The Symbolic Goulden Jackson Cluster Method

Xiangdong Wen

1. Introduction

Let V and B be a finite alphabet and a finite set of $bad\ words$ respectively. Suppose $a(n)$ is the total number of words with length n that avoid the bad words in B as factors. The aim is to find the generating function

$$f(s) = \sum_{n=0}^{\infty} a(n)s^n \tag{1}$$

in an efficient way.

The Goulden-Jackson cluster method[GoJ1,GoJ2] has been beautifully explained, extended, and implemented by J. Noonan and D. Zeilberger([**?**]). The Goulden-Jackson cluster method needs to solve a system of $|B|$ linear equations and it is much more efficient than the Naive approach which needs to solve a system of $\sum |d|^{|v|-1}$, $v \in B$, linear equations. However, their Maple packages require that the cardinality of the alphabet is a *numeric* argument, rather than *symbolic. In this paper, I extend the method into the latter case, thereby initiating the Symbolic Goulden-Jackson Method.*

2. Review of the Goulden Jackson Cluster method

Recall that a **factor** *of a word $w_1 w_2 \cdots w_n$ is any of the words $w_i w_{i+1} \cdots w_{j-1} w_j$, for $1 \le i \le j \le n$. Such a factor is represented by $[i,j]$. Two factors $[i,j]$ and $[i',j']$* **overlap** *if they have at least one common letter, i.e. $i < i' \le j$.*

Define the **weight** *of a word w as $weight(w) = s^{|w|}$, where $|w|$ is the length of w. Obviously, the generating function (1) becomes*

$$f(s) = \sum_{w \in \mathcal{L}(B)} weight(w),$$

where $\mathcal{L}(B)$ is the set of all words that avoid the members of B as factors.

A word with some factors marked is called a **marked word**. *Here we only consider the case where the marked factors are the words in B. A marked word could be written in the following form:*

$$(w; [i_1, j_1], [i_2, j_2], \cdots, [i_l, j_l]),\ \textit{where } [i_r, j_r]\ 1 \le r \le l,\ \textit{are the marked factors.}$$

For example, let $V = \{1,2,3\}$, $B = \{123, 231, 312\}$ and $w = 12312$. There are 2^3 marked words for w:

$$\begin{array}{llll} (12312;), & (12312;[1,3]), & (12312;[2,4]), & (12312;[3,5]), \\ (12312;[1,3],[2,4]), & (12312;[1,3],[3,5]), & (12312,[2,4],[3,5]), & (12312;[1,3],[2,4],[3,5]). \end{array}$$

Define the $\overline{\textbf{weight}}$ *of a marked word w with marked factors S, where $S \subset B$, as:*

$$\overline{weight}(w, S) = (-1)^{|S|} s^{|w|},$$

where $|S|$ is the cardinality of S and $|w|$ the length of w.

Let V^* be the set of all words generated by V, and $Bad(w)$ be the set of bad factors in w. We have

Theorem 1 *:*

$$f(s)=\sum_{w\in\mathcal{L}(B)} weight(w)=\sum_{w\in V^*}\sum_{S\subset Bad(w)}\overline{weight}(w,S). \tag{3}$$

Proof*: The basic idea in the proof is to use the inclusion-exclusion principle. Let $N_B(w)$ denote the number of marked factors, that belong to B, of w. Then,*

$$\begin{aligned} f(s) &= \textstyle\sum_{w\in\mathcal{L}(B)} weight(w) \\ &= \textstyle\sum_{w\in V^*} weight(w)0^{N_B(w)} \\ &= \textstyle\sum_{w\in V^*} s^{|w|}[1+(-1)]^{N_B(w)} \\ &= \textstyle\sum_{w\in V^*} s^{|w|}\sum_{S\subset Bad(w)}(-1)^{|S|} \\ &= \textstyle\sum_{w\in V^*}\sum_{S\subset Bad(w)}(-1)^{|S|}s^{|w|} \\ &= \textstyle\sum_{w\in V^*}\sum_{S\subset Bad(w)}\overline{weight}(w,S) \end{aligned}$$

■

By the theorem, the calculation of the generating function (1) is then transfered to finding the generating function for the weighted marked words (3), and it is much easier to weight-count by the Goulden-Jackson cluster method.

