J Math Chem (2014) 52:355–367
DOI 10.1007/s10910-013-0267-z

ORIGINAL PAPER

Growing self avoiding walk trees

A. S. Padmanabhan

Received: 7 July 2013 / Accepted: 28 September 2013 / Published online: 11 October 2013

Abstract The Growing self avoiding walk model (GSAW) was proposed to explain statistical mechanics of the growth process in polymerization. Close examination of the model reveals that it is suited only for addition polymerization where only one monomer unit is added to the growing polymer chain. In this paper we propose a model for step growth or condensation polymerisation, where all of the monomer is converted first to dimer, all dimers are converted to tetramer etc. In the calculation of probabilities of the walker taking a step in a specified direction in the GSAW model the probability is the reciprocal of the number of steps available if the walker looks one step ahead. It is seen that our model for condensation polymerization GSAW can be generalized to the walker looking a finite number of steps ahead (instead of one only) for the purpose of calculation of probabilities. This is explained in detail in the first section of the paper. In the second section of the paper we explain the use of the Depth first search (DFS) algorithm in the calculation of probabilities and moments for the generalized growing self avoiding walk model. In the third section of the paper we report the exact values of the mean square end to end distances and relevant survival probabilities for the model. When GSAW was first proposed there was a controversy over whether it belonged to the same universality class as the True self avoiding walk $\left(\upsilon = \frac{2}{d+1}\right)$ or that of the self avoiding walk $\left(\upsilon = \frac{3}{d+2}\right)$. At that time it was conclusively shown that the GSAW model cannot belong to the same universality class as the true self avoiding walk model. However it was never conclusively shown that $\upsilon = \frac{3}{d+2}$ for GSAW. In this paper we propose two methods of studying this problem. One method

A. S. Padmanabhan (✉)
School of Chemical Sciences and Centre for High Performance Computing,
Mahatma Gandhi University, Priyadarshini Hills P.O., Kotayam 686560,
Kerala, India
e-mail: as.padmanabhan@gmail.com

is rigorous analysis of the DFS algorithm. We explain how this algorithm can be used to study restricted random walks with finite memory and self avoiding walks. The second method proposed by us is a detailed analysis of the generalised GSAW model proposed by us. This paper is to be viewed as an introductory paper on a new model that should be of interest to both chemists and mathematicians.

Keywords Growing self avoiding walk · Depth first search trees · Condensation polymerization · Growth process in polymerization

1 Introduction

The growing self avoiding walk model (GSAW) was first proposed as a modification of the True self avoiding walk [1–3]. The modification was as follows: in the True self avoiding walk if the walker is trapped he is allowed to select a previously visited neighbouring site at random, whereas this is disallowed in the Growing self avoiding walk. The model was first proposed by Kremer and Lyklema [4–7] and Majid et al. [8]. When the model was first proposed there was a controversy over whether it belonged to the same universality class as the True self avoiding walk $\left(\upsilon = \frac{2}{d+1}\right)$ or that of the self avoiding walk $\left(\upsilon = \frac{3}{d+2}\right)$ [9]. At that time it was conclusively shown that the GSAW model cannot belong to the same universality class as the true self avoiding walk model. Further studies were carried out on other lattices as well [10–13].

Close examination of the GSAW reveals that it is a model that is ideally suited for addition or chain growth polymerization. If we think of a polymer molecule as a necklace of beads (monomers), then the GSAW model could be thought of as stringing a necklace together bead by bead (single) from an infinitely large collection of beads. On the other hand if one were to construct a similar model for condensation polymerization we could think of all the beads in the collection forming dimer (until no single beads exist). Subsequently all the dimers form tetramers, all the tetramers form octamers etc. This idea is illustrated in Fig. 1.

In Fig. 1 the arrow marked in red refers to the third step of walk. Case 1 refers to GSAW [4–7] where one monomer unit is being added at a time. This implies the third step has three possible vacant sites to visit and so the probability of a site being selected is 1/3. Case 2 refers to a condensation reaction wherein a dimer reacts with a dimer. We assume that one of the dimers is a+X+Y as shown on the left side of the figure. Now for the second dimer there are 9 possibilities. All the possibilities are shown in the right side of the Figure. However it is seen that only 8 of these are allowed, the last one being disallowed as it results in a self intersection in the fourth step. As seen if the third step (marked in red) is to take place in the +X direction, then all three possibilities are allowed and the probability of the third step in the +X direction is 3/8. Similarly for the third step in the +Y direction the probability is 3/8. However it is clear from Fig. 1 that the probability of the third step in −X direction is only 2/8 = 1/4. It is hence seen that the probability of the walker taking the third step in a specified direction is different in the models for addition and condensation polymerization. In the model for condensation polymerization the walker has to look two steps ahead

