł (entropią).

Niniejsza praca podaje pełną metodologię postępowania z układami na sieci periodycznej (w każdym węźle sieci jest element z alfabetu $\mathcal{A}$), w których narzucono pewne periodyczne *więzy* - zestaw skończonych układów, które są zabronione. Gdzie przez *periodyczne* rozumiem, że zbiór więzów jest niezmienniczy ze względu na translacje.

W kolejnych rozdziałach przeprowadzę ogólną dyskusję nad jedno, a potem wyżej wyżej wymiarowymi modelami z silnymi więzami. W celu lepszego zrozumienia zawartych tu idei, wszystko robię pod przykład: *Hard Square model* - $X = \mathbb{Z}^2$, $\mathcal{A} = \{0, 1\}$, więzy: nie może być dwóch sąsiednich (każdy węzeł ma 4 sąsiadów) jedynek. Widać że bez więzów można w tym modelu zapisać średnio 1bit/węzeł. Po wprowadzeniu więzów okaże się że pojemność informacyjna spadnie do $H \cong 0.5878911617753406$ bita/węzeł. Dla tego modelu przeprowadziłem też pełne obliczenia. Większość używanych w tym przypadku technik będzie się naturalnie przenosiła do ogólnych problemów.

**Przykład zastosowania**:
Przy zapisywaniu informacji na jakiejś powierzchni, mamy lokalnie siatkę $\mathbb{Z}^2$, w której każdym węźle dokonujemy namagnesowania lub nie - czyli w każdym węźle zapisujemy 1 bit informacji. Problemem technologicznym jest zmniejszenie tej plamki (stałej sieci). Niniejsza praca pokazuje że zwiększenie pojemności informacyjnej powierzchni można też uzyskać przez dokładniejsze pozycjonowanie głowicy(przy takiej samej wielkości plamki): np. zmniejszamy stałą sieci $\sqrt{2}$ razy - teraz plamki sąsiednie zachodzą na siebie - no więc zabraniamy namagnesowania 2 sąsiednich więzów - czyli postawienia na sieci 2 sąsiednich jedynek - wspomniany właśnie model.



Rysunek 1: Przeskalowanie sieci bez zmiany "plamki".

Dostajemy $\sqrt{2}^2 * H \cong 1.17$ bitów/poprzedni węzeł, czyli zyskaliśmy 17%. Analogicznie bardziej zmniejszając stałą sieci (dokładniejsze pozycjonowanie) możemy zyskać jeszcze więcej średniej informacji.
Na przykład w przypadku jednowymiarowym: po jedynce musi być $k$ zer (bierzemy $k + 1$ razy dokładniejsze pozycjonowanie) dostajemy zysk(%):

| $k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| zysk(%) | 0 | 39 | 65 | 86 | 103 | 117 | 129 | 141 | 151 | 160 | 168 | 176 | 183 | 190 |

W **drugim rozdziale** podam pełne ścisłe rozumowanie dla *modelu jednowymiarowego*. Równolegle w tym rozdziale będę wprowadzał aparat teorioinformacyjny - jak oceniać pojemność informacyjną układu i dlaczego jest ona równa *entropii Shanona*.
W tym przypadku zawsze możemy sprowadzić model do przestrzeni ciągów elementów z pewnego alfabetu, takich że po danym symbolu $a$ mogą następować tylko symbole z $M(a)$ - model jest określony przez *macierz więzów*: $M$.
Tutaj nasz algorytm probabilistyczny do generowania układów typowych, to będzie zwykły proces Markowa - poruszamy się np. z lewej do prawej i rozkład prawdopodobieństwa warości kolejnego elementu mamy na podstawie wartościowania ostatniego (problem inicjalizacji poruszony jest na końcu pracy). Na podstawie $M$ wskażemy macierz stochastyczną procesu i udowodnimy, że przy zawężeniu się do elementów wygenerowanych tym procesem Markowa dostaniemy praktycznie wszystkie elementy - pozostałe zostaną przez nie zdominowane. Intuicyjnie: wystarczy proces Markowa - czyli znajomość tylko ostatniego elementu, ponieważ poprzednie są przez niego zasłonięte - nie ma sensu dokładać głębszych korelacji, czyli wiedzy o układzie. Później uogólnimy ten topologiczny argument na wyższe wymiary.

W **trzecim rozdziale** pokażę jak praktycznie zastosować znalezione algorytmy probabilistyczne do zakodowania dowolnej informacji. Mianowicie trzeba uogólnić system dwójkowy, optymlnie kodujący ciąg bitów, w którym $P(0) = P(1) = 0.5$ i nie ma żadnych korelacji na system w którym dalej nie ma żadnych korelacji, ale rozkład prawdopodobieństwa kolejnego symbolu nie jest już jednorodny i może się zmieniać.

W **czwartym rozdziale** wprowadzę formalny aparat do badania ogólnych modeli. Wprowadzę tam słabe warunki - $X = \mathbb{Z}^n$, wiezy skończenie zasięgowe i translatywnie niezmiennicze oraz, że po pewnym "pogrubieniu" zbioru, wartościowania brzegu tego pogrubienia, nie kolidują z wartościowaniami zbioru. Te warunki wystarczą do udowodnienia istnienia entropii dla tego modelu.

W **piątym rozdziale** uzasadnię że jeśli ograniczymy się do elementów danych opisem statystycznym, który dostajemy z uśredniowania po wszystkich elementach w przestrzeni (ospisowi typowemu), to dostaniemy praktycznie wszystkie elementy - biorąc losowy elementy - z prawdopodobieństwem 1 będzie on typowy. Gdzie przez opis statystyczny rozumiem funkcję przyporządkowującą dowolnemu *kształtowi* rozkład prawdopodobieństwa możliwych na nim wartościowań: *wzorców*.

Niestety mimo wskazania wiele argumentów na istnienie owej średniej po

wszystkich elementach, nie udało mi się tego udowodnić - przyjmę to jako odgórne założenie i uzasadnię że jest ono równoważne znikaniu korelacji dalekozasięgowych.

Wprowadzimy też alternatywną definicję typowości (LTP), mówiącą że po ustaleniu wrtościowania na brzegu zbioru, rozkład wartościowań wewnątrz jest już jednorodny.

Dla danego opisu (niekoniecznie typowego) i pewnego ciągu wypełniającego przestrzeń można skonstruować elementy statystyczne dla tego opisu - idziemy po węzłach według tej kolejności i na podstawie wartościowania poprzednich (no i opisu) dostajemy rozkład prawdopodobieństwa według którego losujemy wartościowanie węzła.

W **szóstym rozdziale** badamy uogólnione takie metody generowania elementów - już nie zakładamy, że poprzednich elementów była skończona ilość. Przyjmujemy też, że rozkład prawdopodobieństwa chcemy obliczyć na podstawie sąsiednich węzłów. Nie dostaniemy w ten sposób typowego opisu, ale jednak uda nam się ograniczyć straty informacji poniżej 5%.

Zbadam 2 algorytmy: wypełnienie po zbiorach wewnątrz których nie działają więzy oraz zupełnie losowy. Pokażę też jak losowo zmieniać taki układ, żeby dostać jednorodny rozkład dla skończonej podprzestrzeni.

W **siódmym rozdziale** pokażę w końcu jak podejść dowolnie blisko do optymalnego algorytmu - zamienię model na jednowymiarowy przez zawężenie przestrzeni do paska o skończonej szerokości i nieskończonej długości - czyli problemu jednowymiarowego. Pokażę tam też wyniki numeryczne, pokazujące że rzeczywiście szybko zbiegamy do optymalnego algorytmu.

### Motywacja fizyczna i perspektywy

W fizyce statystycznej potrafimy badać własności statystyczne na przestrzeni zwykle nieprzeliczalnie wielu stanów. Jednak wskazanie konkretnego takiego stanu jest zadaniem dalece niebanalnym. Zwykle robi się to metodami monte-carlo, czyli dokonujemy wielu małych zmian zgodnie z rozkładem bolzmanowskim - jednak żeby dostać dobry generator losowy, dobrze by było każdy punkt zmieniać wiele razy - czym więcej tym lepiej. Metody z tej pracy pokazują jak dla danej dokładności generatora znaleźć metodę generującą typowy układ w jednym przebiegu. Pozwalają też np. zakodować taki układ używając minimalnej ilości klasycznego nośnika. Wystarczy znaleźć opis statystyczny.
Na razie pokazuję dokładną metodę jak znaleźć opis statystyczny, formalizm tylko dla silnych więzów. Popatrzymy się na dowolny układ fizyki statystycznej

na dyskretnej sieci, w której oddziaływania są skończenie zasięgowe. Teraz gdy przejdziemy z temperaturą do 0, zostawimy tylko układy, które w otoczeniu każdego węzła minimalizują lokalnie energię - dostajemy silne więzy. W zależności od modelu - unormowana entropia (ilość układów) może w tej granicy spadać do 0 (np. model Isinga), albo też dążyć do pewnej niezerowej granicy(np. Hard Square) - entropii resztkowej. Czyli metody z tej pracy pomagają w badaniu układów w zerowej temperaturze.
Czy w takich układach można by kodować informację? W niezerowej temperaturze ruchy termiczne ją uszkodzą. Metoda kodowania z trzeciego rozdziału jest na razie zupełnie nieodporna na błędy. Widać, że dobrze byłoby ją rozwinąć o możliwość dołożenia redundancji, w celu późniejszej korekcji błędów. Wtedy przy niewielkich ruchach termicznych takie kodowanie mogłoby mieć sens.
Planuję te metody rozwinąć na inne więzy lokalne/statystyczne - np. słabe (np. dodatkowo $p(0) = 0.3$) i bolzmanowskie. Dzięki temu będzie można np. generować układy w danej temperaturze i składzie procentowym składników.
Więzy lokalne/statystyczne spotykamy jeszcze w teorii grafów losowych. Rozwinięcie metod z tej pracy mogłoby pozwolić na lepsze zrozumienie na przykład rozwoju internetu.

# 2 Model jednowymiarowy

W tym rozdziale będziemy się zajmować następującym modelem:

**Definicja 1.** *Modelem jednowymiarowym* będziemy nazywać trójkę $(X, \mathcal{A}, M)$*:*
*przestrzeń* $X \subset \mathbb{Z}$
*alfabet* $\mathcal{A}$ -skończony zbiór symboli
*więzy* $M : \mathcal{A}^2 \to \{0, 1\}$

wtedy *elementy to* $V(X)$*, gdzie dla* $A \subset X$*:*

$$V(A) := \{v : A \to \mathcal{A} | \forall_{i \in X \cap (X-1)} M(v_i, v_{i+1}) = 1\} \tag{2}$$

Uzasadnię krótko, że dowolny translatywnie niezmienniczy, skończenie zasięgowy model można łatwo sprowadzić do powyższego (o zasięgu więzów = 1).
Mianowicie niech $l + 1$ będzie największym zasięgiem w więzach - np. $v_k = a \Rightarrow v_{k+l+1} = b$. Wtedy jako nowy alfabet bierzemy $\mathcal{A}^l$ - grupujemy $l$ kolejnych symboli. Teraz możemy wprowadzić macierz jak wyżej:

$$M_{(v_i)_{i=1..l}(w_i)_{i=1..l}} = 1 \Leftrightarrow$$
$$\forall_{i=2..l}\ v_i = w_{i-1} \text{ oraz ciąg } v_1 v_2 ... v_l w_l \text{ jest zgodny z więzami}$$

Czyli wystarczy ograniczyć się do zdefiniowanego modelu.

Zastanówmy się ile informacji możemy zapisać w takich ciągach, powiedzmy długości $k$?
Oznaczmy przez $N_k = \#V(\overline{k})$ ($\overline{k} := \{0, 1, ..., k-1\}$) ilość takich ciągów - uzyskanych z powyższego modelu dla $k$ - elementowej odcinka. Czyli informacja jaką możemy tu umieścić to wybór jednego z $N_k$ ciągów.
Do wskazania jednego z $2^n$ elementów potrzebujemy $n$ bitów.
Ogólnie będziemy badać granicę z $n \to \infty$ - możemy przyjąć że
**Do wskazania jednego z $n$ elementów potrzeba $\lg n$ bitów**.
gdzie $\lg \equiv \log_2$ - przez całą pracę będziemy porównywać z układem dwójkowym, w którym kolejny symbol oznacza kolejny bit informacji. Jest to wygodna konwencja przy porównywaniu z układem dwójkowym.

