Exploiting Model Checking in Constraint-based Approaches to the Protein Folding Problem *

Elisabetta De Maria, Agostino Dovier, Angelo Montanari, and Carla Piazza

Dipartimento di Matematica e Informatica, Università di Udine
via delle Scienze 206, 33100 Udine, Italy
{demaria,dovier,montana,piazza}@dimi.uniud.it

Abstract. In this paper we show how model checking can be used to drive the solution search in the protein folding problem encoded as a constraint optimization problem. The application of the model checking technique allows us to distinguish between meaningful protein conformations and bad ones. This classification of conformations can then be exploited by constraint solvers to significatively prune the search space of the protein folding problem. Furthermore, our approach seems promising in the study of folding/energy landscapes of proteins.

1 Introduction

In this paper we show how model checking can be used to drive the solution search in the protein folding problem encoded as a constraint optimization problem. Given the molecular composition of a protein, i.e., a list of amino acids, known as its *primary structure*, the *protein structure prediction* (or *protein folding*) problem consists in determining the 3D shape (*tertiary structure* or *conformation*) that the protein assumes in normal conditions in biological environments [5].

To solve the protein folding problem it is crucial to determine the conformations of the amino acid sequences in the 3D space with minimum energy. It is indeed widely accepted that a state with minimum energy represents the protein's natural shape (a.k.a. the *native conformation*). The energy of a conformation can be modeled by means of suitable *energy functions*, which express the energy level in terms of the interactions between pairs of amino acids [3]. Since the protein folding problem is extremely complex, it is often simplified in several respects. A common simplification consists in using *lattice space models* to restrict the admissible positions of the amino acids in the space [11]. The energy function can be simplified as well, e.g., by adopting the 20×20 potential matrix proposed by [7, 8] or the simpler HP model [1, 2]. For the sake of simplicity, in the following we assume a 2D finite lattice included in $\mathbb{N}^2$ and the HP energy model. However, our approach can be easily extended to 3D lattices. Furthermore, it is not difficult to replace the HP model with a more refined

* This work has been partially supported by PRIN 2005 project 2005015491 and by FIRB 2003 project RBNE03B8KK.

energy model that keeps track of the variety of interactions among the 20 kinds of amino acids.

In this work we show how model checking techniques can be exploited to investigate the relationships among the different possible conformations of proteins. We model the solution space of the protein folding problem as a finite transition system whose states are all the possible conformations of a protein and whose transitions represent admissible transformations of conformations. Then, we take advantage of temporal logic to specify and check relevant properties of such a system. As an example, we show how to check whether there exists a path from a given conformation to a conformation with an energy level below a certain threshold whose length is less than or equal to a given value. In particular, we are interested in identifying patterns common to different proteins. These patterns can be used to improve the solution search in existing constraint-based protein folding algorithms as well as to understand protein functions. In general, constraints allow one to easily model minimization problems. Once the constraint model is defined, a constraint solver can indeed be used to search for solutions. This search exploits the constraints to prune the solution space. In the following, we show how model checking can be used to identify meaningful properties of protein conformations that can be encoded as additional constraints to be used to further reduce the solution space.

The paper is organized as follows. In Section 2 we introduce the HP model. In Section 3 we describe how to generate, for any given protein, the corresponding finite transition system. In Section 4 we show how to express relevant properties of protein conformations in temporal logic. In Section 5 we report preliminary experimental results and we outline some ongoing developments of the work.

2 The HP model of proteins

The HP model on a 2D discrete lattice, where every conformation of a protein is a self-avoiding walk in $\mathbb{Z}^2$, is commonly used to represent the conformations and the energy function of proteins [12]. Such a model reduces the 20-letter alphabet of amino acids to a two-letter alphabet $\{H, P\}$, where H (resp., P) represents a hydrophobic (resp., polar) amino acid. The energy function states that the energy contribution of a *contact* between two amino acids is -1 if both of them are H amino acids, 0 otherwise.

Hereafter, we represent an HP sequence as an element in $\{0,1\}^*$, where 1 (resp., 0) stands for an H (resp., P) amino acid. Furthermore, for $i = 0, 1, \ldots, n$, we denote by s_i the i-th element of a sequence s of $n+1$ elements. The subset of admissible protein conformations is defined as follows.