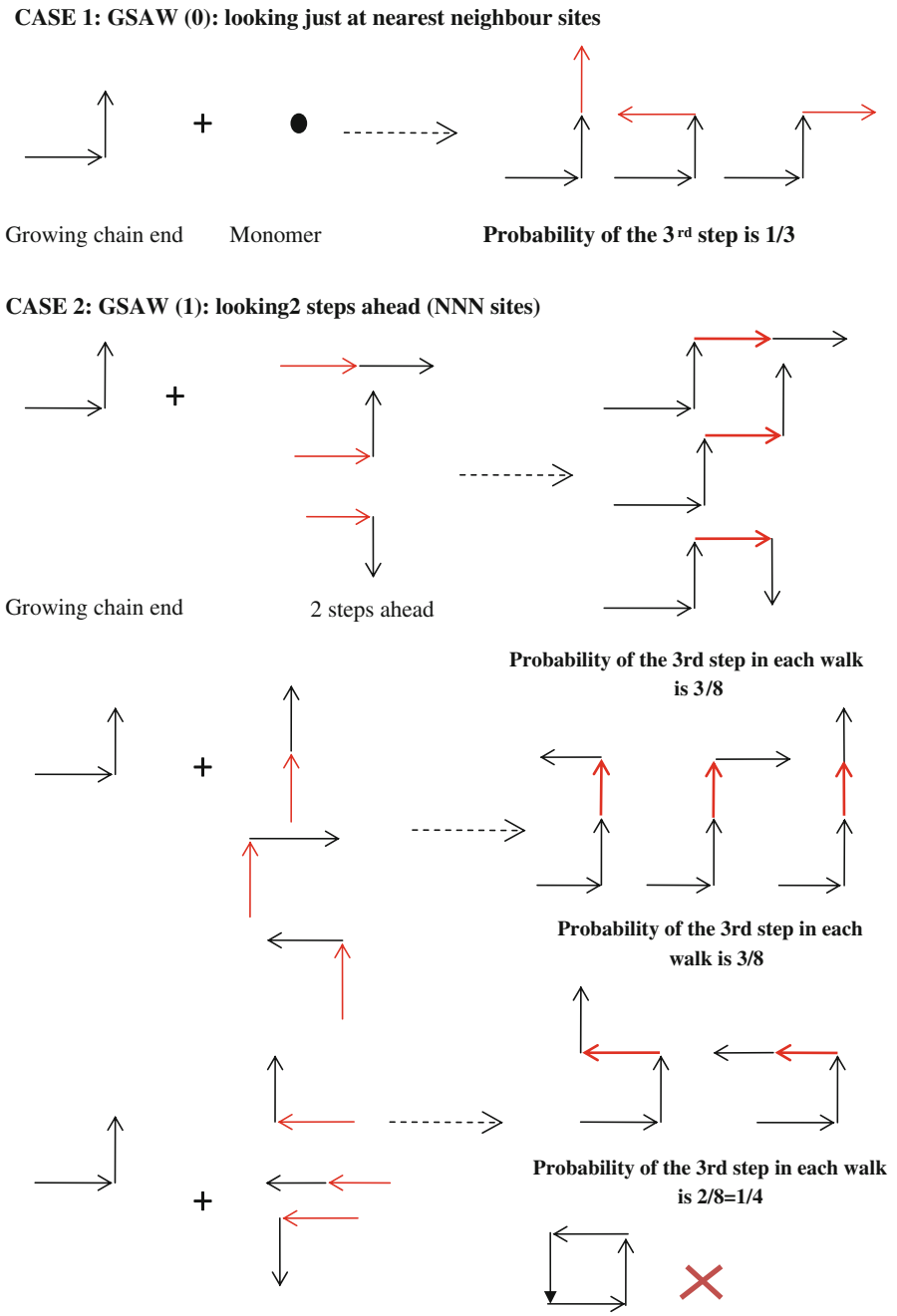


Fig. 1 Calculation of probabilities for GSAW (0) and GSAW (1)