W naszym przypadku, średnia ilość informacji przypadającej na jeden symbol, to

$$H^k := \frac{lg(\#V(\overline{k}))}{k} \tag{3}$$

Policzymy $H := \lim_{k\to\infty} H^k$
Oznaczmy $c_a^k := \#\{v \in V(\overline{k}) : v_{k-1} = a\}$ $\quad c^k := (c_a^k)_{a\in\mathcal{A}}$
Ciągi długości $k+1$ uzyskujemy z ciągów o długości $k$: $\boxed{c^{k+1} = c^k \cdot M}$
Czyli $H$ - dominująca wartość własna $M$, w której wektorze własnym niezerowy wkład ma $c^1 = \{1, 1, ..., 1\}$
Przypomnijmy:

**Twierdzenie 2** (Frobeniusa - Perrona)**.** *Macierz nieredukowalna o rzeczywistych, nieujemnych współczynnikach, ma dominującą rzeczywistą, nieujemną wartość własną i odpowiada jej wektor własny o rzeczywistych, dodatnich współczynnikach.*

Redukowalność oznacza, że możemy tak podzielić współrzędne na przynajmniej 2 podzbiory, że macierz nie ma wyrazów niezerowych między tymi podzbiorami (nie miesza ich).
W naszym przypadku redukowalność oznaczałaby że startując z jednego z podzbiorów alfabetu, nigdy z niego nie wyjdziemy - najpierw wybieramy składową alfabetu, a potem poruszamy się wewnątrz tej składowej, używając macierzy nieredukowalnej. Możemy pominąć te przypadki w rozważaniu.
Czyli w naszym (M jest nieredukowalne) przypadku $H = lg\lambda$, gdzie $\lambda$ jest dominującą wartością własną M.

Zastanówmy się teraz nad znalezionym, odpowiednio unormowanym wektorem własnym: $C^T M = \lambda C^T$ , gdzie $\sum_{a\in\mathcal{A}} C_a = 1$.
Jest to oczywiście rozkład symboli w takich nieskończonych ciągach...
Nie jest to prawdą: **jest to rozkład prawdopodobieństwa symboli na końcu**

**takich ciągów**!

Spróbujmy policzyć rozkład symboli wewnątrz nieskończonych ciągów z danego modelu. Tym razem nie będziemy rozszerzać ciągu w jedną stronę, tylko w obie. Dla pewnego $(v_i)_{i=0..m-1}$ definiujemy

$$(c_v^k)_{a,b} := \{w \in V(\overline{m+2k}) : w_k w_{k+1} ... w_{k+m-1} = v,\ w_0 = a,\ w_{2k+m-1} = b\} \tag{4}$$

Takie $v$ będziemy nazywać *wzorcem*, zbiór na którym jest określony $(\overline{m})$ - jego *kształtem*, a skrajne elementy - jego *brzegiem* - np. $\grave{v} := v_0,\ \acute{v} := v_{m-1}$

Teraz $\quad c_v^{k+1} = M \cdot c_v^k \cdot M$

Dla dominującej wartości własnej mamy wektory własne:

$$C^T M = \lambda C^T \quad (= C^T\ (\lambda C C^T)) \qquad\qquad MD = \lambda D \quad (= (\lambda D D^T) D) \tag{5}$$

Gdzie tym razem przyjęliśmy normalizację $C^T C = D^T D = 1$.
Ponieważ jest to dominująca wartość własna, dla dużych $k$ możemy przyjąć:

$$c_v^k \cong \lambda^{2k} C C^T c_v^0 D D^T = \lambda^{2k} C C^T e_{\grave{v}} e_{\acute{v}}^T D D^T = \lambda^{2k} (C \cdot D) C_{\grave{v}} D_{\acute{v}} \tag{6}$$

ponieważ $(c_v^0)_{a,b} = 1 \Leftrightarrow a = \grave{v},\ b = \acute{v}$.
Popatrzmy się teraz na rozkład prawdopodobieństwa wzorców o danym kształcie $m$ w środku takiego nieskończonego ciągu.

$$p_v = \lim_{k\to\infty} \frac{\sum_{a,b\in\mathcal{A}} (c_v^k)_{a,b}}{\sum_{w\in V(\overline{m})} \sum_{a,b\in\mathcal{A}} (c_w^k)_{a,b}} = \frac{C_{\grave{v}} D_{\acute{v}}}{\sum_{w\in V(\overline{m})} C_{\grave{w}} D_{\acute{w}}} \tag{7}$$

W ten sposób dostaliśmy:

1. Wzorce o takim samym kształcie i wartościach brzegowych są równoprawdopodobne. Czyli innymi słowy, po ustaleniu wartości na brzegu pewnego zbioru, rozkład prawdopodobieństwa możliwych wartościowań wewnątrz tego zbioru jest jednorodny. Jak zobaczymy później - ten warunek uogólnia się do wyższych wymiarów

2. Dla $m = 1$ dostajemy rozkład prawdopodobieństwa symboli:

$$p_a = \frac{C_a D_a}{\sum_{b\in\mathcal{A}} C_b D_b} = \frac{C_a D_a}{CD} \tag{8}$$

3. Dla $m = 2$ dostajemy

$$p_{ab} = \frac{C_a M_{ab} D_b}{\sum_{a',b'\in\mathcal{A}} C_{a'} M_{a'b'} D_{b'}} = \frac{C_a D_a}{CMD} = \frac{C_a D_b}{\lambda CD} \tag{9}$$

Na podstawie powyższych punktów nasuwa się algorytm probabilistyczny, który pozwoli wygenerować "typowy" ciąg w tym modelu:

Mając ustalone wartości węzłów na lewo od danego węzła, chcemy go ustalić jego wartość (symbol który tutaj wstawimy). Czyli na podstawie "historii", chcemy podać rozkład prawdopodobieństwa wartościowania kolejnego węzła.

Z 1. istotny jest dla nas tylko węzeł brzegowy tej "historii" - węzeł poprzedni - powiedzmy $a$. Teraz z 2. i 3. powinniśmy aktualnemu węzłowi nadać wartość na $b$ z prawdopodobieństwem:

$$S_{ab} = \frac{p_{ab}}{p_a} = \frac{M_{ab}}{\lambda}\frac{D_b}{D_a} \tag{10}$$

**Czyli nasz algorytm probabilistyczny to proces Markowa.**

Praktycznie - brakuje inicjalizacji - nie możemy wypełniać od $-\infty$. Ale pierwszy węzeł możemy albo wypełnić ustalonym symbolem (tracimy poniżej 1 bitu), albo możemy po prostu użyć obliczonego $(S_{0a})_{a\in\mathcal{A}}$. Dokładna dyskusja inicjalizacji - na końcu pracy

Czyli nieformalnie - praktycznie wszystkie elementy są wygenerowane tym procesem Markowa. Nam wystarczy że ograniczając się tych elementów - dostaniemy maksymalną możliwą średnią ilość informacji na punkt $H$.

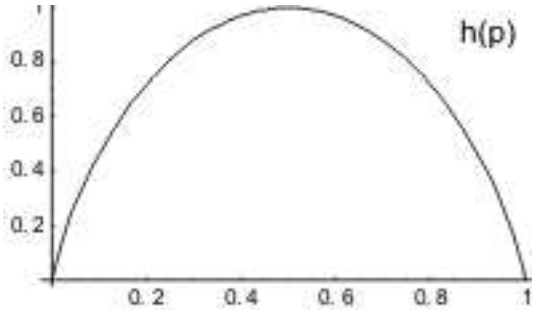
Zastanówmy się najpierw nad problemem: mamy ciąg 0 i 1 tyle że wiemy że prawdopodobieństwo 1 wynosi $p$.
Policzymy ile średnio informacji przypada na jeden symbol: $\tilde{p} := 1 - p$

$$\begin{aligned}\binom{n}{pn} &= \frac{n!}{(pn)!(\tilde{p}n)!} \cong (2\pi)^{-1/2}\frac{n^{n+1/2}e^n}{(pn)^{pn+1/2}(\tilde{p}n)^{\tilde{p}n+1/2}e^n} = \\ &= (2\pi np\tilde{p})^{-1/2}p^{-pn}(\tilde{p}n)^{-\tilde{p}n} = (2\pi np\tilde{p})^{-1/2}2^{-n(p\lg p+\tilde{p}\lg\tilde{p})}\end{aligned}$$

$$H_p = \lim_{n\to\infty}\frac{\lg\binom{n}{pn}}{n} = -p\lg p - \tilde{p}\lg\tilde{p} =: h(p) \tag{11}$$

gdzie skorzystaliśmy z wzoru Stirlinga: $\lim_{n\to\infty}\frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} = 1$

Czyli gdy z ciągów 0 i 1, których ilość zachowuje się jak $2^n$, wybierzemy takie, że prawdopodobieństwo 1 wynosi $p$, to ich ilość będzie się zachowywała jak $2^{nh(p)}$. Czyli jeśli się ograniczamy do ciągów o $p = 1/2$ - dostaniemy maksymalną średnią ilość informacji. Czyli: w systemie dwójkowym dostajemy maksymalną możliwą ilość informacji (1bit/cyfrę) gdy kolejne bity występują z jednorodnym prawdopodobieństwem i nie ma żadnych korelacji.




Rysunek 2: Entropia dla dwuelementowego alfabetu.

Ilość tych ciągów - **typowych** rośnie tak samo jak ilość wszystkich ciągów, natomiast pozostałych (dla ustalonego $p$) rośnie istotnie wolniej - dla $n \to \infty$ ich wkład staje się nieistotny

Teraz gdy mamy większy alfabet $\mathcal{A}$ oraz rozkład prawdopodobieństwa $p$ : $\sum_{a\in\mathcal{A}}\ p_a = 1$, średnia ilość informacji na bit, wynosi (przez indukcję)

$$H_p = -\sum_{a\in\mathcal{A}} p_a \lg p_a \tag{12}$$

czyli tak na prawdę szukamy entropii Shanona.

Czyli gdy w danym momencie chcemy zapisać informację przez wybranie jednego z symboli według zadanego rozkładu prawdopodobieństwa $p$, zapisujemy $H_p$ bitów. Czyli żeby policzyć średnią ilość informacji na węzeł musimy uśrednić $H_p$ po wszystkich sytuacjach.

W przypadku znalezionego wyżej procesu Markowa $S = (S_{ab})_{a,b\in\mathcal{A}}$:

1. $\forall_{a,b\in\mathcal{A}}\ M_{ab} \geqslant S_{ab} \geqslant 0$
2. $\forall_{a\in\mathcal{A}} \sum_{b\in\mathcal{A}} S_{ab} = 1$
3. $pS = p,\ \sum_{a\in\mathcal{A}} p_a = 1$

Sprawdźmy jego średnią entropię:

$$\begin{aligned}
H &= -\sum_a p_a \sum_b S_{ab} \lg S_{ab} = \\
&= -\sum_a \frac{C_a D_b}{CD} \sum_b \frac{M_{ab}}{\lambda} \frac{D_b}{D_a} \lg \frac{1}{\lambda} \frac{D_b}{D_a} = \frac{-1}{\lambda CD} \sum_{ab} C_a M_{ab} D_b \lg \frac{1}{\lambda} \frac{D_b}{D_a} = \\
&= \frac{CMD \lg \lambda}{\lambda CD} + \frac{1}{\lambda CD} \sum_{ab} (C_a \lg D_a M_{ab} D_b - C_a M_{ab} D_b \lg D_b) = \\
&= \frac{CMD \lg \lambda}{\lambda CD} + \frac{1}{\lambda CD} \sum_{ab} (C_a \lg D_a \lambda D_b - C_a \lambda D_b \lg D_b) = \lg \lambda
\end{aligned}$$

Czyli rzeczywiście znaleziony algorytm probabilistyczny - skopiowanie struktury statystycznej elementów - jest optymalny.
Przypominam, że obliczona entropia mówi, że ilość ciągów wygenerowanych tym algorytmem, zachowuje się jak $2^{nH}$. Ponieważ nie możemy osiągnąć więcej ciągów niż rzeczywiście tam jest, więc w przestrzeni procesów Markowa zgodnych z tym modelem ($M_{ab} = 0 \Rightarrow S_{ab} = 0$), znaleziony proces spełnia maksimum na entropię.