A **cluster** *is a marked word*

$$(w_1w_2\cdots w_n;[i_1(=1),j_1],[i_2,j_2],\cdots,[i_l,j_l(=n)]),$$

where $[i_k,j_k]$ overlaps with $[i_{k+1},j_{k+1}]$, for all $k=1...l-1$.

A marked word is either an empty word, or ends with a letter that is not part of a cluster, or ends with a cluster. Peeling-off the maximal cluster, we get a smaller marked word. So we have the following decomposition:

$$\mathcal{M}=\{empty\_word\}\cup\mathcal{M}V\cup\mathcal{M}\mathcal{C}.$$

Taking weight on both sides and solve for $\overline{weight}(\mathcal{M})$ we have,

$$f(s)=\overline{weight}(\mathcal{M})=\frac{1}{1-ds-\overline{weight}(\mathcal{C})} \quad . \tag{4}$$

The only step left is to find $\overline{weight}(\mathcal{C})$.

For a given word $w=w_1w_2\cdots w_n$, let $HEAD(w)$ be the set of all proper prefixes:

$$HEAD(w):=\{w_1w_2\cdots w_k|k=1,2,\cdots,n-1\},$$

and $TAIL(w)$ be the set of all proper suffixes

$$TAIL(w):=\{w_kw_{k+1}\cdots w_n|k=2,3,\cdots,n\},$$

and let

$$OVERLAP(u,v):=TAIL(u)\cap HEAD(v).$$

Let u/v denote the operation of the word u chopping off its head v. For example: $12321/12=321$. *Let*

$$u:v=\sum_{x\in OVERLAP(u,v)}\overline{weight}(v/x).$$

The set of clusters $\mathcal{C}$, can be partitioned into

$$\mathcal{C}=\bigcup_{v\in B}\mathcal{C}[v] \quad ,$$

where $\mathcal{C}[v]$ ($v \in B$), is the set of clusters whose last entry is v.

Given a cluster in $\mathcal{C}[v]$, $v \in B$, it either consists of just v, or chopping v results in a smaller cluster in $\mathcal{C}[u]$, $u \in B$ if $OVERLAP(u,v)$ is not empty. On the other hand, given a cluster in $\mathcal{C}[u]$, we could always reconstitute the bigger cluster in $\mathcal{C}[v]$ by adding some words in $OVERLAP(u,v)$. Hence there exists a bijection:

$$\mathcal{C}(v) \leftrightarrow \{(v;[1,|v|])\} \bigcup_{u \in B} \mathcal{C}[u] OVERLAP(u,v).$$

Taking weights on both sides, we have:

$$\overline{weight}(\mathcal{C}[v]) = (-1)\,\overline{weight}(v) - \sum_{u \in B} (u:v)\,\overline{weight}(\mathcal{C}[u]). \tag{5}$$

This is a system of $|B|$ linear equations with $|B|$ unknowns $\overline{weight}(\mathcal{C}[v]), v \in B$.

After solving these equations we could simply obtain $\overline{weight}(\mathcal{C})$ by: $\overline{weight}(\mathcal{C}) = \sum_{v \in B} \overline{weight}(\mathcal{C}[v])$

Notice that $\overline{weight}(\mathcal{C})$ is independent of the cardinality of the alphabet, we can see from equation (4) that the symbolic Goulden Jackson could be easily implemented.

3 Symmetric Cases

Given an alphabet $V = \{1,2,3\}$, let us find the generating function for the number of words which do not have three consecutive different letters as factors, i.e. $B = \{123, 132, 213, 231, 321\}$.