Definition 1 (Folding). *A* folding ω *of a sequence* $s = s_0 \ldots s_n$ *is a function* $\omega : [0 \ldots n] \to \mathbb{Z}^2$ *such that*

(i) $\forall 0 \leq i < n$ $(|\omega(i) - \omega(i+1)| = 1)$*, that is, if* $\omega(i) = (X_i, Y_i)$ *and* $\omega(i+1) = (X_{i+1}, Y_{i+1})$*, then* $|X_i - X_{i+1}| + |Y_i - Y_{i+1}| = 1$;
(ii) $\forall i \neq j (\omega(i) \neq \omega(j))$ *(ω is self avoiding).*

We say that two amino acids s_i and s_j of a given folding ω are *connected neighbors* if $j = i \pm 1$ and that they are *topological neighbors* if they are not connected and $|\omega(i) - \omega(j)| = 1$.
In the HP model, the energy of a folding is given by the opposite of the number of topological HH neighbors, e.g., if there exist k topological HH neighbors in ω, then the energy of ω is $-k$.

Definition 2 (Folding Energy). *Given a sequence $s = s_0 \dots s_n$, the energy of a folding of s is:*

$$E = \sum_{1 \leq i+1 < j \leq n} B_{i,j} \cdot \delta(s_i, s_j)$$

where $B_{i,j}$ is equal to -1 whenever both s_i and s_j are H amino acids, 0 otherwise, and $\delta(s_i, s_j)$ is 1 if s_i and s_j are topological neighbors, 0 otherwise.

Hence, a folding has minimum energy if it maximizes the number of HH contacts.
Given a sequence $s = s_0 \dots s_n$, we assume its length to be n, i.e., it is equal to the number of "segments" it is made of. To represent the conformations of a sequence of length n, we use the subset $\mathcal{L} = \{(i, j) : i \in [0, 2n], j \in [0, 2n]\}$ of $\mathbb{N}^2$.
Without loss of generality, we assume $\omega(0) = (n, n)$ and, in order to avoid simple symmetries, we fix $\omega(1) = (n, n + 1)$.
Notice that, once the coordinates of a segment have been fixed, the next segment in the sequence can only assume three possible directions with respect to the preceding one: left (l), forward (f), and right (r). As a result, a folding of a sequence of length n can be represented as a string of length $n - 1$ on the alphabet $\{l, f, r\}$ [1]. As an example, the sequence of Figure 1 is represented by the string *rllf*.

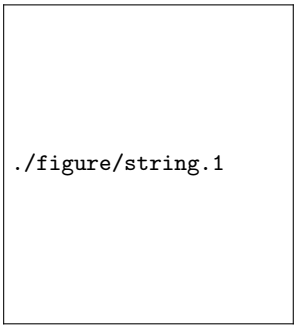

Fig. 1. String *rllf* on 10×10 lattice.

[1] To avoid symmetries it is possible to consider only strings with prefixes of the form f^*r.

The number of all possible foldings of a sequence of length n, where the orientation of the first segment is fixed as above, is bounded by 3^{n-1}. It is commonly accepted that the number C_n of self-avoiding walks of length n grows according to the following formula $C_n = B \cdot \mu^n \cdot n^{\gamma-1}$, where $B \sim 1.93$, $\mu \sim 2.63$, and $\gamma = 43/32$ [13], and thus the number of self-avoiding walks of length n, where the orientation of the first segment is fixed, is $D_n = C_n/4$. In [15] Ngo and Marks have shown that protein folding problem on 2D-lattices is NP-complete.

Now we formally define the set of valid transformations among foldings. Roughly speaking, a valid transformation of a given folding f consists in selecting at random a position in f and performing a rotation of the part of f between this position and the ending position (*pivot move*).