Rozważmy następnie pewien **przykład**:

**Definicja 3.** *k-model:*
$X = \mathbb{Z} \quad A = \{0, 1\}$ więz*: po "1" następuje k "0"*

Ponieważ więzy mają zasięg $k+1$, więc należy zgrupować $k$ poprzednich elementów w 1. Mamy nowe stany $0, 1, ..., k$:

- "0" : wcześniej było $k$ "0"(teraz może być "1")
- "$i$" : $k - i + 1$ pozycji temu była "1" (za $i$ pozycji może być 1)

W stanach powyżej 0, musimy postawić "0".
Czyli cały algorytm(proces Markowa) to
"$q \in [0, 1]$-prawdopodobieństwo, że w stanie 0 postawimy 1"

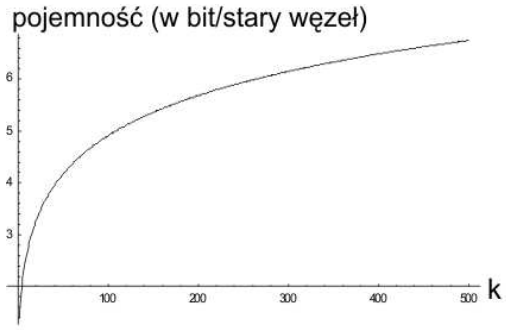
Zróbmy 1 krok($p := p_0$): $\begin{cases} p_k = pq \\ p_i = p_{i+1} \quad \text{dla } i \in \{1, ..., k-1\} \\ p = p_1 + p\tilde{q} \end{cases}$
Czyli $p_1 = p_2 = ... = p_k = pq$
$p = 1 - p_1 - p_2 - ... - p_k = 1 - kpq \qquad \boxed{p = \frac{1}{1+kq}}$
Czyli optymalna średnia entropia dla algorytmu, wynosi: $\boxed{\text{H}=\max_{q\in[0,1]} \frac{h(q)}{1+kq}}$.

We wstępie podaliśmy przykład zastosowania niniejszych rozważań do kompresji danych na płaszczyźnie.
Właśnie policzyliśmy ogólną jednowymiarową wersję tamtego podejścia: na taśmie mamy stałą szerokość $d$ plamki magnetycznej, którą zapisujemy informację.
Możemy:

- przyjąć, że obszar niezamagnesowany ma szerokość będącą wielokrotnością $d$ - czyli zapisujemy $1\text{bit}/d$
- przyjąć, że obszary nienamagnesowane mają szerokość będącą wielokrotnością $\frac{d}{k+1}$ dla pewnego ustalonego $k \in \mathbb{N}$

Czyli w drugim przypadku przy stałej szerokości plamki, musimy $k + 1$ razy dokładniej pozycjonować głowicę.

Zauważmy, że $k$ - model, gdzie $X = \mathbb{Z}_{\frac{d}{k+1}}$, plamka to $10^k$ (1 i k zer) spełnia założenia.
Czyli dostajemy $(k+1)\max_{q\in[0,1]} \frac{h(q)}{1+kq}$ bit/$d$ - stąd tabelka we wstępie.
Nie można łatwo opisać tą funkcji-wyniki numeryczne przedstawiam na rysunku 3.




Rysunek 3: Pojemność informacyjna dla $k$ - modeli/stary węzeł.

# 3 Kodowanie

Pokażemy teraz jak wykorzystać np. znaleziony proces Markowa (ogólnie - algorytm probabilistyczny). Do zakodowania konkretnej informacji.
Uzasadniliśmy że wystarczy to zrobić lokalnie - mamy na wejściu pewną informację i konkretną sytuację - rozkład prawdopodobieństwa możliwych teraz do wyboru symboli.
Przyjmijmy że są 2 możliwe symbole: 0 i 1, przy czym 1 ma być wybrana z prawdopodobieństwem $q$. Gdy $q = 1/2$ powinniśmy dostać zwykły system dwójkowy.

W ogólnym przypadku, możemy np. podzielić symbole na 2 podzbiory - pierwsza sytuacja to wybór podzbioru i tak dalej.

Mamy najpierw pewną informację $x \in \mathbb{N}$ oraz $q$ jak wyżej. Chcemy teraz mieć:

$$x \quad \underset{\text{odkodowanie}}{\overset{\text{kodowanie}}{\longrightarrow \atop \longleftarrow}} \quad (x', d) \tag{13}$$

Czyli chcemy (w kodowaniu) z $x$ (dużej, losowej liczby) wyciągnąć informację $H(q)$ w postaci $d \in \{0, 1\}$ i pozostałą informację umieścić w $x'$.

Popatrzmy się na $\#\{k \in \mathbb{N} : k < x\} = k$ - podzielmy ten zbiór na 2 podzbiory: takie które dostaną $d = 1$ i $d = 0$.

Przy czym tych pierwszych powinno być około $qx$.
Przyjmijmy na przykład $\boxed{x_1 := \lceil xq \rceil}$ - ilość elementów w podzbiorze z $d = 1$
a więc analogicznie $\boxed{x_0 := x - \lceil xq \rceil}$
A więc $x$ jest w pierwszym podzbiorze wtw gdy po nim jest skok $\lceil xq \rceil$, czyli

$$\boxed{d := \lceil (x+1)q \rceil - \lceil xq \rceil} \tag{14}$$

Wybraliśmy $d$ - w ten sposób przeszliśmy do jednego z tych dwóch podzbiorów
Czyli przyjmujemy (**"stare" kodowanie**) $\boxed{x' = x_d}$

na przykład dla $q = 0.3$

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_0$ | | 0 | 1 | | 2 | 3 | | 4 | 5 | 6 | | 7 | 8 | | 9 | 10 | | 11 | 12 |
| $x_1$ | 0 | | | 1 | | | 2 | | | | 3 | | | 4 | | | 5 | | |

Odkodowywanie: idziemy od końca - mamy $d$ i $x'$, szukamy $x$.
Oznaczmy $r := \lceil xq \rceil - xq \in [0, 1)$
$d := \lceil (x+1)q \rceil - \lceil xq \rceil = \lceil (x+1)q - \lceil xq \rceil \rceil = \lceil (x+1)q - r - xq \rceil = \lceil q - r \rceil$
Czyli $d = 1 \Leftrightarrow r < q$

Gdy $\boxed{d = 1}$ - $x' = \lceil xq \rceil = xq + r$
Czyli ponieważ $0 \leqslant r < q \quad x = \frac{x'-r}{q} = \left\lfloor \frac{x'}{q} \right\rfloor$ - bo musi $\in \mathbb{N}$

Gdy $\boxed{d = 0}$ - $q \leqslant r < 1$ czyli $\tilde{q} \geqslant 1 - r > 0$
$x' = x - \lceil xq \rceil = x - xq - r = x\tilde{q} - r$

$$x = \frac{x' + r}{\tilde{q}} = \frac{x' + 1}{\tilde{q}} - \frac{1 - r}{\tilde{q}} = \left\lceil \frac{x' + 1}{\tilde{q}} \right\rceil - 1$$

Ostatecznie: **"stare" odkodowanie**

| $d = 0$ | $d = 1$ |
|---|---|
| $x = \boxed{\frac{x'+1}{\tilde{q}}} - 1$ | $x = \boxed{\frac{x'}{q}}$ |

Dla dużych $x$ ilość informacji jaką zawiera, to w przybliżeniu $\lg(x)$.
Czyli po kodowaniu np. prawdopodobieństwem $q$ pozbawiamy go $\lg(q)$ bitów ($\lg(x/q) = \lg(x) - \lg(q)$).
Czyli średnio tracimy $-q \lg(q) - \tilde{q} \lg(\tilde{q})$ - dokładnie tyle ile umieściliśmy w $d$.

Z definicji - prawdopodobieństwo się zgadza. Natomiast gdy $x$ - rośnie, kolejne cyfry rosną niezależnie - są nieskorelowane.

Pokażemy teraz jak kodować nieskończone ciągi bitów.
Mamy na razie algorytm, który potrafi pobrać z dużej liczby $x$ pewną informację. Nie możemy jednak wykonywać operacji matematycznych na nieskończenie długich liczbach.
Wybierzmy więc pewien przedział na którym będziemy pracować:
Przyjmijmy np. że $x \in I := [2^n, 2^{n+1} - 1]$

W tym przypadku **kodowanie** wygląda następująco:

1. $(t, d) =$ "stare kodowanie" $(x)$ - czyli $t < x$. Wykorzystujemy $d$ (np. do wyboru kolejnego symbolu w procesie Markowa)
2. $k = n - \lfloor \lg t \rfloor$ - ile bitów trzeba pobrać z wejścia
3. $x = t \cdot 2^k + 2^{k-1}b_0 + 2^{k-2}b_1 + ... + b_k$(bity z wejścia, chronologicznie)

Natomiast **odkodowanie**:

1. Dopóki $(t =$ "stare odkodowanie"$(x, d)) \geqslant 2^{n+1}$:
   zrzuć na wyjście najmłodszy bit $x$, $x = \lfloor x/2 \rfloor$
2. $x = t$

Pozostaje tylko zainicjować:
**Kodowanie** zaczynamy od pobraniu z wejścia pierwszych $n$ bitów, a kończymy pomijając pobieranie z wejścia - znajdujemy kolejne $d$, nie przejmując się że zmniejszamy $x$, aż dojdziemy do $x = 0$. Dla małych $x$ "stare kodowanie" gorzej zachowuje własności statystyczne, ale dalej daje bijekcję między ciągami bitów a symboli.

Natomiast przy **odkodowaniu** robimy dokładnie na odwrót - najpierw używamy "starego odkodowania" od $x = 0$ do $x \in I$. Na końcu (jak skończą się symbole) - zrzucamy co zostało ($n$ bitów) na wyjście.

Pokazany algorytm ma praktycznie (zamortyzowany) stały koszt kroku - jest tani zarówno czasowo jak i pamięciowo. Obliczona średnia ilość informacji na symbol jest wartością oczekiwaną dla losowych danych. Jednak skonstruowanie "złośliwego" przypadku wymagałoby dokładnej znajomości algorytmu probabilistycznego (teoretycznie: liczby rzeczywiste) i jego przebiegu - jest to raczej rzadki przypadek.
Natomiast **wadą** tego algorytmu jest to, że gdy w jednym miejscu się pojawi błąd - dalej dostaniemy praktycznie losową informację (praktycznie nieskorelowany ciąg bitów z $q = 1/2$) - więc pojedynczy błąd wydaje się że można wykryć - dane od pewnego punktu przestaną "mieć sens". Jednak żeby stwierdzić czy dane "mają sens" muszą mieć dodatkową redundancję. Czyli wydaje się że ta wada jest tylko złudna - możemy zużyć część informacji na zabezpieczenia przed błędami.

# 4 Pojemność informacyjna ogólnych modeli

W niniejszym rozdziale podam ogólne rozważania nad ogólnymi modelami. Udowadniam też istnienie entropii dla sporej klasy modeli.

**Definicja 4.** *Modelem* będziemy nazywać trójkę $(X, \mathcal{A}, M)$
*przestrzeń* $X \subset \mathbb{Z}$
*alfabet* $\mathcal{A}$ -skończony zbiór symboli
*więzy* $N$(przez stany zabronione)*:*
$N \subset \{((p_1, p_2, ..., p_k), (s_1, s_2, ...s_k)) : k \in \mathbb{N} \quad \forall_i \ p_i \in X \ s_i \in \mathcal{A}\}$

wtedy *elementy* to $V(X)$*, gdzie dla* $A \subset X$

$$V(A) := \{v : A \to \mathcal{A} | \forall_{(p,s)\in N} \ \neg\forall_i \ v_{p_i} = s_i\} \qquad V := \bigcup_{A\in X:\ \#A<\infty} V(A)$$

Dygresja: Jest to bardzo ogólna definicja. Oczywiście zamiast definiować przez stany zabronione, można przez stany dozwolone - żeby zamienić, wystarczy np. wziąć odpowiedni duży zbiór i wskazać wszystkie dozwolone wartościowania. W tym podejściu łatwo zobaczyć na przykład, że możemy dostać problemy kafelkowania - każdy kafelek oznaczę innym symbolem i w stanach dozwolonym zapiszę jego kształt tym symbolem. W kafelkowaniu mogą się zdarzać różne rzeczy, np. może wymuszać nieperiodyczność. Żeby troszkę lepiej kontrolować ten model, później ograniczę trochę tą klasę.