By the Goulden-Jackson cluster method, we need set up and solve a system of $|B| = 6$ linear equations with six unknowns $\overline{weight}(\mathcal{C}[v])$, $v \in B$:

$$\begin{cases} \overline{weight}(\mathcal{C}[123]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[312]) - s^2\,\overline{weight}(\mathcal{C}[321]) - s\,\overline{weight}(\mathcal{C}[231]) \\ \overline{weight}(\mathcal{C}[132]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[231]) - s^2\,\overline{weight}(\mathcal{C}[213]) - s\,\overline{weight}(\mathcal{C}[321]) \\ \overline{weight}(\mathcal{C}[213]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[312]) - s^2\,\overline{weight}(\mathcal{C}[321]) - s\,\overline{weight}(\mathcal{C}[132]) \\ \overline{weight}(\mathcal{C}[231]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[132]) - s\,\overline{weight}(\mathcal{C}[312]) \\ \overline{weight}(\mathcal{C}[312]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[213]) - s^2\,\overline{weight}(\mathcal{C}[231]) - s\,\overline{weight}(\mathcal{C}[123]) \\ \overline{weight}(\mathcal{C}[321]) &= -s^3 - s^2\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[132]) - s\,\overline{weight}(\mathcal{C}[213]) \end{cases}.$$

By the symmetry of B, all the clusters $\mathcal{C}[v]$ have the same generating function $\overline{weight}(\mathcal{C})$. Thus we can reduce these six equations to one equation:

$$\overline{weight}(\mathcal{C}[123]) = -s^3 - 2s^2\,\overline{weight}(\mathcal{C}[123]) - s\,\overline{weight}(\mathcal{C}[123]).$$

After solving it we have

$$\overline{weight}(\mathcal{C}) = 6\,\overline{weight}(\mathcal{C}[123]) = \frac{-6s^3}{1+2s^2+s},$$

and

$$f(s) = \frac{1}{1-3s-\frac{-6s^3}{1+2s^2+s}} = -\frac{2s^2+s+1}{s^2+2s-1}.$$

Assuming the cardinality of the alphabet V changed, $V = \{1,2,3,\cdots d\}$, let us find the generating function for the number of words which do not have three consecutive different letters as factors, i.e.

$$B = \{123, 124, 125, \cdots, d(d-1)(d-3), d(d-1)(d-2)\}.$$

By the original Goulden-Jackson cluster method, we need set up and solve a system of $|B| = d(d-1)(d-2)$ *linear equations. Notice the symmetry of* B*, we only need set up and solve one equation:*

$$\overline{weight}(\mathcal{C}[123]) = -s^3 - (d-1)(d-2)s^2\,\overline{weight}(\mathcal{C}[123]) - (d-2)s\,\overline{weight}(\mathcal{C}[123]).$$

Thus,

$$\overline{weight}(\mathcal{C}) = d(d-1)(d-2)\,\overline{weight}(\mathcal{C}[123]) = \frac{-d(d-1)(d-2)s^3}{1+(d-1)(d-2)s^2+(d-2)s},$$

and

$$f(s) = \frac{1}{1-ds-\frac{-d(d-1)(d-2)s^3}{1+(d-1)(d-2)s^2+(d-2)s}} = \frac{(-d^2+3d-2)s^2+(-d+2)s-1}{(d-2)s^2+2s-1}.$$

In general, if the set of bad words B *is invariant under the action of the symmetric group, we could take advantage of the symmetry of* B*.*

Two words u,v are **equivalent**, $u \equiv v$*, if there exists a permutation* λ *such that* $\lambda(u) = v$*. By symmetry, all the elements in the equivalence class of* v *have the same cluster generating function* $\overline{weight}(\mathcal{C}[v])$ *.*

Define the **dimension** *of a letter* v*,* $dim(v)$*, as the number of different letters appeared in* v*. Then the equivalence class of* v *has* $\binom{d}{dim(v)}$ *different words.*