and enumerate the number of vacant sites. The probability of him taking a step in a particular direction then depends on ratio of the number of vacant sites after that step has been taken to the total number of vacant sites after two steps have been taken. This idea can then be generalized to the walker taking any finite number of steps ahead and it forms the basis for our model for condensation polymerization. In Fig. 2 we illustrate the case of a four step walk in two dimensions, where the walker looks 1 and 2 steps ahead. We use the notation $GSAW(m-1)$ for the walker looking m steps ahead. Here we illustrate the case of the walker taking a step only in the +X direction. The sum of the probabilities is 1/4. Symmetry dictated the sum of the probabilities is the same if the first step is taken along the +Y, −X, and −Y directions. On the left column of the figure we display 4 step self avoiding walk as a figure and on the right side we report the probabilities of each step and the probability of the walk as a whole. For example in the first figure of *GSAW* (0) the probability of the first step is 1/4, the second step is 1/3, the third step is 1/3 and the final step is 1/2 and the total probability for the walk is computed as 1/72. Subsequently we provide a general definition for the probabilities based on these ideas.

Exact enumeration of the expectation of the square of the end to end distance and the survival probabilities calculated using the Depth first search (DFS) algorithm in two dimensions ($m = 1$ to 10) and three dimensions ($m = 1$ to 8) are reported in the third section of this paper.

1.1 Definition of probabilities for GSAW $(m-1)$ walks

Let S_{n+m} be the set of self avoiding walks of $n+m$ steps starting at the origin [14]. Let $P_n = (p(0), \ldots, p(n))$ be the sites of an n step self avoiding walk starting at the origin. Let $F_{n+m}(P_n)$ denote the subset of walks in S_{n+m} for which $P_n = (p(0), \ldots, p(n))$ are the first $n+1$ sites and $|F_{n+m}(P_n)|$ is the cardinality of this subset. Let $P'_{n+1} = (p(0), \ldots, p(n), p(n+1))$ be the sites of an $n+1$ step self avoiding walk starting at the origin whose first n sites are $p(0), \ldots, p(n)$.

We now define the probability of the $n+1$ step being along sites p_n and p_{n+1} in GSAW $(m-1)$ as:

$$pr(n+1) = \frac{\left|F_{n+m}(P'_{n+1})\right|}{|F_{n+m}(P_n)|}, |F_{n+m}(P_n)| > 0. \tag{1}$$

$pr(n+1)$ is a conditional probability. It is the probability of the n+1th step being along sites p_n and p_{n+1} given that the walk has survived up to n steps. The probability of the walk $P'_{n+1} = (p(0), \ldots, p(n), p(n+1))$ in GSAW $(m-1)$ is given by:

$$Prob\left(P'_{n+1)}\right) = \prod_{i=1}^{n+1} pr(i) \text{ where } pr(i) \text{ is given by (1).}$$

Note: $|F_{n+mI}(P'_{n+1})| = 0$ for GSAW (0) if P'_{n+1} is a trap. Similarly if P_{n+1} leads into a trap and say the walker is trapped after M steps then $|F_{n+mI}(P'_{n+1})|$ is not equal to 0 for $m < M$ but $|F_{n+mI}(P'_{n+1})| = 0$ for m greater than or equal to M.