Definition 3 (Pivot move). *Let $f = f_2 \dots f_n$, with $f_i \in \{l, f, r\}$ for all $2 \leq i \leq n$, be a folding of a sequence s of length n. A folding f' of s is obtained from f through a* pivot move *with pivot $k-1$, with $2 \leq k \leq n$, if $f'_i = f_i$ for all $i \neq k$ and $f'_k \neq f_k$.*

Given a folding of a sequence of length n, since the number of possible pivots is $n-1$ and each one may give rise to two moves, i.e., rotations, the number of successor foldings is at most $2(n-1)$ (some of these conformations could violate the self avoiding condition). As an example, consider the sequence of length 4 whose folding is represented by the string *fff*. The foldings obtained by pivot moves are the 6 foldings *lfl*, *rfl*, *fll*, *frl*, *fff*, *ffr*. They are graphically depicted in Figure 2. It is possible to show that pivot moves are ergodic, namely, they cover the entire folding space [5].

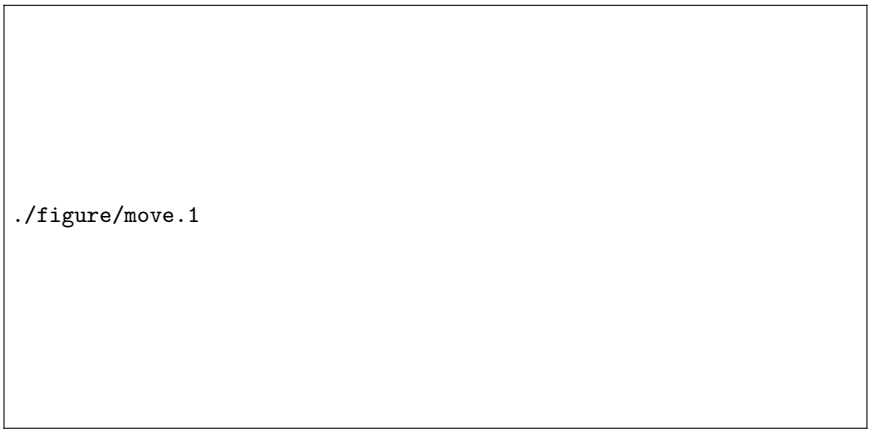

Fig. 2. Pivot moves from string *fff*.

3 Protein transition systems

In this section we propose an approach to the formal verification of interesting protein conformation properties based on *model checking* [4]. Model checking allows one to verify desirable properties of a system by an exhaustive enumeration of all the states reachable by the system. We model the set of protein foldings and their relationships as a finite transition system and we use (linear or branching) propositional temporal logic to specify relevant system properties [9, 17].

Definition 4 (Transition System). *Let AP be a set of atomic propositions. A* transition system *over AP is a tuple $M = (Q, T, L)$, where*

- *Q is a finite set of states;*
- *$T \subseteq Q \times Q$ is a total transition relation, that is, for every state $q \in Q$ there is a state $q' \in Q$ such that $T(q, q')$;*
- *$L : Q \to 2^{AP}$ is a labeling function that maps every state into the set of atomic propositions that hold at it.*

The 2D Protein Transition System is defined as follows:

Definition 5 (2D Protein Transition System). *The 2D Protein Transition System of a string P of length n over $\{0, 1\}$ is a tuple $M_P = (Q, T, L)$, where*

- *Q is the set of all foldings of length n on the $2n \times 2n$ 2D lattice;*
- *$T \subseteq Q \times Q$ contains the pairs of states (q_1, q_2) such that q_2 can be obtained from q_1 by a pivot move;*
- *$L : Q \to 2^{AP}$ is a labeling function over the set AP of atomic propositions which consists of the following 3(n-1) predicates*
$$2nd\_l, \ldots, nth\_l, \quad 2nd\_f, \ldots, nth\_f, \quad 2nd\_r, \ldots, nth\_r,$$
plus the following three predicates
$$min\_en, \quad inter\_en, \quad max\_en,$$
where for all $2 \leq i \leq n$, the predicate ith_l (resp., ith_f, ith_r) holds at a state q if the i-th segment of q has a left *(resp.,* forward, right*) orientation and min_en (resp., $inter\_en$, max_en) holds at a state q if the energy of q is minimum (resp., intermediate, 0).*

It is possible to prove that the 2D Protein Transition System corresponding to a given protein has the following properties.

Proposition 1 (Properties of the 2D Protein Transition System).