Suppose the bad words set B *is partitioned into different equivalent classes* $B_1, B_2, B_3, \cdots, B_k$*, and* $b_1, b_2, b_3, \cdots, b_k$ *are the representives respectively. Define* $(b_i : B_j) := \sum_{b\in B_j}(b_i : b)$*, then the system (5) becomes*

$$\overline{weight}(\mathcal{C}[b_i]) = -\overline{weight}(b_i) - \sum_{j=1}^{k}(b_i : B_j)\overline{weight}(b_j),\ i = 1,\cdots,k. \tag{7}$$

It is a system of k linear equations with k unknowns $\overline{weight}(\mathcal{C}[b_i])$*,* $i = 1,\cdots,k$*. Remember that k is the number of different equivalent classes in* B*. Now there are many fewer equations and many fewer unknowns than in the original Goulden-Jackson cluster method, and thus everything is much more efficient. After solving the system, we could obtain* $\overline{weight}(\mathcal{C})$ *by*

$$\overline{weight}(\mathcal{C}) = \sum_{i=1}^{k}\binom{d}{dim(b_i)}\overline{weight}(\mathcal{C}[b_i]). \tag{8}$$

Given $u = u_1u_2u_3\cdots u_n$*, let* $H_i(u)$ *be the HEAD of u with length i, i.e.* $H_i(u) := u_1u_2\cdots u_i$*. Let* $T_i(u)$ *be the TAIL of u with length i, i.e.* $T_i(u) := u_{n-i+1}u_{n-i+2}\cdots u_{n-1}u_n$*. We have*

$$b_i : B_j = \sum_{m=1}^{min(|b_i|,|b_j|)-1} I(T_m(b_i) \equiv H_m(b_j))\binom{d}{dim(b_j) - dim(H_m(b_j))}s^{|v|-m},$$

where

$$I(T_m(b_i) \equiv H_m(b_j)) = \begin{cases} 1, & \text{if } T_m(b_i) \text{ and } H_m(b_j) \text{ are equivalent },\\ 0, & \text{otherwise.}\end{cases}$$

In the two examples below, the first can still be done with the unextended Goulden-Jackson, since the number of letters is numeric, 3, but the second one requires the new extension, since the number of letters is d, i.e. a symbol.

Example 1*: Let* $V = \{1, 2, 3\}$*. Find the generating function for the number of words which do not have three consecutive different letters or three consecutive same letters as factors, i.e.*

$$B = \{123, 132, 213, 231, 312, 321, 111, 222, 333\}.$$

By the symmetry of B we have,

$$\overline{weight}(\mathcal{C}[123]) = \overline{weight}(\mathcal{C}[132]) = \cdots = \overline{weight}(\mathcal{C}[321]),$$

and

$$\overline{weight}(\mathcal{C}[111]) = \overline{weight}(\mathcal{C}[222]) = \overline{weight}(\mathcal{C}[333]).$$

Thus by system (5) and equation (8), we have

$$\begin{cases} \overline{weight}(\mathcal{C}[123]) &= -s^3 - 2s^2\,\overline{weight}(\mathcal{C}[123]) - s\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[111]) \\ \overline{weight}(\mathcal{C}[111]) &= -s^3 - 2s^2\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[111]) - s\,\overline{weight}(\mathcal{C}[111]) \end{cases},$$

and

$$\overline{weight}(\mathcal{C}) = 6\,\overline{weight}(\mathcal{C}[321]) + 3\,\overline{weight}(\mathcal{C}[111].$$

Solving the system, finally we have

$$f(s) = \frac{1}{1-3s-\overline{weight}(\mathcal{C})} = \frac{1}{1-3s-[6\,\overline{weight}(\mathcal{C}[321]) + 3\,\overline{weight}(\mathcal{C}[111])]} = -\frac{3s^2+s+1}{2s-1}$$