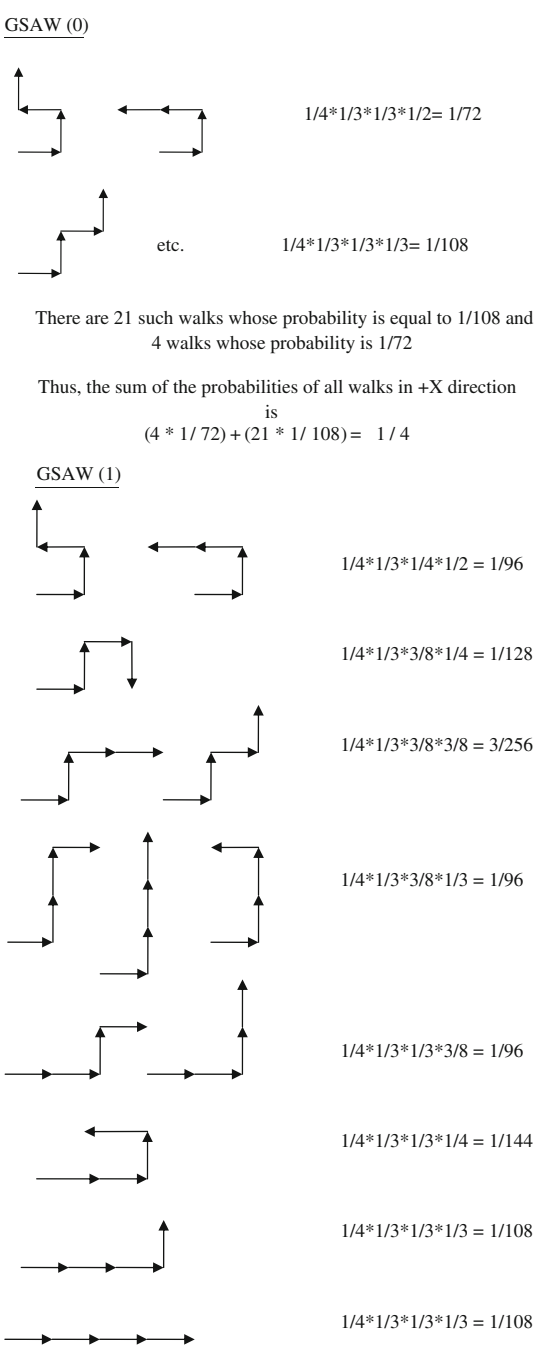


Fig. 2 Method of calculation of one step probabilities for GSAW (0) and GSAW (1) for a four step walk in two dimensions. The first step is in the +X direction. Since there is a symmetry factor of 4 for each such walk, the sum of the probabilities is 1/4. Likewise there are +Y, −X and −Y directions and so the total probability is 1

2 Depth first search trees

The DFS method in combination with the pivot algorithm has been used to estimate the connective constant of self avoiding walks [15–17] The author along with a fellow worker has recently used the DFS tree to derive the generating function for a memory four walk [18]. The method used is to trace the search and keep track of the vertices in the DFS tree where self intersection occurs. The vertex in the tree at which this occurs is thereby deleted along with all its descendants. This information in then used to derive recursion formulae using the inclusion exclusion criteria followed by the search along the DFS tree. The algorithm used by us has been reported in Ref. [17] and is being repeated here verbatim for the sake of continuity in discussion.

We first define the terms used in the algorithm (Graph theory terms commonly used such as vertex, level of a vertex, first visit to a vertex, last visit to a vertex are used without definition and are the same as in Bondy and Murthy [19]):

If $-\varepsilon_1, \varepsilon_1, -\varepsilon_2, \varepsilon_2, \ldots, -\varepsilon_d, \varepsilon_d$ are the $2d$ unit vectors in d dimensional Euclidean space.

$\boldsymbol{F}$ refers to the set $(-\varepsilon_1, \varepsilon_1, -\varepsilon_2, \varepsilon_2, \ldots, -\varepsilon_d, \varepsilon_d)$, accessible in that order. $l\,(v)$ refers to the level of the vertex $\boldsymbol{v}$., $t\,(v)$ refers to the time of visit to a vertex $\boldsymbol{v}$, *first* ($\boldsymbol{v}$) refers to the time of the first visit to $\boldsymbol{v}$, *last*($\boldsymbol{v}$) refers to the time of the second (last) visit to $\boldsymbol{v}$.

C refers to the family of n step self avoiding walks. This might refer to the co ordinates of the entire walk, or the moments of the end to end distance, radius of gyration or any such relevant parameter.

St refers to the Stack used in the algorithm.

Set the vertex $\boldsymbol{v} := \boldsymbol{u}$ refers to the co ordinates of the self avoiding walk resulting from appending the vector $\boldsymbol{u}$ to the self avoiding walk at the previous level.

Algorithm 1

INPUT: Number of steps *n*, dimension *d*, Family of unit vectors in *d* dimension ***F***.

OUTPUT: Family of self avoiding walks of *n* steps $\mathbf{C}$, $\sum_{i=0}^{n} C_i$, time of visit of each vertex of the DFS tree, time of first visit to each vertex of the DFS tree, time of second (last) visit to each vertex of the DFS tree.

1: Set $i := 0$ and $St := \varphi$.

2: Set the origin as the root of the tree **r** and set **v**:=**r**.

3: increment *i* by 1.

4: Set $t(\boldsymbol{v}) := 1$, *first*(*v*) := *t*(***v***), and $l(\boldsymbol{v}) := 0$.

5: Place ***v*** at the top of the stack *St* .