Jako model przykładowy przyjmę

**Definicja 5.** *Hard-square model (HS):*

$$(X = \mathbb{Z}^2, \mathcal{A} = \{0, 1\}, N = \{(((i, j), (k, l)), (1, 1)) : i, j, k, l \in \mathbb{Z}, |i - k| + |j - l| = 1\}) \tag{15}$$

Czyli w każdym punkcie $\mathbb{Z}^2$ wstawiamy 0 lub 1 tak, żeby nie było dwóch sąsiednich jedynek. Spotkałem się też z inną nazwą tego modelu w dynamice symbolicznej (tzw. shifty wielowymiarowe): "grid graph" [1].

Jest to jeden z najprostszych istototnie trudniejszych problemów od jednowymiarowych. Jednak okazuje się, że metody analizy tego problemu muszą być na tyle ogólne, że można je łatwo przenieść do innych.

O tym modelu można myśleć jako o granicy w $T \to 0$ modelu podobnego do modelu Isinga: $\mathcal{H} = \pm \sum_{(i,j)-\text{sąsiedzi}} s_i s_j$, tyle że tam alfabet to było $\{-1, 1\}$. Ta

niby drobna różnica powoduje, że w modelu Isinga minimum energii osiągane jest dokładnie dla dwóch różnych sieci - zależnie od znaku - wszystkie spiny w jedną stronę lub na przemian - czyli średnia entropia resztkowa (w $T \to 0$) wynosi 0.

Metody pozwalające na dokładne rozwiązanie 2 - wymiarowego modelu Isinga nie dają się bezpośrednio tu przenieść. W [5] można znaleźć metodę dla problemów z niezerową entropią resztkową. Udaje się nią "rozwiązać" np. model hard - hexagon (zabronione są tym razem 2 jedynki u 6 sąsiadów - dochodzi lewy-górny i prawy-dolny), jednak np. dla hard - square powstają pewne osobliwości, których nie udaje się pozbyć. Napisałem "rozwiązać" bo mówię tu tylko o znalezieniu analitycznej formuły na średnią entropię. Dla nas to i tak byłoby za mało - będziemy potrzebować korelacji, których tamta metoda nie oferowała. Jednak używając wymienionej metody udało się w [6] otrzymać rozwinięcie entropii dla HS z dokładnością do 43 miejsc po przecinku - będę tego wyniku używał do oceny znalezionych algorytmów.

Określmy:

**Definicja 6.**
*Sąsiedztwem punktu* $x \in X$ nazywamy

$$N_x := \{y \in X : \exists_{((x^1,\ldots,x^k),(a_1,\ldots,a_k))\in N}\ \exists_{i,j}\ x^i = x,\ x^j = y\}$$

*Zasięgiem więzów,* nazywamy $L := \max\{|y_i - x_i| : x \in X,\ y \in N_x,\ i \in \mathbb{N}\}$
*Wnętrzem* $A \subset X$ nazywamy

$$A^- := \{x \in A : N_x \subset A\}$$

*Brzegiem* $A \subset X$ nazywamy $A^o := A \backslash A^-$
*Pogrubieniem* $A \subset X$ nazywamy

$$A^+ := \bigcup_{x\in A} N_x$$

Zbiór $A \subset X$ nazywamy *spójnym,* gdy

$$\forall_{x,y\in A}\exists_{n,(x_i)} \quad x^0 = x,\ x^n = y,\ \forall_i \sum_j |x_j^i - x_j^{i+1}| = 1.$$

Oczywiście $A^{-+} \subset A \subset A^{+-}$

Dla translatywnie niezmienniczych modeli sąsiedztwo wystarczy znać dla jednego punktu, np. dla HS: $N_x = x + \{(1,0),(0,1),(-1,0),(0,-1)\}$.

Sąsiedztwo jest zawsze zbiorem symetrycznym($x \in N_y \Leftrightarrow y \in N_x$), $x \in N_x$, czyli

$$\rho(x,y) = \min_n \exists_{(x^0=x,x^1,\ldots,x^n=y)}\ \forall_i x^{i+1} \in N_{x_i}$$

jest naturalną metryką dla modelu.

Model HS wykazuje wysoką symetrię:

**Definicja 7.** *Symetrią modelu* będziemy nazywać bijekcję $S : X \hookrightarrow X$ taką, że

$$((x^1, ..., x^k), (a_1, ..., a_k)) \in N \Leftrightarrow ((S(x^1), ..., S(x^k)), (a_1, ..., a_k)) \in N \qquad (16)$$

Czyli w HS mamy symetrie generowane przez obrót o $\pi/2$, symetrię osiową i translację.

**W dalszym ciągu przyjmiemy, że model ma skończony zasięg więzów, $X = \mathbb{Z}^m,\ m \in \mathbb{N}$(periodyczność przestrzeni), model jest niezmienniczy ze względu na translacje ($t_x(y) := y + x$) i że jest prosty(#):**

**Definicja 8.** Model nazywamy *prostym(#),* gdy $\#V(X) > 1$ i
istnieje $N > n \geqslant 0$ :

$$\forall_{\text{spójne } A \subset X}\ \forall_{v \in V((A^N)^o)}\ \forall_{u \in V(A^n)}\ \exists_{w \in V((A^N)^- \setminus A^n)} \quad u \cup w \cup v \in V(A^N)$$

gdzie $A^0 := A,\ A^{i+1} := (A^i)^+$.
Czyli: dla odpowiednio "ładnych" zbiorów: postaci $A^n$, istnieje pogrubienie ($N - n$ razy), takie że niezależnie jak ustalimy wartości (symbole) na brzeg tego pogrubienia, możemy dalej dowolnie wartościować $A^n$.

Dla modeli z *symbolem neutralnym* - takim, który nie występuje w żadnym więzie (zawsze można go wstawić, np. 0 w HS), $n = 0,\ N = 2\ : A^+ \setminus A$ wypełniamy tym symbolem.

**Lemat 9.**

$$\exists_{n' \geqslant \max(L,1)} \forall_{v \in V(\beta^+_{n'} \setminus \beta_{n'})} N(\beta^+_{n'}, v) > 1$$

gdzie *blok o boku* $k$, to $\beta_k = \{0, 1, ..., k-1\}^m$,

$$N^A_u \equiv N(A, u) := \#\{w \in V(A) : w|_{D(u)} = u\}, \quad N(A) \equiv N(A, \{\}).$$

Czyli bloków od pewnego rozmiaru (większego od zasięgu więzów), nie można "zakleszczyć", czyli tak ustalić wartości jego sąsiadów, żeby się go nie dało "zawartościować".

*Dowód:*
Weźmy $k$: $\#V(\beta^k) > 1$ ($\#V(X) > 1$).
Teraz $\beta^{N-1}_k$ zawiera się w pewnym bloku o boku $n' > L$.
To $n'$ jest dobre z (#).

**Twierdzenie 10.** *Dla modeli spełniające powyższe warunki, istnieje i jest dodatnia średnia entropia:*
*istnieje ciąg zbiorów* $(A_i)$*:* $A \subset X,\ \#A < \infty,\ \bigcup_i A_i = X$*, że istnieje granica:*

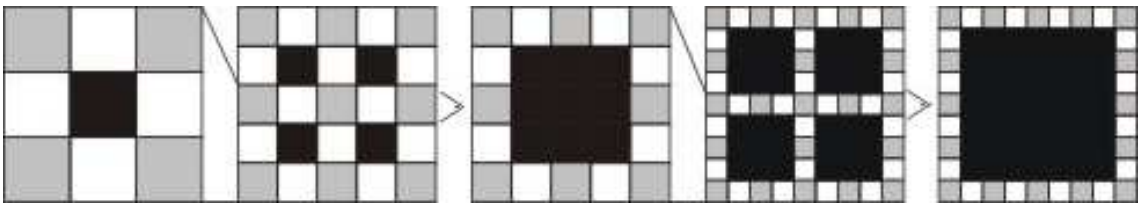
$$H = 0 < \lim_{i\to\infty} \frac{lg(N(A_i))}{\#A_i} \leqslant \lg \#\mathcal{A}$$

*Dowód:*
Weźmy $n'$ jak w lemacie. Będziemy operować na *klockach*: $B \equiv \beta_{n'}$ umieszczanych w węzłach sieci $n'\mathbb{Z}^m$. Wartościowanie klocka jest ograniczane tylko przez wartościowanie *klocków przylegających*: $B + n'(\{-1,0,1\}^m \backslash \{0\})$.

Jako ciąg zbiorów przyjmujemy $A_i := B + n'\{0,...,2^i\}^m,\ H_i := \frac{lg(N(A_i))}{\#A_i}$
Zauważmy, że $H_i \leqslant \#\mathcal{A}$ - taką dostalibyśmy średnią entropię bez więzów.

Żeby udowodnić, że $H_i$ ma granicę, wskażę ciąg rosnący $H'_i$, taki że $\forall_i H'_i \leqslant H_i$ i pokażę, że w granicy $i \to \infty,\ H'_i \to H_i$. Ponieważ ciąg rosnący i ograniczony ma granicę, więc i $H_i$ będzie miał granicę, co chcieliśmy udowodnić.

Przeprowadzę teraz pewną konstrukcję w celu zapewnienia monotoniczności:



Rysunek 4: Podział dla $A_1,\ A_2,\ A_2,\ A_3,\ A_3$. Czarne - esencja. Pozostałe (wypełniacz) wartościujemy tak, że bloki o tych samych kolorach wartościujemy tak samo.

Zdefiniujmy "wypełniacz":

$$C_i := B + n'\{x \in \{0,...,2^i\}^m : \exists_i x_i \in \{0, 2^i\}\}$$

czyli zewnętrzne klocki $A_i$.
$H'_i$ = lg( na ile sposobów możemy "zawartościować esencję": $A_i \backslash C_i$ po wcześniejszym wypełnieniu wypełniacza pewnym konkretnym wzorcem)$/\#A_i$

Oczywiście zmniejszyliśmy licznik przez ograniczenia, mamy więc jak potrzebujemy: $H'_i \leqslant H_i$.
Żeby pokazać $H'_{i+1} \geqslant H'_i$ wprowadzę krok pośredni dla wypełniacza

$$C'_{i+1} := B + n'\{x \in \{0,...,2^i\}^m : \exists_i x_i \in \{0, 2^i, 2^{i+1}\}\}$$

Teraz esencję: $A_{i+1}\backslash C'_{i+1}$ stanowią $2^m$ esencje z poprzedniego kroku.
Załóżmy teraz, że tak wypełniliśmy wypełniacz $C'_{i+1}$, że każdy blok esencji ma tak samo "zawartościowane" przylegające klocki (dostajemy $H''_i$), czyli każdą wartościujemy niezależnie na tyle samo sposobów: licznik we wzorze na entropię wzrośnie $2^m$ razy w porównaniu do $H'_i$. Natomiast mianownik wzrośnie $\left(\frac{2^{i+1}+1}{2^i+1}\right)^m < 2^m$, czyli $H'_i < H''_{i+1}$ (stąd dostajemy też $H > 0$).
Nierówność $H''_{i+1} \leqslant H'_{i+1}$ jest oczywista - zmniejszamy ograniczenia dla licznika.

Pozostaje pokazać, że można tak nadać wartości węzłom wypełniacza, że każdy z bloków esencji z $A_{i+1}\backslash C'_{i+1}$ ma tak samo "zawartościowane" przylegające klocki jak $A_i\backslash C_i$.

