Spectrum finder algorithm

February 20, 2017

$T = (V, E)$ — is Tanner Graph of base matrix B.

M is fixed expand modulo.

Edges in path have orientation, so e^{st} — edge start vertex and e^{en} — edge end vertex, $e^{backEdge}$ — same edge but going in opposite direction.

Closed path is a sequence of edges $e_0, e_1, \dots, e_l$ so that $e_i^{en} = e_{i+1}^{st}$ and $e_l^{en} = e_0^{st}$. Additional constraint for our case is $e_i^{backEdge} \neq e_{i+1}$. Different closed paths which can be constructed from one another with cycle shift or reverse are equivalent.

Algorithm finds number of closed path with length from 1 to L.

Let's say that we have edge marking $F : E \to \mathbb{N}$ so that $0 \leq F(e) < M$ and $e^c = F(e)$.

Number r called *order* of path $p = e_0, \dots e_l$ if it is maximal number so that p can be presented in form $\underbrace{ss\dots s}_{r \text{ times}}$ where s is subpath.

Algorithm presented below count paths without cycle shift equivalence, e.g, path of order 1 and length l will be counted l times, path of order 2 will be counted $l/2$ times and so on. To fix it we can use inclusion-exclusion principle, which will be described after algorithm.

ALGORITHM WITHOUT CYCLE SHIFT EQUIVALENCE

Iterate over all possible starting edges $e_{start} = (startVertex, u)$.

$dp[e][l][w]$ is number of paths which starts with e_{start} and ends with edge e which consist of l edges and $\sum_{e \in path} e^c \equiv w \pmod{M}$.

Initialization:

$$dp[e_{start}][1][e^c] = 1$$

Iteration $l = 2..L$**:**

$$\forall e \in E\ \forall w \in [0, M)\ dp[e][l][w] = \sum_{\substack{e_{prev} \neq e^{backEdge} \\ e_{prev}^{en} = e^{st}}} dp[e_{prev}][l-1][w - e_{prev}^c]$$

Then save it to fix cycle shifts:

$$cnt[l] = \sum_{\substack{e \\ e^{en} = startVertex \\ e^{backEdge} \neq e_{start}}} dp[e][l][0]$$

FIXING CYCLE SHIFTS

We need to count every path of order r exactly r times, so that later we can just divide $cnt[l]$ by $2l$ because every cycle of length l will be counted $2l$ times — l cycle shifts and 2 directions.

$cntOrder[r][l]$ — number of paths of order r and length l.

$$cnt[l] = \sum_{d|l} cntOrder[d][l]$$

and we need

$$fcnt[l] = \sum_{d|l} d \cdot cntOrder[d][l]$$

Let's define additional function

$$add[i] = i - \sum_{\substack{d|i \\ d>1}} add[i/d]$$

Then

$$i = \sum_{d|i} add[i]$$

We'll prove that:

$$fcnt[l] = \sum_{d|l} add[d] \cdot cnt[l/d]$$

Proof:

$$c[l] = cntOrder[1][l]$$

$$cntOrder[d][l] = c[l/d]$$

Let's consider $fcnt[l]$ as polynomial of variables $c[l_1]$ and check that every coefficient equals to what it should be:

$$fcnt[l] = \sum_{d_1|l} add[d_1] \cdot cnt[l/d_1] = \sum_{d_1|l} add[d_1] \cdot \sum_{d_2|(l/d_1)} c[l/(d_1 d_2)] = \sum_{d|l} k_d c[d]$$

$$fcnt[l] = \sum_{d|l} d \cdot cntOrder[d][l] = \sum_{d|l} d \cdot c[l/d]$$

So we should prove that $k_d = \frac{l}{d}$:

$$k_d = \sum_{d_1|(l/d)} add[d_1] = (\frac{l}{d} - \sum_{\substack{d_2|l/d \\ d_2>1}} add[l/(dd_2)]) + \sum_{\substack{d_1|(l/d) \\ d_1<l/d}} add[d_1] =$$

$$= \frac{l}{d} + (-\sum_{\substack{d_3|l/d \\ d_3<l/d}} add[d3] + \sum_{\substack{d_1|(l/d) \\ d_1<l/d}} add[d_1]) = \frac{l}{d} \blacksquare$$

Case with weights:

Actually it's a little bit more complicated, we need to consider weight of path as well, so that it will be rewritten as:

$$cnt[w][l] = \sum_{\substack{e \\ e^{en}=startVertex \\ e^{backEdge}\neq e_{start}}} dp[e][l][w]$$

$$cnt[w][l] = \sum_{d|l} cntOrder[w][d][l]$$

$$fcnt[w][l] = \sum_{d|l} d \cdot cntOrder[w][d][l]$$

We'll prove that:

$$fcnt[w][l] = \sum_{\substack{d|l \\ w_1\in[0,M) \\ w_1\cdot d=w \pmod M}} add[d]\cdot cnt[w_1][l/d]$$

Proof:

$$c[w][l] = cntOrder[w][1][l]$$

$$cntOrder[w][d][l] = \sum_{\substack{w_1\in[0,M) \\ w_1\cdot d=w \pmod M}} c[w_1][l/d]$$

Let's consider $fcnt[w][l]$ as polynomial of variables $c[w_1][l_1]$ and check that every coefficient equals to what it should be:

$$fcnt[w][l] = \sum_{\substack{d_1|l \\ w_1\in[0,M) \\ w_1\cdot d_1=w \pmod M}} add[d_1]\cdot cnt[w_1][l/d_1] =$$

$$\sum_{d_1|l} add[d_1]\cdot \sum_{w_1\cdot d_1=w \pmod M}^{w_1\in[0,M)} \sum_{\substack{d_2|(l/d_1) \\ w_2\in[0,M) \\ w_2\cdot d_2=w_1 \pmod M}} c[w_2][l/(d_1 d_2)] =$$

$$\sum_{d_1|l} add[d_1]\cdot \sum_{\substack{d_2|(l/d_1) \\ w_3\in[0,M) \\ w_3\cdot d_1\cdot d_2=w \pmod M}} c[w_3][l/(d_1 d_2)] =$$

$$\sum_{\substack{d|l \\ w_1\in[0,M) \\ w_1\cdot d=w \pmod M}} k_d^{w_1} c[w_1][d]$$

$$fcnt[w][l] = \sum_{d|l} d\cdot cntOrder[w][d][l] = \sum_{d|l} d\cdot c[w][l/d]$$

So we should prove that $k_d^w = \frac{l}{d}$ and it's exactly the same as without weights:

$$k_d^w = \sum_{d_1|(l/d)} add[d_1] = (\frac{l}{d} - \sum_{\substack{d_2|l/d \\ d_2>1}} add[l/(dd_2)]) + \sum_{\substack{d_1|(l/d) \\ d_1<l/d}} add[d_1] =$$

$$= \frac{l}{d} + (-\sum_{\substack{d_3|l/d \\ d_3<l/d}} add[d3] + \sum_{\substack{d_1|(l/d) \\ d_1<l/d}} add[d_1]) = \frac{l}{d} \blacksquare$$

Then we add $fcnt[0][l]$ to $spectrum[l]$.

And finally after processing all possible starting edges we divide every $spectrum[l]$ by $2l$ to compress equivalent pathes.