6: **for** *j*:=1 **to** *j*:=*N* **do** set *E*(*j*) := ***F***

7: **while** *St* is non empty **do**

8: **if** *l*(***v***) <= *n* **do**

9: set k:=1

10: **if** *E*(*k*) is non empty **do**

```
11:                Consider the head u of E(k) and set v:=u
12:              Place v at the top of the stack St and label the resulting walk as x
13:                    if x is self avoiding do
14:                    remove u from E(k)
15:                    set l(v) := k
16:                    increment k  by 1 and i by 1.
17:                    set t(v) := i , and first(v) := t(v)
18:                    end if
19:                    else remove u from E(k)
20:               end if
21:               else
22:                set E(k) := F
23:               remove v from the top of the stack St
24:               decrement k by 1 and increment i by 1
25:               set t(v) := i and last(v) := t(v)
26:               end else
27:         end if
28:         else
29:         increment i by 1
30:         set t(v) := i and last(v) := t(v)
31:          append x to C
32:         remove v from the top of the stack St
33:         decrement k by 1
34 end else
35:   end while
36: return (C,t,first,last)
```

2.1 Structure of the DFS tree for self avoiding walks of four steps in two dimensions

In the Figure given below we display a branch of DFS tree for 4 step self avoiding walks in two dimensions. The first number in the box represents the chronological order in which a vertex appears, 1 represents (0,0), 2 represents (−1,0), 3 represents (−2,0) etc. The second number represents the time of first visit to a vertex and the third number the time of last (second) visit to a vertex. To explain further the first walk to be stored is [(0,0), (−1,0), (−2,0), (−3,0), (−4,0)] and this is given by the left most branch:

1 1 306, 2 2 77, 3 3 28, 4 4 11, 5 5 6

The second walk to be stored is [(0,0),(−1,0),(−2,0),(−3,0),(−3,−1)] and this is given by the branch:

1 1 306, 2 2 77, 3 3 28, 4 4 11, 6 7 8

3 Exact enumeration

In Table 1 we report the exact enumeration values of the conditional expectation of the square of the end to end distance of n step walks, given that the walk has survived up to that many steps in two dimension for $GSAW(m-1)$ for m from 1 to 10. The second number reported is $\sum_{\Omega} p_{\omega\varepsilon\Omega}$ and is equal to the sum of the probabilities of walks that survive up to the given number steps, reported up to 3 decimal places. This number is usually referred to as the partition function in the physics and chemistry literature. For example second number 0.997 in the GSAW (0) column for 8 steps means that the sum of the probabilities that survive up to 8 steps is 0.997 (some walks are trapped after 7 steps). When no second number is reported the sum of the probabilities is greater than 0.999. The first column refers to the Uniform distribution i.e. the probability of a walk of n steps is the reciprocal of the number of such walks. Values reported in all cases (including the first column) are calculated using the DFS algorithm. Values reported in the first column have been reported by other authors (for a much larger number of steps) [14]. Values in the second column have also been reported by other authors before [4–6].

In Table 2 we report the exact enumeration values of the conditional expectation of the square of the end to end distance of n step walks, given that the walk has survived up to that many steps, in three dimension for $GSAW(m-1)$ for m from 1 to 8. No second number is reported as the sum of the probabilities (partition function in the physics and chemistry literature) is greater than 0.999 in all cases. The first column refers to the Uniform distribution i.e. the probability of a walk of n steps is the reciprocal of the number of such walks. Values reported in all cases are calculated using the DFS algorithm. Values reported in the first column have been reported by other authors (for a much larger number of steps) [14]. Values in the second column have also been reported by other authors before [4–6] (Fig. 3).

4 Discussion

Examination of Fig. 3 will help in understanding the inclusion exclusion followed in the DFS tree. This then can be stated formally as a theorem for restricted random walks of finite memory with the uniform distribution.

First we explain some notation that is used:

All standard definition of Self avoiding walks apply [19].

Let C_n^{τ} be the number of memory τ restricted random walks of n steps in d dimension.