Weźmy siatkę $Y = 2n'\mathbb{Z}^m$ oraz ponumerujmy dowolnie $\{0,1\}^m = \{x^i\}_{i=1,\dots,2^m}$.
Teraz każdy z bloków z $Y + B + x^1$ możemy "zawartościować" niezależnie od pozostałych ($n' \leqslant$ zasięg więzów), powiedzmy na $w_1 \in V(B)$.
Teraz, korzystając z lematu 9, po ustaleniu wartości na tamtych klockach, analogicznie znajdujemy $w_2 \in V(B)$ dla $Y + B + x^2$. I tak dalej, dostając uniwersalny periodyczny wypełniacz, np.:

$$k_i = V\left(A_i, \bigcup_{j=1}^{2^m} W_j|_{A_i}\right)$$

gdzie $D(W_j) = Y + B + x^j$, dla dowolnego $y \in Y$, $x \in B : W_j(x + y + x^j) := w_j(x)$ który pasuje do założeń (2 dzieli $2^i$).

Pozostaje pokazać, że $\lim_{i\to\infty} H_i - H'_i = 0$
Popatrzmy się na zbiór $D_i := (A_i\backslash C_i)\overbrace{--\dots-}^{N-1}\overbrace{++\dots+}^{n}$.
$(A^{-+} \subset A)$, czyli z (#): niezależnie od wartości na $C_i$, możemy dowolnie wartościować $D_i$.

Ponieważ $\beta_k^- \supset \beta_{k-2L} + (l, l, \dots, l)$, więc $\beta_{2^i-2NL} + (NL, \dots, NL) \subset D_i$,
więc dla dużych $i$:

$$\#D_i \sim (2^i)^m \qquad \#(A_i\backslash D_i) \sim (2^i)^{m-1}$$

Czyli $\lim_{i\to\infty} \frac{\#(A_i\backslash D_i)}{\#D_i} = 0$.

Mamy $N(D_i) \leqslant k_i \leqslant N(A_i)$
Ale $\frac{N(A_i)}{N(D_i)} \leqslant (\#\mathcal{A})^{\#(A_i\backslash D_i)}$ - równość byłaby bez więzów.
Czyli $\frac{\lg N(A_i) - \lg N(D_i)}{\#A_i} \leqslant \frac{\#(A_i\backslash D_i)\lg(\#\mathcal{A})}{\#A_i} \to 0$.

Czyli z trzech ciągów: $\lim_{i\to\infty} H_i - H'_i = 0$ cnd.

**Czyli dla "sensownych" modeli możemy mówić o średniej ilości informacji na węzeł sieci - *entropii modelu.***

W dalszej części pracy do założeń o modelu, dołożę *nieredukowalność*:

**Definicja 11.** Translatywnie niezmienniczy model nazywamy *nieredukowalnym,* gdy:

$$\bigcup_i A_i = x$$

Gdzie $A_0 := \{0\}, \quad A_{i+1} := (A_i)^+$.

Załóżmy, że układ nie jest nieredukowalny (jest redukowalny):

$$Y := \bigcup_i A_i \neq X$$

Ponieważ $Y = -Y,\ Y = Y + Y$, więc $y \in Y \Rightarrow y\mathbb{Z} \subset Y$ - $Y$ jest siecią periodyczną:

$$Y = \sum_{i=1,\dots,m'} \mathbb{Z}y^i \qquad (\forall_{i\neq j} \mathbb{Z}y^i \cap \mathbb{Z}y^j = \{0\})$$

Teraz wystarczy dokonać transformacji $y^i \to (\delta_{ij})_{j=1,\dots,m'}$ i dostajemy układ nieredukowalny.

$x \sim y \Leftrightarrow x - y \in Y$ jest relacją równoważności, czyli $X = \bigcup_i x^i + Y$ (suma rozłączna dla pewnego ciągu $(x_i)$) możemy traktować jako identyczne, niezależne sieci. Na przykład średnia entropia będzie taka sama jak dla układu zredukowanego.

# 5 Podejście statystyczne

Będziemy teraz chcieli, podobnie jak w przypadku jednowymiarowym znaleźć *opis statystyczny*, który otrzymamy uśredniając po wszystkich możliwych eleamentach(wartościowaniach przestrzeni). Takie elementy powinny istotnie (wykładniczo od wielkości układu) dominować pozostałe, czyli w wyniku tego uśrednienia powinniśmy dostać rzeczywiście opis elementów, które zdominują pozostałe - *elementy typowe* - biorąc losowy element, z prawdopodobieństwem 1 jest on typowy.

Korzystając z niezmienniczości ze względu na symetrię translacyjną, narzuca się:

**Definicja 12.** *Opis statystyczny:* funkcja

$$p : \{(A, f) : A \subset X,\ \#A < \infty,\ f : A \to \mathcal{A}\} \to [0, 1]$$

*taka, że*

$$\forall_{A\subset X,\ \#A<\infty}\forall_{x\in X\setminus A}\forall_{f:A\to\mathcal{A}}\ \sum_{a\in\mathcal{A}} p_{f\cup\{(x,a)\}} = p_f \tag{17}$$

$$p_{\{\}} = 1$$

gdzie $p_f \equiv p(D(f), f)$, $\quad D(f)$ - dziedzina $f$.

Czyli chcemy mieć odwzorowanie, które dowolnemu kształtowi $A$ przyporządkowuje rozkład prawdopodobieństwa wartościowań $f$ na tym kształcie.
Ze względu na symetrię translacyjną, nie musimy podawać bezwzględnej pozycji kształtu - przyjmę zapis np. $p(01)$ - prawdopodobieństwo, że po wybraniu dowolnego węzła i jego sąsiada po prawej dostaniemy na nich odpowiednio 0 i 1.
Warunki zapewniają unormowanie prawdopodobieństwa do 1
(np. $p(10) + p(00) = p(0)$).

Teraz będziemy chcieli wyśredniować po wszystkich elementach w celu otrzymania opisu typowego. Problem w tym że elementów jest nieskończenie (nawet nieprzeliczalnie) wiele. Musimy więc wybrać jakoś "sensownie" przeliczalny ciąg po którym będziemy średniować. Ponieważ łatwo się średniuje gdy przestrzeń jest skończona, możemy przybliżać przestrzeń ciągiem skończonych podzbiorów:

**Definicja 13.**
*Ciągiem normalnym* nazywamy ciąg zbiorów skończonych $(A_i)_{i\in\mathbb{N}}$ taki, że:

$$A_0 \subset A_1 \subset A_2 \subset ...$$

$$\bigcup_{i\in\mathbb{N}} A_i = X$$

$(A, v)$ - *aproksymacją opisu typowego* (dla $D_f \subset A \subset X,\ v \in V(A^o)$), nazywamy:

$$p_f^{A,v} := \frac{N(A, v\cup f)}{N(A, v)}$$

Określmy jeszcze: $p_f^A := p_f^{A,\emptyset}$.

Teraz dla pewnego $B \supset A,\ v \in V(B^o),\ D_f \subset A^o,\ f \in V(A)$:

$$\begin{aligned} p_f^{B,v} &= \frac{N(B, v\cup f)}{N(B, v)} = \frac{\sum_{u\in V(A^o)} N(B\setminus A^-, u\cup v)N(A, u\cup f)}{N(B, v)} = \\ &= \sum_{u\in V(A^o)} p_f^{A,u}\frac{N(B\setminus A^-, u\cup v)N(A, u)}{N(B, v)} = \sum_{u\in V(A^o)} p_f^{A,u}p_u^{B,v} \end{aligned} \tag{18}$$

Podzieliliśmy sumę po wszystkich wartościowaniach na sumę wewnątrz i na zewnątrz zbioru rozdzielającego $A^o$. Przypomina to wspomnianą w przypadku

jednowymiarowym, niezależność między zbiorami rozdzielonymi. Rozwinę niedługo ten wątek.

Będziemy szukać $p^o_f$ - opisu typowego przez kolejne aproksymacje - $p^{A,v}_f$ dla kolejnych zbiorów z ciągu normalnego. Pokazaliśmy właśnie, że przechodząc do kolejnego, większego zbioru, **dostaniemy pewną średnią ważoną z poprzednich aproksymacji** dla tego $f$.
Czyli uzasadniliśmy, że w większym zbiorze dostajemy na pewno nie gorszą aproksymację:

$$A \subset B \Rightarrow \hat{p}^A_f \geqslant \hat{p}^B_f \geqslant \check{p}^B_f \geqslant \check{p}^A_f$$

gdzie

$$\check{p}^A_f := \min_{v \in V(A^o)} p^{A,v}_f \qquad \hat{p}^A_f := \max_{v \in V(A^o)} p^{A,v}_f \qquad d^A_f := \hat{p}^A_f - \check{p}^A_f$$

Uzasadniliśmy, że $d^A_f$ maleje w ciągu normalnym. Jeśli udowodnilibyśmy, że dla pewnego ciągu normalnego $(A_i)$, $d^{A_i}_f$ maleje do 0, to biorąc dowolny inny ciąg normalny $(B_i)$, ponieważ $\forall_i \exists_j\ A_i \subset B_j$, więc $\lim_{i\to\infty} p^{A_i}_f = \lim_{i\to\infty} p^{B_i}_f$.
Czyli ta granica jest jedynym sensownym opisem typowym.

Jednak nie jestem w stanie tego formalnie udowodnić, dlatego przyjmę, wydawałoby się - intuicyjne:

**Założenie 14** (*)**.** $d^{A_i}_f \to 0$ *dla pewnego ciągu normalnego* $(A_i)$ *i dowolnego wartościowania* $f$*.*

Pozostała część pracy będzie się opierać na tym założeniu(*). Możemy teraz zdefiniować:

**Definicja 15.** *Opis typowy,* to $p^o_f := \lim_{i\to\infty} p^{A_i}_f$
gdzie $(A_i)$ - dowolny ciąg normalny.

Pokażemy, że opis typowy zachowuje symetrie.
$S$ - pewna symetria modelu
$S' : V(X) \to V(S(X)) = V(X) : S'(v)(x) = v \circ S^{-1}$ jest bijekcją na $V(X)$
Teraz dla ciągu normalnego $(A_i)$ i pewnego wzorca $f$, weźmy (z dowolności) ciąg normalny $(B_i) : B_i = S(A_i)$ i wzorzec $S'(f)$:

$$\begin{aligned} p^o_f &= \lim_{i\to\infty} \frac{\#\{w \in V(A_i) : w|_{D(f)} = f\}}{\#\{w \in V(A_i)\}} = \\ &= \lim_{i\to\infty} \frac{\#\{S'(w) \in V(B_i) : S'(w)|_{D(S'(f))=S(D(f))} = S'(f)\}}{\#\{S'(w) \in V(B_i)\}} = p^o_{S'(f)} \end{aligned}$$

**Spostrzeżenie 16.** *Dla dowolnej symetrii modelu* $S$ *i wzorca* $f$*, zachodzi* $p^o_f = p^o_{S'(f)}$

Przejdźmy teraz do *lokalnego warunku na typowość (LTC-local typicality condition).*
W przypadku, gdy mamy skończonej przestrzeni, mamy skończoną ($N$) ilość elementów, najwięcej informacji ($\lg(N)$) dostaniemy, gdy rozkład prawdopodobieństwa tych elementów jest jednorodny - nie wyróżniamy żadnych.
Mamy analog tego warunku dla nieskończonej przestrzeni: gdy ustalimy wartości elementów na brzegu pewnego skończonego zbioru, to elementy w jego wnętrzu wartościujemy niezależnie od tych na zewnątrz. Wartościowań wewnątrz jest skończenie wiele - LTC mówi, że wszystkie są równoprawdopodobne.