1. *It is* strongly connected*, i.e., for each pair of states q_1 and q_2, there is a path from q_1 to q_2.*
2. *It is* symmetric*, i.e., for each pair of states q_1 and q_2, if (q_1, q_2) belongs to T, then (q_2, q_1) belongs to T.*
3. *The* maximum incidence degree *D=$\max_{q \in Q} |\{(q, q') : (q, q') \in T\}|$ is 2(n-1).*

Item 1 of Proposition 1 holds since pivot moves are ergodic [5]. Item 2 of Proposition 1 holds because, if state q_2 can be obtained from state q_1 performing a pivot move, then q_1 can be obtained from q_2 performing the opposite move. Item 3 immediately follows from Definition 3.

As far as the energy of a protein is concerned, from our experimental results it turns out that the majority of states has a high energy and that only a few states have minimum energy. Furthermore, the value of the energy difference between the source and destination nodes of most edges is 0.

4 Model checking properties of proteins

Temporal logics are formalisms for describing sequences of transitions between states. We restrict our attention to two well-known fragments of the *computation tree logic* CTL*, namely, the *branching time* logic CTL and the *linear time* logic LTL [9]. CTL* formulae describe properties of computation trees and they are obtained by (repeatedly) applying Boolean connectives, *path quantifiers*, and *state quantifiers* to atomic formulae. The path quantifier **A** (resp., **E**) can be used to state that all paths (resp., some path) starting from a given state have some property. The state quantifiers are the next time operator **X**, which can be used to impose that a property holds at the next state of a path, the operator **F** (sometimes in the future), that requires that a property holds at some state on the path, the operator **G** (always in the future), that specifies that a property is true at every state on the path, and the until binary operator **U**, which holds if there is a state on the path where the second of its argument properties holds and, at every preceding state on the path, the first of its two argument properties holds.

CTL allows one to quantify over the paths starting from a given state. Unlike CTL*, it constrains every state quantifier to be immediately preceded by a path quantifier. In LTL one may only describe events along a single computation path. Its formulae are of the form $\mathbf{A}f$, where f does not contain path quantifiers, but it allows the nesting of state quantifiers. CTL and LTL have different expressive powers [9]. We chose to use both of them to benefit from their advantages. On the one hand, the complexity of model checking for CTL is linear in the number of states and edges of the transition system, while the model checking problem for LTL is PSPACE-complete. Furthermore, there are many tools for checking if finite state systems satisfy CTL formulae (see, e.g., SMV [14]). On the other hand, algorithms for on-the-fly model checking, a technique that allows one to contrast the state explosion problem trying not to build the entire transition system, mainly deals with LTL formulae. As a matter of fact, all the relevant properties of Protein Transition Systems we identified belong to the intersection of CTL and LTL.

Given a 2D Protein Transition System $M_P = (Q, T, L)$ and a temporal logic formula f expressing some desirable property of the system, the *model checking*

problem consists in finding the set of all states in Q satisfying f:

$$[\![f]\!] = \{q \in Q : M_P, q \models f\}.$$

When a state does not satisfy a formula, model checking algorithms produce a counterexample that falsifies it, thus providing an insight to understand failure causes and important clues for fixing the problem.

We conclude the section by showing how meaningful properties of 2D Protein Transition Systems can be encoded in both CTL and LTL.

F1: Does it exist a path of length at most k that reaches a state with minimum energy?
CTL:$min\_en \vee EXmin\_en \vee \cdots \vee \underbrace{EX \ldots EX}_{k} min\_en \equiv$

$$\bigvee_{i=0}^{k} E_1X_1 \ldots E_iX_imin\_en.$$

LTL: $A(\neg min\_en \wedge X\neg min\_en \wedge XX\neg min\_en \wedge \cdots \wedge \underbrace{X \ldots X}_{k} \neg min\_en) \equiv$

$$A(\bigwedge_{i=0}^{k} X_1 \ldots X_i \neg min\_en).$$

Notice that the property expressed in LTL actually is the negation of property F1. However, it is sufficient to complement the set of states that satisfy this property to obtain the set of states satisfying F1.

F2: Is energy the minimum one? Alternatively, if energy is the maximum one, is it possible to reach a state with minimum energy without passing through states with intermediate energy?
CTL, LTL: $A(max\_en\ U\ min\_en)$.