$For n > 2\tau, 2 < \lambda \leq \tau, let\ P_n^{\tau}(\lambda)$ be the number of restricted random walks of n steps in d dimension whose first $n-1$ steps obey the memory τ condition, and whose nth step results in a λ step reversal. In other words $P_n^{\tau}(\lambda)$ is the number of

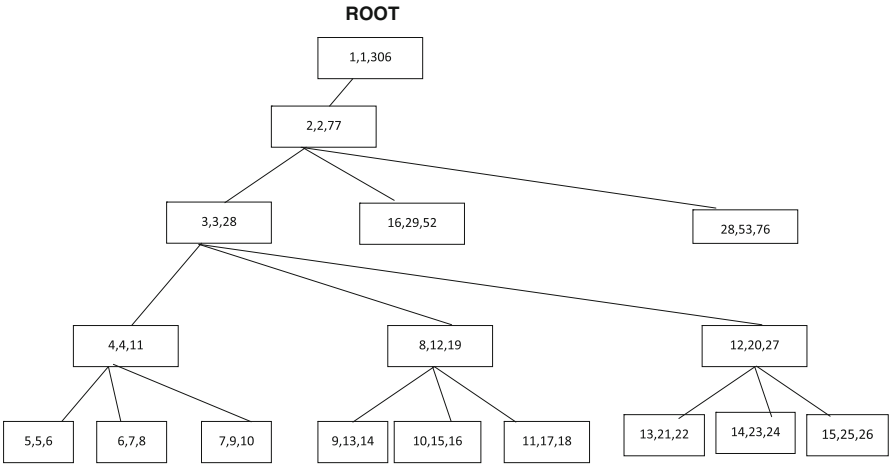
Table 1 Mean square end to end distance for GSAW ($m-1$) in two dimension

No. of steps	Conditional expectation of the square end to end distance the walk has survived in 2d of GSAW ($m-1$) and Uniform distribution										
	UD	GSAW (0)	GSAW (1)	GSAW (2)	GSAW (3)	GSAW (4)	GSAW (5)	GSAW (6)	GSAW (7)	GSAW (8)	GSAW (9)
4	7.040	6.741	7.215	7.395	7.399	7.393	7.394	7.382	7.381	7.371	7.369
5	9.563	9.000	9.904	10.185	10.253	10.242	10.265	10.241	10.247	10.226	10.228
6	12.574	11.428	12.760	13.233	13.366	13.389	13.430	13.409	13.420	13.395	13.396
7	15.556	13.872	15.768	16.440	16.700	16.753	16.844	16.820	16.850	16.816	16.824
8	19.013	16.481 *0.997*	18.880	19.821	20.218	20.343	20.483	20.478	20.525	20.493	20.507
9	22.411	19.093 *0.995*	22.091	23.312	23.900	24.099	24.323	24.333	24.418	24.385	24.415
10	26.243	21.845 *0.990*	25.383	26.925	27.717	28.028	28.340	28.389	28.512	28.491	28.537
11	30.018	24.606 *0.986*	28.745 *0.999*	30.619	31.658	32.088	32.519	32.609	32.789	32.780	32.854
12	34.187	27.491 *0.979*	32.177 *0.998*	34.406	35.702	36.283	36.839	36.992	37.233	37.250	37.353
13	38.304	30.380 *0.973*	35.668 *0.997*	38.254	39.843	40.582	41.292	41.511	41.833	41.877	42.023
14	42.786	33.390 *0.966*	39.219 *0.995*	42.180 *0.999*	44.064	44.989	45.860	46.166	46.573	46.659	46.849
15	47.218	36.395 *0.958*	42.826 *0.993*	46.154 *0.999*	48.363	49.481	50.537	50.935	51.446	51.576	51.825
16	51.993	39.522 *0.949*	46.484 *0.991*	50.195 *0.999*	52.729	54.060	55.310	55.818	56.438	56.627	56.937
17	56.716	42.638 *0.941*	50.196 *0.988*	54.278 *0.998*	57.158	58.709	60.173	60.798	61.544		
18	61.766	45.872 *0.931*	53.954 *0.985*	58.422 *0.997*	61.643 *0.999*	63.432	65.116	65.875			
19	66.766	49.092 *0.922*	57.763 *0.982*	62.601 *0.996*	66.184 *0.999*	68.213	70.137				
20	72.077	52.430 *0.912*	61.614 *0.978*	66.838 *0.995*	70.773 *0.999*	73.056					
21	77.337	55.748 *0.902*	65.517 *0.975*	71.106 *0.994*	75.412 *0.999*						
22	82.896	59.183 *0.891*	69.457 *0.971*	75.429 *0.992*							
23	88.404	62.596 *0.881*	73.447 *0.966*								
24	94.201	66.125 *0.870*									
25	99.947	69.628 *0.859*									

The number in italics represents the probability that the walk has survived up to the given number of steps