**Spostrzeżenie 17.** *Typowość opisu statystycznego p jest równoważna z LTC: dla dowolnego* $B \subset X,\ \#B < \infty,\ A \subset B^-,\ f \in V(B)$*:*

$$p(f) = \frac{p(f|_{B\setminus A})}{N(B, f|_{B\setminus A})}$$

*Dowód:*

1. Weźmy $p = p^o,\ (A_i)$ - ciąg normalny, $A,\ B,\ f$ jak wyżej

   $$N(A_i, f|_{B\setminus A}) = \#\{u \in V(A) : u \cup f|_{B\setminus A} \in V(B)\} N(A_i, f) = N(B, f|_{B\setminus A}) N(A_i, f)$$

   $$p_f^o = \lim_{i\to\infty} \frac{N(A_i, f)}{N(A_i)} = \lim_{i\to\infty} \frac{N(A_i, f|_{B\setminus A})}{N(B, f|_{B\setminus A}) N(A_i)} = \frac{p(f|_{B\setminus A})}{N(B, f|_{B\setminus A})}$$

2. załóżmy teraz LTC dla $p : B \subset X :\ \#B < \infty,\ D(f) \subset A = B^-$

   $$p(f) = \sum_{v \in V(B^o)} p(f \cup v) = \sum_{v \in V(B^o)} N(B, v \cup f) \frac{p(v)}{N(B, v)} = \sum_{v \in V(B^o)} p(v) p_f^{B,v} \quad (19)$$

   gdzie pierwsza równość była z unormowania (17), druga z LTC (możemy wypełnić $A\setminus D(f)$ na $N(B, v \cup f)$ sposobów)
   Czyli z LTC wynika, że opis jest średnią ważoną aproksymacji opisu typowego, czyli z założenia(*), dla $B$ - kolejne elementy ciągu normalnego dostajemy tezę.

**Uwagi**

1. dla wzorca $f,\ A \subset X : D(f) \subset A^-$ zachodzi:

   $$p_f^o = \sum_{v \in V(A^o)} p_v^o p_f^{A,v}$$

   patrz (19)
   Teraz gdy bierzemy na przykład ciąg $A_0 := 0,\ A_{i+1} := A_i^+$, wiemy z założenia,

że $p_f^{A_i,v} \to p_f^o$, czyli $p_f^o$, z założenia(*), coraz mniej zależy od elementów z brzegu dużych zbiorów:

Czyli **założenie(*) jest równoważne założeniu o znikaniu korelacji dalekozasięgowych.**

2. LTC jest równoważne *pLTC - point local typicality condition*:

$$\forall_{B:N_0^o\subset B\subset X\setminus\{0\},\ \#B<\infty}\forall_{v\in V(B)}\forall_{a,b\in\mathcal{A}}\ p_{v\cup\{(0,a)\}} = p_{v\cup\{(0,b)\}}$$

gdzie $N_0^o = N_0\setminus\{0\}$

Daje to prostszy zestaw warunków na LTC, czyli typowość opisu statystycznego - że wystarczy sprawdzić dla $\#A = 1$. Wykorzystamy to później do generowania przybliżeń elementów typowych.
Mówi na przykład dla HS, że dla dowolnego wartościowania pewnego skończonego zbioru, musi dalej zachodzić

$$p\begin{pmatrix} x & 0 & x \\ 0 & 0 & 0 \\ x & 0 & x \end{pmatrix} = p\begin{pmatrix} x & 0 & x \\ 0 & 1 & 0 \\ x & 0 & x \end{pmatrix}$$

Gdzie "$x$" oznaczają to wartościowanie gdzieś poza $N_0$.
czyli

$$\forall_{A\subset X\setminus N_0:\ \#A<\infty}\forall_{v\in V(A)}\ p_{f_0^*\cup v} = p_{f_1^*\cup v}$$

gdzie $f^* := 0|_{N_0\setminus\{0\}}$ , $f_a^* := f^*\cup\{(0,a)\}$

Inne wartościowania $N_0^o$ wymuszają wartość 0 w środku.

*Dowód:* przez indukcję ze względu na $\#A$: dla 1 - mamy
załóżmy, że mamy dla $\#A = k-1$
Weźmy pewne $A, B, v : \#A = k,\ B\supset A^+,\ v\in V(B\setminus A)$
Chcemy pokazać, że dla dowolnych $f, g\in V(A)$

$$p_{v\cup f} = p_{v\cup g}$$

Gdy istnieje $x\in A$ takie, że $f(x) = g(x)$, to przerzucamy $x$ do $B$ i korzystamy z założenia indukcyjnego.
Jeśli nie to robimy krok pośredni do $f' = f\setminus\{(x,f(x)\}\cup\{(x,g(x)\}$ dla pewnego $x\in A$.

3. weźmy ciąg: $A_0 := 0,\ A_{i+1} := A_i^+$
   wtedy:
   $$\lim_{i\to\infty} \frac{\lg(N(A_i))}{-\sum_{v\in V(A_i)} p_v^o \lg p_v^o} = 1$$
   *dowód:* ustalmy $i \in \mathbb{Z}$
   Po ustaleniu wartościowania na $A_{i+1}^o$ (dowolnie), mamy jednorodny rozkład na $A_i$, a z (#) ($n$ i $N$), $A_{i-N+n}$ możemy dowolnie wartościować, czyli na pewno na $A_i$ mamy więcej wartościowań niż $N(A_{i-N+n})$, czyli $-\sum_{v\in V(A_i)} p_v^o \lg p_v^o \geqslant \lg N(A_{i-N+n})$, teraz powtarzamy rozumowanie z końca dowodu twierdzenia 10 ($\lim_{i\to\infty} \frac{\#A_{i-N+n}}{\#A_i} = 1$) dostajemy tezę.

Uzasadniliśmy, że **opis typowy ma taką samą średnią entropię na punkt, co model.**

W opisie statystycznym mamy pewną nadmiarowość informacji, przez warunki normalizacji.
Możemy się pozbyć tej nadmiarowości na przykład na dwa sposoby:

1. możemy się ograniczyć do takich wartościowań, w których nie ma konkretnego symbolu ($a$). Np. dla HS $s : A \to p_{0|_A}$
   *Dowód:* Chcemy się dowiedzieć o prawdopodobieństwo $f$ o $k+1$ wystąpieniach $a$, na przykład $f(x) = a$, teraz:
   $$p_f = p_{f\setminus\{(x,a)\}} - \sum_{b\in\mathcal{A}\setminus\{a\}} p_{f\setminus\{(x,a)\}\cup\{(x,b)\}}\ .$$

2. *Opis ciągowy:*
   Ponumerujmy sobie dowolnie punkty przestrzeni: $\{x^i\}_{i=0,1,\ldots\infty} = X$
   i wybierzmy pewne $a \in \mathcal{A}$
   $$\forall_{v=(v_0,v_1,\ldots,v_{k-1})\in\mathcal{A}^k}\ \forall_{b\in\mathcal{A}\setminus\{a\}} \quad q_b(v) := \frac{p_{f_v\cup\{(x_k,b)\}}}{p_{f_v}}$$
   gdzie $D(f^v) := \{x_i\}_{i=0,\ldots,k-1},\ f_v(x^i) := v_i$

   Idziemy po kolei według ustalonego ciągu po przestrzeni. W kolejnym kroku, na podstawie wartościowania poprzednich węzłów, $q$ daje nam rozkład prawdopodobieństwa wartościowania kolejnego.

   *Dowód:* Chcemy np. odtworzyć $p_f$,
   $\exists_k\ \{v_0,...,v_k\} \supset D(f),\ A = \{v_0,...,v_k\}$
   $$\forall_{u\in V(A)}\ p_u = \prod_{i=0}^{k} q_{u(x_i)}((u(x_0),...,u(x_{i-1})) \qquad \left(q_a(w) = 1 - \sum_{b\in\mathcal{A}\setminus\{a\}} q_b(w)\right)$$

$$p_f = \sum_{u \in V(A \setminus D(f)) : u \cup f_v \in V(A)} p_{f_v \cup u} \ .$$

Później uogólnimy ten opis do algorytmów probabilistycznych, pomijając założenie że wcześniejszych elementów była skończona ilość.

Czyli możemy generować elementy zgodne z danym opisem: bierzemy kolejne węzły według kolejności i według rozkładu $q$ na podstawie wartościowania poprzednich punktów generujemy wartościowanie danego węzła.
Możemy zdefiniować ***elementy statystyczne*** dla opisu $p$ jako elementy wygenerowane w ten sposób.

Mamy tutaj mamy pewien proces graniczny - na przykład dla $q = \frac{1}{2}$ rozkład średniej gęstości będzie się zachowywał jak coraz węższy gauss, a w granicy dostajemy deltę Diraca - z prawdopodobieństwem 1 wylosujemy element zgodny z tym opisem.

Elementy niestatystyczne, to np. takie w których przestrzeń można podzielić na "domeny" - nieskończone podzbiory o różnym lokalnym opisie statystycznym.

# 6 Alogrytmy probabilistyczne

Powiedzmy że znamy już opis statystyczny (najlepiej typowy). Chcemy teraz skonstruować dla niego element statystyczny. W wyniku tej konstrukcji wybieramy konkretny taki element. Więc jeśli to byłby opis typowy (elementów tego typu jest najwięcej), w tym wyborze moglibyśmy zakodować najwięcej informacji.

Narzuca się wykorzystanie opisu ciągowego z poprzedniego rozdziału.

Jednak żeby w ten sposób zapełnić przestrzeń, trzeba np. rozbudowywać wokół jakiegoś punktu, etc. W poprzednim rozdziale uzasadniliśmy, że korelacje maleją z odległością do 0, czyli jeśli chcemy dostać element z bliskiego opisu, to funkcję dającą rozkłady prawdopodobieństwa ($q$), wystarczy uzależnić od wartości najbliższych węzłów. W takim podejściu już nie musimy się już przejmować wypełnianiem przestrzeni według jakiegoś skomplikowanego schematu.

**Definicja 18.** *Algorytmem probabilistycznym* dla danego modelu,
nazywamy parę $(<, q)$*:*

- " $<$ " $\subset X^2$ - porządek liniowy na $X$
- $q_a : \{(x, v) : v \in V(x_<)\} \to [0, 1] \qquad (v \cup \{(x, a)\} \notin V \Rightarrow q_a(x, v) = 0)$

takie, że $\sum_{a \in \mathcal{A}} q_a((x, v)) = 1$

gdzie $x_< := \{y \in X : y < x\}$

W praktyce $q$ będzie zależało tylko od bliskich węzłów.

Generując algorytmem element, dostajemy pewien ciąg rozkładów prawdopodobieństwa - podobnie jak w przypadku jednowymiarowego problemu, średnia ilość informacji na węzeł, to będzie pewna średnia ważona z informacji uzyskiwanej z jednego wyboru. Nie zapiszę tego formalnie - będzie to widać na przykładach.

*Optymalność algorytmu* oceniamy po tym jak blisko jest maksymalnej wartości, czyli entropii problemu. Algorytm, który osiągał by to maksimum (kopiował opis typowy), nazywamy *algorytmem optymalnym*. W przypadku jednowymiarowym udało nam się takie skonstruować. W ogólnym przypadku nie udało mi się to. Podejdziemy jednak praktycznie dowolnie blisko.

Przeanalizuję dwa proste algorytmy dla Hard Square.

**Alogrytm I: Wypełnianie po zbiorach niezależnych**

Podzielmy przestrzeń na rozłączne zbiory $Y_i$, wewnątrz których nie działają więzy i które w sumie dają całą przestrzeń.
Np. ogólnie $Y = L\mathbb{Z}^m,\ I = \{1, ..., L^m\},\ \{x^i\}_{i\in I} = \{0, ..., L-1\}^m$ ($L$ -zasięg więzów)
Teraz kolejne zbiory to $Y_i = Y + x^i$

Dla modelu Hard Square mamy na przykład $Y_i = \{(i, j) : \mathrm{mod}(i + j, 2) = i\}$, czyli węzły o sumie współrzędnych parzystej/nieparzystej.

Teraz algorytm: wypełniamy najpierw $Y_0$ z takim samym prawdopodobieństwem (powiedzmy z prawdopodobieństwem $q$ wstawiamy 1). Potem dla pozostałych węzłów robimy podobnie, z prawdopodobieństwem $q'$. Tym razem ten wybór już nic nie psuje dalej, więc ponieważ szukamy największej możliwej entropii, możemy przyjąć $q' = 1/2$.

Czyli cały nasz algorytm jest opisywany jedną liczbą: $q$

Policzmy entropię: $H_q = \frac{1}{2}h(q) + \frac{1}{2}(1 - q)^4$
Ponieważ w połowie węzłów dostajemy entropię z $q$ a w pozostałych dostajemy 1bit/węzeł, ale tylko w węzłach, których wartości 4 sąsiadów zostało wcześniej ustalonych na 0.

Czyli najlepszy algorytm, jaki możemy uzyskać, to $\max_q H_q \cong H - 0.0217$
czyli tym algorytmem tracimy $\Delta H$ =0.0217 bit/węzeł, czyli jakieś 4%.