F3: Is it possible to reach in one step a folding where the first half of the sequence is a helix of the form $rrllrr \ldots$?
Here we must distinguish between the case in which $m = \lfloor n/2 \rfloor$ is even and that in which it is odd.
If m is odd, we have:
CTL: $EX(\bigwedge_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(ith\_r \wedge i+1th\_r) \wedge \bigwedge_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(ith\_l \wedge i+1th\_l))$.

LTL: $AX(\bigvee_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(\neg ith\_r \vee \neg i+1th\_r) \vee$
$\bigvee_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(\neg ith\_l \vee \neg i+1th\_l))$.

If $m = 2 + 4 \cdot j, j \geq 0$, we have:
CTL: $EX(\bigwedge_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(ith\_r \wedge i+1th\_r) \wedge \bigwedge_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(ith\_l \wedge i+1th\_l) \wedge$
$mth\_r)$.

LTL: $AX(\bigvee_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(\neg ith\_r \vee \neg i+1th\_r) \vee$
$\bigvee_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(\neg ith\_l \vee \neg i+1th\_l) \vee \neg mth\_r)$.

If $m = 4 + 4 \cdot j, j \geq 0$, we have:
CTL: $EX(\bigwedge_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(ith\_r \wedge i+1th\_r) \wedge \bigwedge_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(ith\_l \wedge i+1th\_l) \wedge mth\_l)$.

LTL: $AX(\bigvee_{i=2,i=2+4\cdot j,j\geq 0}^{m-1}(\neg ith\_r \vee \neg i+1th\_r) \vee$
$\bigvee_{i=4,i=4+4\cdot j,j\geq 0}^{m-1}(\neg ith\_l \vee \neg i+1th\_l) \vee \neg mth\_l)$.

F4: Is it true that every state which is at most k steps far from the current one has maximum energy, i.e., energy equal to 0?
CTL: $max\_en \wedge AXmax\_en \wedge \cdots \wedge \underbrace{AX \ldots AX}_{k} max\_en \equiv$

$$\bigwedge_{i=0}^{k} A_1X_1 \ldots A_iX_imax\_en.$$

LTL: $A(max\_en \wedge Xmax\_en \wedge \cdots \wedge \underbrace{X \ldots X}_{k} max\_en) \equiv$

$$A(\bigwedge_{i=0}^{k} X_1 \ldots X_imax\_en).$$

In the next section we report the outcomes of some experiments where we model checked these (and other) properties on proteins of small dimension.

5 Experimental results and future developments

We implemented the proposed approach to the verification of properties of foldings in SICStus Prolog and we experimented it on some simple test cases. More precisely, we developed an algorithm for encoding 2D Protein Transition Systems, and then we implemented model checking algorithms to verify whether some specific 2D Protein Transition Systems satisfy or not a set of relevant properties, including F1-F4. We confined ourselves to test cases where protein length was at most 10. As for $F1$, for instance, we searched for states with energy equal to 0 that satisfy property $F1$ when $k = 1$, i.e., states with maximum energy that reach in one step a state with minimum energy. For n=8, given the string 111111111, where $min\_en = -4$, it came out that only 8 states fulfil the request. They are (every state is followed by the state testifying the satisfiabily of the property): $lrflflf \rightarrow llflflf$, $lfflflf \rightarrow llflflf$, $rlfrfrf \rightarrow rrfrfrf$, $rffrfrf \rightarrow rrfrfrf$, $flflfrl \rightarrow flflfll$, $flflffl \rightarrow flflfll$, $frfrflr \rightarrow frfrfrr$, and $frfrffr \rightarrow frfrfrr$. Similar experiments were performed in the cases of properties F2-F4. We used our tool to model check a number of other meaningful properties. As an example, we used it to check whether there exist states with an energy different from the minimum one that may reach in one step a state with a greater energy which, in its turn, may reach in a few steps (how many, it depends on the length of the protein) a state with minimum energy. The answer is positive. For example, for n=7, given the string 11111111, where $min\_en = -3$, the following state satisfies the property (the entire witness path is reported and every state is followed by its energy): $lrlfll(-2) \rightarrow lrlffl(0) \rightarrow lrllfl(-3)$. The existence of such paths shows that,

in order to decrease the number of edges of the 2D Protein Transition System, it is not sound to cut edges connecting states where the source energy is lower than the destination energy because from the destination state we could rapidly reach states with minimum energy.