Table 2 Mean square end to end distance for GSAW ($m-1$) in three dimension

No. of steps	Conditional expectation of the square end to end distance given that the walk has survived in 3d of GSAW ($m-1$) and Uniform distribution (UD)								
	UD	GSAW (0)	GSAW (1)	GSAW (2)	GSAW (3)	GSAW (4)	GSAW (5)	GSAW (6)	GSAW (7)
4	5.554	5.424	5.621	5.675	5.695	5.708	5.710	5.711	5.712
5	7.234	6.968	7.347	7.438	7.486	7.505	7.515	7.517	7.520
6	9.071	8.546	9.120	9.273	9.348	9.384	9.401	9.408	9.414
7	10.897	10.128	10.930	11.144	11.261	11.314	11.346	11.357	11.369
8	12.845	11.737	12.767	13.058	13.217	13.297	13.343	13.363	13.382
9	14.781	13.353	14.627	14.996	15.209	15.314	15.382	15.412	15.441
10	16.817	14.990	16.505	16.964	17.230	17.370	17.458	17.502	17.541
11	18.841	16.632	18.399	18.948	19.277	19.450	19.566	19.624	
12	20.953	18.292	20.307	20.954	21.345	21.559	21.702		
13	23.052	19.955	22.228	22.972	23.433	23.688			
14	25.228	21.631	24.158	25.007	25.538				
15	27.393	23.309	26.099	27.052					
16	29.627	24.998	28.047						
17	31.849	26.689							
18	34.135								



Note: 1. 1st Self intersection occurs after	23,41,42
2. 2nd Self intersection occurs after	34,63,64
3. 3rd Self intersection occurs after	57,109,110
4. 4th Self intersection occurs after	68,131,132
5. 5th Self intersection occurs after	89,173,174
6. 6th Self intersection occurs after	100,196,201
7. 7th Self intersection occurs after	123,241,242
8. 8th Self intersection occurs after	134,264,269

Fig. 3 Branch of DFS tree for 4 step SAW in 2D

$n$step restricted random walks of memory τ but whose head has a λ step self avoiding polygon.

$For\ n > 2\tau, 2 < \beta \leq \tau, let\, Q_n^{\tau}(\beta, \lambda)$ be the number of restricted random walks of n steps in d dimension whose first $\beta - 1$ steps obey the memory τ condition, whose βth step results in a β step reversal, whose next $n - 1 - \beta$ steps obey the τ memory condition, but whose last step results in a λ step reversal. In other words $Q_n^{\tau}(\beta, \lambda)$ is the number of $n$step restricted random walks of memory τ but whose tail has a β step self avoiding polygon and whose head has a λ step self avoiding polygon.

Then the following theorem holds:

Theorem 4.1

$$C_n^{\tau} = [2d-1]\left[C_{n-1}^{\tau} - \sum_{k=2}^{\frac{\tau}{2}} P_{n-1}^{\tau}(2k)\right] + \sum_{k=2}^{\frac{\tau}{2}}\sum_{j=2}^{\frac{\tau}{2}} Q_n^{\tau}(2k, 2j) \text{ for all } n \geq 2\tau.$$

Proof First we note that:

$$C_n^{\tau} = [2d-1]\, C_{n-1}^{\tau} - \sum_{k=2}^{\frac{\tau}{2}} P_n^{\tau}(2k)$$

Next using the reasoning on the comparison of $n-1$ and n step DFS trees we find that:

$$P_n^{\tau}(2k) = [2d-1]\left[P_{n-1}^{\tau}(2k)\right] + \sum_{j=2}^{\frac{\tau}{2}} Q_n^{\tau}(2k, 2j)$$

□

This theorem has been used to evaluate $\chi^4(z) = \sum_{n=0}^{\infty} C_n^4 z^n$ [17] and work is in progress to evaluate this function for restricted random walks of higher memory. The method to be followed is to express $Q_n^{\tau}(\beta, \lambda)$ in terms of $P_i^{\tau}(\lambda)$ $for\, i < n$ and $P_i^{\tau}(\lambda)$ in terms of C_l^{τ} $for\, l < i$.