Dlaczego nie dostaliśmy optimum?
Możemy tak dobrać $q$, żeby spełnione było

$$(p_* :=)p\begin{pmatrix} & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{pmatrix} = p\begin{pmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{pmatrix}, \text{ ale wtedy np. } p\begin{pmatrix} 1 & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{pmatrix} > p\begin{pmatrix} 1 & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{pmatrix}$$

Uzasadnienie:
Jeśli środkowy element jest z $Y_1$ - w obydwu przypadkach jest równość ($q' = 1/2$)
Silną nierówność dostajemy, gdy środkowy element jest z $Y_0$.
Prawdopodobieństwo, że ustalimy mu wartość 1 (prawa strona) wynosi $q$. Podczas gdy żeby dostać lewą stronę, po ustaleniu wartości środka na 0, musimy ustalić wartość jego sąsiadów na 0 (to już nie jest wymuszone) - na co jest większe prawdopodobieństwo, gdy wymusiliśmy wcześniej jedynkę w rogu.

Nasuwa się spostrzeżenie, że jeśli chcielibyśmy optymalny algorytm, ponieważ mamy przeliczalną ilość warunków (np. pLTC), nie wystarczy skończona ilość parametrów.

Do numerycznego zbadania kolejnego algorytmu, będziemy potrzebować algorytmu znajdującego przybliżeń zbiorów typowych na skończonych podprzestrzeniach $A \subset X$. Przypominam, że na skończonym zbiorze wszystkie wartościowania powinny być po prostu równoprawdopodobne, czyli innymi słowy: potrzebujemy dobrego generatora takich wartościowań.

### Generator układów typowych

W algorytmie probabilistycznym każdy węzeł odwiedzamy dokładnie raz. Okazuje się, że gdy pominiemy założenie odwiedzeń, łatwo można wskazać metodę dostania typowego elementu. Jednak wykorzystanie tej metody do kodowania wydaje się wysoce niepraktyczne - trzeba by przestrzeń przechodzić wiele razy.

Dla HS ta metoda będzie polegała (po dowolnym zainicjowaniu z $V(A)$) na wielokrotnym powtarzaniu:
**losuj punkt w $A$, jeśli jego 4 sąsiadów jest ma wartości 0, zmień jego wartość**

Czyli pozwalamy układowi fluktuować zgodnie z pLTC.
Uzasadnię, że kontynuując ten proces, będziemy się zbliżać do rozkładu jednorodnego na $V(A)$.

Zauważmy, że jest to proces Markowa na $V(A)$. Z $f \in V(A)$ możemy w jednym kroku przejść do $g \in V(A)$ wtw gdy różnią się tylko w jednym węźle. Wtedy prawdopodobieństwo przejścia wynosi $\frac{1}{\#A}$. Tyle samo wynosi prawdopodobieństwo przejścia z powrotem.
Czyli macierz stochastyczna dająca ten proces jest symetryczna - jest bistochastyczna (suma wyrazów zarówno w wierszach jak i w kolumnach daje 1) - $(1, 1, ..., 1)$ jest jest wektorem własnym odpowiadającym wartości własnej 1.
Poza tym ta macierz jest nieredukowalna, czyli z tw. Frobeniusa - Perrona ta wartość własna jest niezdegenerowana, czyli iterując ten proces zbliżamy się do rozkładu jednorodnego.

Ogólnie widać, że **dowolny proces bistochastyczny** by tutaj odpowiadał.

Praktycznie używałem około 2-5 $\#A$ iteracji, a potem żeby dostać kolejny nieskorelowany około $\#A$ iteracji.

**Uśredniając po takich układach możemy liczyć opis typowy**, jednak zbieżność jest dość powolna.

Możemy się pokusić o zbadanie "intuicyjnie narzucającego się" algorytmu:

**Alogrytm II: Random seed**

Dla każdego węzła przestrzeni wylosujmy z jednorodnym rozkładem prawdopodobieństwa liczbę z $[0, 1]$ – $t : X \to [0, 1]$

Teraz kolejność: $x < y \equiv t(x) < t(y)$

Czyli w kolejnym kroku algorytmu będziemy losować niewylosowany jeszcze węzeł w przestrzeni i:
- jeśli miał już sąsiada o wartości 1, to wartościujemy go na 0,
- jeśli nie - z pewnym prawdopodobieństwem wartościujemy na 0 lub 1.

Pozostaje określić to prawdopodobieństwo.

Gdy jesteśmy w punkcie o danym $t$, to znaczy, że (procentowo) $t$ węzłom płaszczyzny już próbowaliśmy nadać wartość. Czyli narzuca się przyjęcie pewnej funkcji
*profil ładowania* - $q : [0, 1] \to [0, 1]$ - gdy jesteśmy w węźle o danym $t$ to, jeśli można, to z prawdopodobieństwem $q(t)$ wartościujemy go na 1.

Żeby obliczyć entropię, potrzebujemy dla $q$ znaleźć drugą funkcję:

$a : [0,1] \to [0,1]$ - prawdopodobieństwo, że węzłowi o danym $t$ będzie można nadać wartość 1.

Przyjmijmy, że $a, q$ - ciągłe.
Teraz entropia to będzie:

$$H_q = \int_0^1 a(t)h(q(t))dt$$

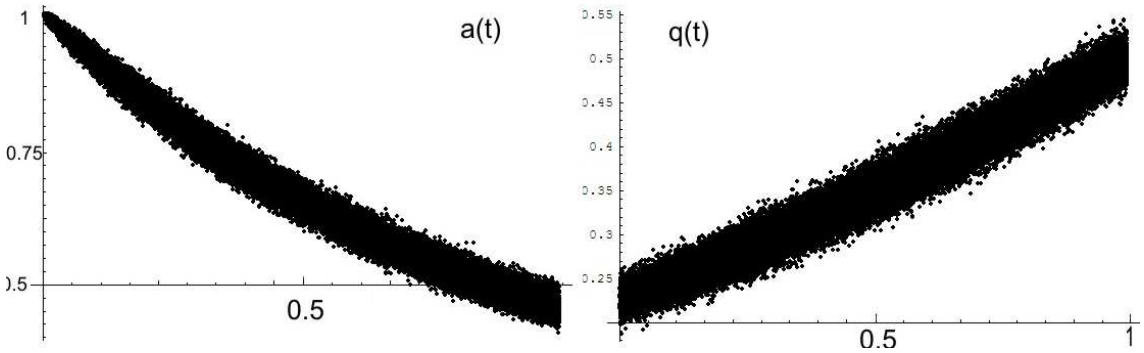
Zauważmy, że dla optymalności powinniśmy oczekiwać:

- $a(0) = 1$ - nie ma jeszcze żadnych 1
- $q(1) = \frac{1}{2}$ - te 1 już nic nie blokują
- $a(1) = 2p_*^o$ - sąsiednie 4 zera już muszą być obstawione
- $q(0) = p_*^o$ - te elementy na pewno($a = 1$) będą 1

Niestety obliczenie $a$ z $q$, a co dopiero maksymalizacja po $q$, jest daleko niebanalne.

Możemy jednak użyć podejścia w stylu metod monte-carlo:
Na pewnym skończonym zbiorze, np. kwadracie, generujemy różne kolejności elementów oraz różne elementy typowe (używając podanego generatora).

Uśredniając po wszystkich takich pomiarach możemy dostać na przykład wykresy z rysunku 5 ($A = 1, ..., 300^2$, 4000 pomiarów).




Rysunek 5: Znalezione numerycznie $q$ i $a$.

Te wykresy są bardzo rozmyte - przyczyną jest to, że za bardzo uprościliśmy - **tak na prawdę $q$ powinno też zależeć od kolejności rozmieszczania sąsiednich elementów**.
Po dopasowaniu wielomianów 4 stopnia i scałkowaniu, dostałem więcej niż entropia

modelu! To jest nauczka, że trzeba uważać przy takim średniowaniu - znalezione $a$ nie odpowiada już $q$.
Przy generowaniu tym algorytmem $(q)$ elementów, dostawałem $\Delta H \cong 0.01 - 0.02$bita/węzeł.

W kolejnym rozdziale będziemy badać algorytmy z kolejnością zachowują nie wyróżniającą żadnych punktów - uporządkowaniem leksykograficznym po współrzędnych.

# 7 Droga do optimum

W tym rozdziale wskażę praktyczną metodę, dzięki której, o ile założenie(*) jest prawdziwe, można policzyć opis dowolnie bliskiego typowemu.
Wskazuję też, jak używając ten opis, skonstruować algorytm dowolnie bliski optymalnemu.
Badania numeryczne wykazały, że dla HS rzeczywiście podchodzimy dowolnie blisko optimum.

Idea tego podejścia to przybliżenie tego problemu tak, żeby można go było sprowadzić do modelu jednowymiarowego:

1. Uproszczenie modelu: wszystkie wymiary oprócz jednego(*wyróżnionego*) obcinamy do skończonej wartości
2. Nowy alfabet: jako przestrzeń stanów wprowadzamy na przykład wszystkie możliwe przekroje (w kierunku prostopadłym do wyróżnionego) o szerokości $L - 1$
3. Macierz transferu: wprowadzamy macierz (w literaturze nazywana macierzą transferu) możliwych kolejnych (według wyróżnionego kierunku) stanów
4. Rozwiązanie problemu jednowymiarowego: na podstawie metody z drugiego rozdziału
5. Znalezienie algorytmu: na podstawie tego opisu znajdujemy algorytm, gdzie jako kolejność przyjmujemy, że wypełniamy przekroje po kolei, a na przekroju przyjmujemy taką kolejność, że wszystkie węzły są tak samo traktowane (np. leksykograficznie względem niewyróżnionych współrzędnych)

Dobrze zrobić jeszcze jeden krok:

6. Ocena algorytmu: używając algorytmu zrekonstruować opis statystyczny potrzebny do obliczenia prawdopodobieństw poszczególnych zdarzeń i obliczyć entropię.

Prześledzę teraz te kroki dla HS.

1. Wyróżniam kierunek (1,0) na płaszczyźnie. Przyjmuję szerokość obcięcia $n$. Nowa przestrzeń, to
$$Y := \mathbb{Z}(1,0) + \overline{n}(1,1)$$
gdzie $\overline{n} := \{0, ..., n-1\}$

   Pozostaje nam wprowadzić warunki brzegowe, można to zrobić na przykład na 2 sposoby:

   (a) *cykliczne*: $\forall_i v(i+n, n) := v(i, 0)$

   (b) *zerowe*: $\forall_i v(i, -1) = v(i, n) := 0$

   O warunkach cykliczności możemy myśleć jako o dodatkowych więzie - czyli entropia tu uzyskana będzie mniejsza niż prawdziwa.
   Natomiast o warunkach zerowych możemy myśleć jako o pasach pokrywających całą przestrzeń. Tutaj od HS różnimy się tym, że między liniami $ni$ a $ni+1$ ($i \in \mathbb{Z}$) nie obowiązują więzy (mogą być sąsiednie jedynki) - zmniejszamy ilość więzów - dostaniemy entropię większą niż prawdziwa.

   Czyli **możemy dowolnie przybliżać entropię z dołu i z góry** zależnie od przyjętych warunków brzegowych.

   Nas nie interesuje liczenie entropii, tylko rozkładu prawdopodobieństwa. Chcielibyśmy, żeby dla danego $n$ nowy alfabet był jak najmniejszy. W tym celu najlepsze będą warunki cykliczne - wiele elementów będziemy mogli utożsamić.

2. Szerokość pasów wystarczy 1.
   Idziemy na ukos - więzy nie działają: nowy alfabet, to $\mathcal{A}' := (1,1)\overline{n} \to \{0,1\}$ tyle że utożsamiamy elementy $v, w \in \mathcal{A}'$, takie że:

   (a) przesunięcie cykliczne: $\exists_k\ \forall_i\ v(i) = w(i+k \bmod n)$

   (b) odbicie: $\forall_i\ v(i) = w(n-1-i)$

   (c) nieistotne zero: $v(0) = v(1) = v(2) = 1 \wedge w(1) = 0 \wedge \forall_{i \neq 1} v_i = w_i$

   Pierwsze dwa warunki wynikają z cykliczności.
   Trzeci mówi, że te dwie jedynki blokują sąsiadów środkowego 0 - takie samo będzie zachowanie, jakby tam była 1.
   Ten warunek jest właśnie powodem, dla którego wybrałem ukośne pasy:

Dla pionowych pasów mamy warunek, że nie ma 2 sąsiednich jedynek - ich ilość rośnie jak $\varphi^n \cong 1.618^n$. Natomiast dla ukośnych dla małych $n$ mniej więcej jak $1.5^n$.