As for the future developments of our work, one of the main issues of model checking is the state explosion problem. In our case, a protein of length n gives rise to a transition system where the number of states is $\Theta(3^{n-1})$. This leads to both time and space problems. On-the-fly model checking [6, 10] has been proposed to cope with the state explosion problem. This approach in many cases avoids the construction of the entire state space of the system, because the property to test guides the construction of the system. When a state falsifying the property under analysis is reached, the construction is stopped. Only in the worst case (when the property is satisfied) the entire system must be built. Exploiting on-the-fly model checking, we plan to apply our approach to proteins with a significant length.
Another technique proposed to control the state-explosion problem is symbolic model checking [14, 4]. Symbolic model checking is based on the use of Ordered Binary Decision Diagrams (OBDDs) to compactly represent transition systems. In the worst case, the OBDD and the represented system have the same size. However, this is usually not the case when the transition system has some "regularities". We intend to study what happens if we use OBDDs to represent 2D Protein Transition Systems and, if possible, to exploit symbolic model checking techniques.
Finally, we plan to extend our approach to 3D-lattices and to switch to an energy model that considers all the 20 kinds of aminoacids. In this context we intend to analyse the usefulness of our approach not only for the protein folding problem, but more in general for the study of folding/energy landscapes of proteins.

References

[1] R.Backofen and S.Will. Excluding symmetries in constraint-based search. *Constraints*, 7(3):333–349, 2002.

[2] R.Backofen and S.Will. A Constraint-Based Approach to Structure Prediction for Simplified Protein Models that Outperforms Other Existing Methods. *Proc. of ICLP 2003*, pp. 49-71, Springer Verlag, 2003.

[3] M.Berrera, H.Molinari, and F.Fogolari. Amino acid empirical contact energy definitions for fold recognition in the space of contact maps. *BMC Bioinformatics*, 4(8), 2003.

[4] E.M.Clarke, O.Grumberg, and D.A.Peled. *Model Checking*. The MIT Press, 1999.

[5] P.Clote and R.Backofen. *Computational Molecular Biology*. John Wiley & Sons, 2001.

[6] C.Courcoubetis, M.Y.Vardi, P.Wolper, and M. Yannakakis. Memory efficient algorithms for the verification of temporal properties *Formal Methods in System Design*, 1:275-288, 1992.

[7] A.Dal Palù, A.Dovier and F.Fogolari. Constraint logic programming approach to protein structure prediction. *BMC Bioinformatics*, 5(186), 2004.

[8] A.Dal Palù, A.Dovier and E. Pontelli. A Constraint Logic Programming Approach to 3D Structure Determination of Large Protein Complexes. *Proc. of LPAR'05*, pp. 48-63, 2005.

[9] E.A. Emerson *Temporal and modal logic.* In Handbook of Theoretical Computer Science, Volume B (chapter 16), J. van Leeuwen ed., Elsevier Science Publisher, 1990.

[10] J.C.Fernandez, C.Jard, T.Jeron, and G.Viho. Using on-the-fly verification techniques for the generation of test suites. In *Proceedings of the 1996 Workshop on Computer-Aided Verification*, LNCS 1102:348-359, 1996.

[11] A.Kolinski and J.Skolnick. Reduced models of proteins and their applications. *Polymer*, 45:511–524, 2004.

[12] K.F.Lau and K.A.Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:3986-3997, 1989.

[13] N.Madras and G.Slade. The self avoiding walk. (Boston: Birkhäuser), 1993.

[14] K.L.McMillan. Symbolic Model Checking: An Approach to the State Explosion Problem. *Kluwer Academic*, 1993.

[15] J.T.Ngo and J.Marks. Computational complexity of a problem in molecular structure prediction. *Protein Engineering* 5:313-321, 1992.

[16] A.Ponitz and P.Tittmann. Improved upper bounds for self-avoiding walks in Z^d. *Electronic J. Comb. 7*, 2000.

[17] J.P. Quielle and J.Sifakis. Specification and verification of concurrent systems in CESAR. In *Logics and Models of Concurrent Systems, NATO ASI 13*. Springer, 1984.