Rigorous mathematical analysis of $GSAW(m-1)$ walks is different from the analysis of self avoiding walks with uniform distribution in one very important respect, viz., the probabilities and expectation of the moments of the end to end distance are conditional probabilities and expectations in $GSAW(m-1)$, the condition being that the walk survives attrition due to traps up to the given number of steps. In other words the conditional probability assigned to a walk of n steps given that the walk has survived up to n steps is equal to $\frac{p_\omega}{\sum_{\omega\varepsilon\Omega} p_\omega}$ and the conditional expectation of the end to end distance is given by:

$$\frac{\sum_{\Omega} r_{\omega\varepsilon\Omega}^2 p_{\omega\varepsilon\Omega}}{\sum_{\Omega} p_{\omega\varepsilon\Omega}}$$

This quantity [4–9] along with $\frac{\sum_{\Omega} r_{\omega\varepsilon\Omega}^4 p_{\omega\varepsilon\Omega}}{\sum_{\Omega} p_{\omega\varepsilon\Omega}}$ and $\frac{\sum_{\Omega} r_{\omega\varepsilon\Omega}^6 p_{\omega\varepsilon\Omega}}{\sum_{\Omega} p_{\omega\varepsilon\Omega}}$ have been studied [5] by earlier authors.

In an earlier work the author and co worker had explained an Algorithm based on the Breadth first search tree [18] (BFS) and showed that the BFS and DFS trees are isomorphic. Both trees contain and enormous amount of information about SAW and GSAW $(m-1)$. The number of vertices at each level of the tree gives the number of self avoiding walks for the number of steps equal to the level. Also to borrow terms from genealogy (family tree) the probability of a vertex in the tree being selected in

$GSAW(0)$ is equal to the reciprocal of the number of siblings the vertex has plus one (for the vertex itself). In GSAW (1) it is equal to the ratio of the number of children the vertex has to the total number of children he and his siblings have and in GSAW (2) it is equal to the ratio of the number of grand children the vertex has to the total number of grand children he and his siblings have, and so on. So on a single run of a computer program that uses Algorithm 1 all the data in Tables 1 or 2 can be obtained. This explains the reason for deriving Theorem 4.1 in an introductory paper on a model for condensation polymerization.

5 Conclusion

In this paper we:

(1) Introduced a new model for condensation polymerization and showed how this can incorporated into a general model for polymerization based on a model for addition polymerization developed many years ago [4–9].
(2) Showed how models for equilibrium and non equilibrium properties of polymers in good solvent can be studied rigorously using the Breadth first search and DFS techniques commonly used in computer science.

Acknowledgments The author would like to gratefully acknowledge the very useful and relevant suggestions to the advisors of Journal of Mathematical Chemistry who reviewed this manuscript.

References

1. D.J. Amit, G. Pairisi, L. Peliti, Phys. Rev. B. **27**, 1635–1645 (1983)
2. L. Pietronero, Phys. Rev. B. **27**, 5887–5889 (1983)
3. F. Family, M. Daoud, Phys. Rev. B. **29**, 1506–1507 (1984)
4. J.W. Lyklema, K. Kremer, J. Phys. A. Math. Gen. **17**, L691–L696 (1984)
5. K. Kremer, J.W.J. Lyklema, Phys. A. Math. Gen. **18**, 1515–1531 (1985)
6. J.W. Lyklema, K. Kremer, J. Phys. A. Math. Gen. **19**, 279–289 (1986)
7. K. Kremer, J.W. Lyklema, Phys. Rev. Lett. **54**(9), 267–269 (1985)
8. I. Majid, N. Jan, A. Coniglio, H.E. Stanley, Phys. Rev. Lett. **52**(15), 1257–1260 (1984)
9. K. Kremer, J.W. Lyklema, Phys. Rev. Lett. **55**(19), 2091–2092 1985 (this is a comment on Ref. 8 and the response by the authors of Ref. 8)
10. A. Weinrib, S.A. Trugman, Phys. Rev. B. **31**(5), 2993–2997 (1984)
11. P. Meakin, Phys. Rev. A. **37**(7), 2644–2659 (1988)
12. S.S. Manna, A.J. Guttmann, J. Phys. A. Math. Gen. **22**, 3113–3122 (1989)
13. M.R. Bradley, P.N. Strenski, J.M. Debierrie, Phys. Rev. A. **45**(12), 8513–8524 (1992)
14. N. Madras, G. Slade, *The Self Avoiding Walk* (Birkhauser, Boston, 1993)
15. N.J. Clisby, Stat. Phys. **140**(3), 49–92 (2010)
16. N. Clisby, Phys. Rev. Lett. **104**, 055702 (2010)
17. N.J. Clisby, Phys. A. Math. Theory **40**, 245001 (2013)
18. A.S. Padmanabhan, S.J. Jacob, Math. Chem. (submitted 2013)
19. J.A. Bondy, U.S.R. Murthy, *Graph Theory, (Graduate Texts in Mathematics)* (Springer, Berlin, 2008)