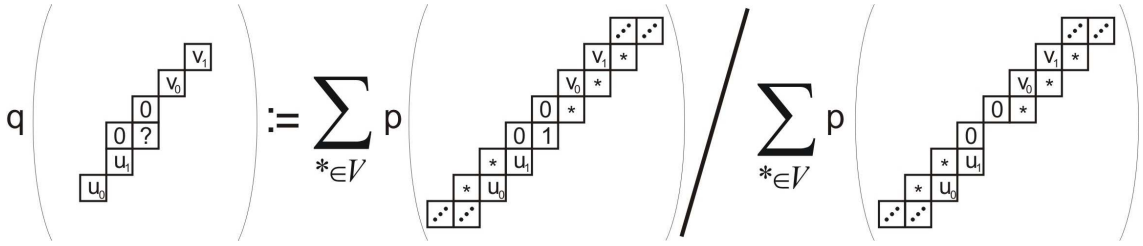
Dzięki tym trzem utożsamieniu np. dla $n = 21$ zamiast $2^{21}$ symboli dostajemy 609 razy mniej: 3442.

3. $M_{vw} = 1 \Leftrightarrow u \in V$
   gdzie $D(u) := \{(0,0),(1,0)\} + (1,1)\overline{n},\ u(i,i) := v(i),\ u(i+1,i) := w(i)$.

4. teraz dla macierzy $M$ znajdowałem metodą potęgową wektory własne i na podstawie metody z drugiego rozdział: macierz procesu Markowa $S$ (witać tu analogię markowowskość - LTC). Wartość własną brałem z [6]. Czyli mamy przybliżenie opisu.

5. **Algorytm III: Pasmowy** typu (a,b)
   Kolejność: $(i,j) < (k,l) \Leftrightarrow (i-j < k-l) \vee (i-j = k-l \wedge i < k)$
   $q : \{0,1\}^a \times \{0,1\}^b \to [0,1]$
   $q(u,v) =$ jeśli nie ma sąsiednich jedynek i
   $u$ - wartościowanie poprzednich $a$ węzłów,
   $v$ - wartościowanie $b$ węzłów, które będą wpływać na następne węzły,
   to z takim prawdopodobieństwem nadaj węzłowi (?) wartość 1.

Wartości $q$ dostajemy korzystając z opisu dla dwóch pasów - patrz rysunek dla $a = b = 2$




Rysunek 6: Liczenie $q$.

Oto przykładowe algorytmy znalezione w ten sposób:

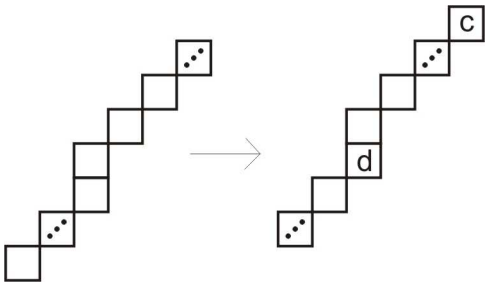
(a) $a = b = 0$ $\quad q = 0.3602994$

(b) $a = b = 1$ $\quad q = \begin{pmatrix} 0.3365899 & 0.2946553 \\ 0.4406678 & 0.3999231 \end{pmatrix}$

(c) $a = b = 2$ $\quad q = \begin{pmatrix} 0.3350090 & 0.2918920 & 0.3265314 & 0.2910261 \\ 0.3507870 & 0.3075943 & 0.3423356 & 0.3067152 \\ 0.4419816 & 0.4001011 & 0.4342912 & 0.3992940 \\ 0.4421921 & 0.4004149 & 0.4345211 & 0.3996084 \end{pmatrix}$

Gdzie numer wiersza oznacza $u$ traktowany jako liczbę w układzie dwójkowym - numeracja bitów rośnie w kierunku (1,1) (jak na rysunku).
Analogicznie numer kolumny oznacza $v$.

6. Oceniamy algorytm.
Żeby skorzystać z wzoru na entropię, trzeba znać rozkład prawdopodobieństwa używanych $(u, v)$ dla $q$.
W tym celu trzeba znaleźć opis probabilistyczny (dla kształtu jak na rysunku poniżej), który jest opisem dawanym przez algorytm (powinien być bliski znalezionemu wcześniej).
Powinien być on punktem stałym następującego przekształcenia:




Rysunek 7: Iteracja, której szukany opis powinien być punktem stałym.

gdzie rozkład dla $d$ znajdujemy z algorytmu,
natomiast z $c$ jest problem - nie możemy go bezpośrednio znaleźć. Możemy jednak przyjąć, że z przybliżonego aktualnie opisu możemy znaleźć przybliżony opis dla $(1, 1)\overline{k}$ - stąd biorę $c$.

Czyli szukamy punktu stałego pewnego skomplikowanego przekształcenia. Pamiętajmy jednak, że przyjmujemy że ten opis jest bliski znalezionemu wcześniej (aproksymacji typowego).

Algorytm:

(a) jako opis początkowy przyjmij znaleziony wcześniej

(b) dopóki w kolejnej iteracji dostajemy opis dalszy od tego z poprzedniej niż pewna ustalona granica:

i. z aktualnego opisu oblicz opis dla $(1,1)\overline{k}$

ii. na podstawie algorytmu i opisu $(1,1)\overline{k}$ przeprowadź kilka iteracji z rysunku

W przypadkach które badałem, okazało się że powyższy algorytm jest szybko zbieżny.
Oto wyniki: $-\log_{10}(H - H_q)$ dla różnych $a,\ b$:

| $a \setminus b$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 2.06 | 2.113 | 2.1153 | 2.1153 | 2.1153 | 2.1153 |
| 1 | 3.32 | 3.82 | 3.99 | 4.001 | 4.002 | 4.003 |
| 2 | 3.50 | 4.37 | 5.19 | 5.44 | 5.466 | 5.436 |
| 3 | 3.50 | 4.42 | 5.55 | 6.43 | 6.74 | 6.78 |
| 4 | 3.50 | 4.42 | 5.58 | 6.71 | 7.60 | 7.96 |
| 5 | 3.50 | 4.42 | 5.58 | 6.74 | 7.83 | 8.72 |

Widać że najlepiej brać $a$ i $b$ podobne.

### Inicjalizacja

Gdy w praktyce chcielibyśmy korzystać z tego algorytmu, potrzeba go jeszcze zainicjalizować. Walczymy o mały ułamek bit/węzeł na brzegu pewnego zbioru - możemy wziąć praktycznie dowolny rozkład i straty informacji będą marginalne.

Gdyby nam jednak zależało na optymalnym wykorzystaniu tych węzłów, na pierwszy rzut oka, powinniśmy użyć np. zwykłego opisu statystycznego dla $(1,1)\overline{k}$. Przypomnijmy jednak, że z lewej strony nie będziemy mieć ograniczeń - powinniśmy tutaj zmieścić więcej informacji niż normalnie.

Ogólnie - przyjmijmy, że znamy rozkład prawdopodobieństwa "poprzedniego pasa" (np. same zera z prawdopodobieństwem 1).
Na podstawie podstawie znanego nam procesu Markowa $(S)$ dostajemy rozkład prawdopodobieństwa par pasów: poprzedni-ten. Stąd znajdujemy algorytm który w tym pasie powinniśmy stosować.

Np. gdy wcześniej nic nie było, możemy przyjąć, że "poprzedni pas" był wypełniony samymi zerami. Czyli z $S_{0v}$ dostajemy algorytm wartościowania pierwszego pasa.
W kolejnym pasie rozkład poprzedni($v$)-ten($u$) to $p(u,v) = S_{0v}S_{vw}$ - dostajemy algorytm.
Oczywiście postępując tak dalej coraz bardziej zbliżamy się do ogólnego algorytmu.

# 8 Podsumowanie

W skończonych przestrzeniach, gdy mamy dany pewien rozkład prawdopodobieństwa ($> 0$) elementów, każdy z nich może wystąpić. Jednak gdy z rozmiarem przestrzeni zmierzamy do nieskończoności(granica termodynamiczna) - będziemy mieli do czynienia z sytuacją podobną do tej w twierdzeniach granicznych z rachunku prawdopodobieństwa - będziemy zmierzać do pewnej delty diraca w przestrzeni parametrów. Sytuacja wygląda tak: mamy pewne więzy - przetłumaczamy je na równania na parametry (np. $p(11) = 0$). Teraz w tej okrojonej przestrzeni parametrów szukamy takich, które maksymalizują entropię (opis typowy) - elementy te całkowicie zdominują pozostałe w granicy termodynamicznej.
Jak szukać takiego opisu? Po pierwsze - wybieramy sposób opisu, żeby wykorzystać naturalne symetrie modelu - opis powinien je zachowywać. Teraz, ponieważ wiemy że szukane elementy dominują pozostałe, możemy wyśredniować po wszystkich elementach - wkład od pozostałych będzie nieistotny.
Przy skończenie zasięgowych więzach możemy dzielić przestrzeń na dwa podzbiory (jeden skończony), które są rozdzielone pewnym zbiorem, tak że tamte dwa podzbiory nie wpływają na siebie - możemy je rozważyć osobno. Na zbiorze skończonym jednak łatwo z reguły przewidzieć zachowanie. Dostajemy dzięki temu zestaw koniecznych (często i wystarczających) do typowości (LTC) - czyli typowość staje się warunkiem lokalnym.
Trzeci sposób na szukanie typowego opisu, jest po prostu maksymalizacja po entropii. Dla podanych w pracy opisów statystycznych - liczenie entropii jest dość proste: wypełniamy przestrzeń pewnym ciągiem i entropię liczymy punkt po punkcie.
Dla modeli jednowymiarowych, znalezienie typowego opisu okazuje się dość proste - z LTC jest dany łańcuchem markowa. Dla wielowymiarowych już potrzeba nieskończenie wiele parametrów. Możemy je znaleźć np. uśredniając po wielu losowych elementach lub obcinając wszystkie oprócz jednego wymiaru przesztrzeni do skończonej wartości i korzystać z metod jednowymiarowych. Na przykład dla HS ta metoda daje praktycznie dowolne przybliżenie typowości.
Mając teraz pewne przybliżenie opisu typowego, możemy generować elementy zgodne z tym opisem. Gdy do tego generowania użyjemy pewnej informacji, tak żeby ten proces był odwracalny - dostaniemy kodowanie. Możemy je np. wykorzystać do zwiększenia pojemności informacyjnej nośnika.

# Literatura

[1] N. Calkin, H.Wilf, *The number of independent sets in a grid graph*, SIAM J. Discrete Math 11 (1998) 1,54-60

[2] Zs. Nagy, K. Zeger, *Bit Stuffing Algorithms and Analysis for Run Length Constraned Channels in Two and Three Dimensions*, IEEE Transactions of Information Theory 50(2004) 1, 3146-3169

[3] A. Kato, K. Zeger, *On the capacity of two-dimensional run-length constrained channels*, IEEE Transactions of Information Theory 45(1999) 5,1527-1540

[4] A. Burstein, *Independent sets in certain classes of (almost) regular graphs*, arXiv:math.CO/0310379

[5] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic, London (1982)

[6] R. J. Baxter, *Planar lattice gases with nearest-neighbour exclusion*, arXiv:cond-mat/9811264

[7] R. J. Baxter, *Variational Approximations for Square Lattice Models in Statistical Mechanics*, J. Stat. Phys. 19 (1978) 461 – 478

[8] W. Guo, H. W. J. Blöte, *Finite-size analyzis of the hard-square lattice gas*, Physical Review E 66, 046140 (2002)

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, *Introduction to Algorithms*, Mc Graw Hill, 1990

[10] J. J. Binney, N. J. Dowrick, A. J. Fisher, M. E. J. Newman, *The Therory of Critical Phenomena*, Oxford University Press, Oxford, 1992