Example 2: *Let $V = \{1,2,3,\cdots,d\}$. Find the generating function for the number of words which do not have three consecutive different letters or three consecutive same letters as factors.*

By (5) and (8), we have

$$\begin{cases} \overline{weight}(\mathcal{C}[123]) &= -s^3 - (d-1)(d-2)s^2\,\overline{weight}(\mathcal{C}[123]) - (d-2)s\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[111]) \\ \overline{weight}(\mathcal{C}[111]) &= -s^3 - (d-1)(d-2)s^2\,\overline{weight}(\mathcal{C}[123]) - s^2\,\overline{weight}(\mathcal{C}[111]) - s\,\overline{weight}(\mathcal{C}[111]) \end{cases},$$

and

$$\overline{weight}(\mathcal{C}) = d(d-1)(d-2)\,\overline{weight}(\mathcal{C}[321]) + d\,\overline{weight}(\mathcal{C}[111]).$$

Finally, we could obtain

$$f(s) = \frac{1}{1-ds-\overline{weight}(\mathcal{C})} = \frac{(-d^2+2d)s^3 + (-d^2+2d-1)s^2 + (1-d)s - 1}{(d-1)s^2+s-1}$$

4 Finite Memory Self-Avoiding Walks

The set of s-called self-avoiding walks could be regarded as a the set of words in the alphabet $V = \{1,-1,2,-2,\cdots\}$, that avoid, as factors, words with as many i's as $-i$'s for each i between 1 and d. In other words, words avoiding the 'bad words' in $B = \{[1,-1],[1,2,-1,-2]\cdots\}$ and all their images under the action of group of signed permutations. J. Noonan ([3]) has a detailed discussion about the finite memory self-avoiding walks for the memory up to 8. We have implemented the procedures for symmetric cases under signed permutations too. Using our maple package we could automatically get the formula for the generating functions for 2-step, 4-step and 6-step memory self-avoiding walks. For 8-step memory self-avoiding walk, the package set up a system of 112 linear equations but our own computer was not big enough to solve it.

5. The Maple package

All the procedures are included in the package "SYMBOLIC_GJ", downloadable from the web address: http://www.math.temple.edu/~wen/gj/gj/SYMBOLIC_GJ . The program takes the cardinality of the alphabet as symbolic input. Moreover it could compute generating functions for the symmetric cases and for the finite memory self avoiding walks.

6. Acknowledgment

This work will be a part of the author's Ph.D. dissertation, written under the direction of Professor Doron Zeilberger (Rutgers University). I wish to thank Dr. Zeilberger for his generous support and encouragement.

References

[1] *[Brinkhuis] J.Brinkhuis, "Non-repetitive sequences on three symbols",* Quart. J. Math*, Oxford(2), 145-149, 34(1983).*

[2] *[Zeilberger] John Noonan and Doron Zeilberger, " The Goulden-Jackson Cluster Method: Extensions, Applications, and Implementations",* J. Difference Eq. Appl.*, 355-377. 5(1999),*

[3] *[Noonan] John Noonan, " New Upper Bounds for The Connective Constants of Self-Avoiding Walks",* J. Stat. Phys.*, 871-888. 91(1998),*

[4] *[CoGuy] J.H. Conway and R.K. Guy, "The book of numbers",* Copernicus, Springer, New York. *(1996).*

[5] *[MS] N. Madras,and G. Slade, "The Self avoiding Walk", Birkhauser, Boston (1993).*

[6] *[GoJ1] I. Goulden and D.M. Jackson,* An inversion theorem for cluster decompositions of sequences with distinguished subsequences*, J. London Math. Soc.(2)***20** *(1979), 567-576.*

[7] *[GoJ2] I. Goulden and D.M. Jackson,* "Combinatorial Enumeration"*, John Wiley, 1983, New York.*

Department of Mathematics, Temple University, Philadelphia, PA 19122. wen@math.temple.edu