SIGACT News Complexity Theory Column ??

Lane A. Hemaspaandra
Dept. of Computer Science, University of Rochester
Rochester, NY 14627, USA lane@cs.rochester.edu

Counting Self-avoiding Walks in Some Regular Graphs

Maciej Liśkiewicz[1] ***Mitsunori Ogihara***[2] ***and Seinosuke Toda***[3]

1 The Complexity of Counting Simple Paths in Graphs

A self-avoiding walk (SAW, for short) is a path on a graph G that does not visit any node more than once. The problem of counting the SAWs in a given "regular" graph G, such as the one shown in Figure 1, plays a crucial role in modeling many important problems in different areas of science, such as combinatorics, statistical physics, theoretical chemistry, and computer science. By "two-dimensional grid" we mean the two-dimensional rectangular lattice Z^2 with origin $(0, 0)$. One of the most prominent applications of counting SAWs is the modeling of spatial arrangement of linear polymer molecules in a solution. Here a SAW represents a molecule composed of monomers linked together in a chain by chemical bonds. Other application areas include the percolation model, the Ising model, and the network reliability model.

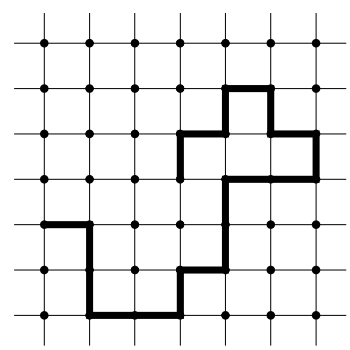

Figure 1: A length-18 in the complete 2D grid.

Valiant [Val79b] is the first to find connections between the problem of counting SAWs and computational complexity theory. He showed that the problem of counting SAWs between two given points, the problem of counting Hamiltonian cycles, and the problem of counting Hamiltonian paths between two given points are all #P-complete under polynomial parsimonious reductions (that is, polynomial-time reductions of functions not requiring post-computation) both for directed graphs and for undirected graphs. One might ask for which types of graphs these counting problems remain #P-complete. The goal of this paper is to present some natural regular graphs for which the corresponding counting problems are #P-complete and discuss some open issues.

2 Preliminaries

Let M be a nondeterministic Turing machine. By $\#acc_M$ we denote the function that maps each string x to the number of accepting computation paths of M on input x. A class #P of Valiant [Val79a] is defined as $\{\#acc_M \mid M$ is a polynomial-time nondeterministic Turing machine$\}$.

[1]© Maciej Liśkiewicz, 2003. Institut für Theoretische Informatik, Universität zu Lübeck, Wallstraße 40, D-23560, Lübeck, Germany. E-mail: liskiewi@tcs.uni-luebeck.de. On leave from Instytut Informatyki, Uniwersytet Wrocławski, Poland.

[2]© Mitsunori Ogihara, 2003. Department of Computer Science, University of Rochester, Rochester, NY 14627-0226, USA. E-mail: ogihara @cs.rochester.edu.

[3]© Seinosuke Toda, 2003. Department of Applied Mathematics, Nihon University, 3-25-40 Sakurajyou-shi, Setagaya-ku, Tokyo 156, Japan. E-mail: toda @am.chs.nihon-u.ac.jp.

Next we define polynomial-time reductions between counting functions. Let f and g be functions from Σ^* to $\mathbf{N}$. We say that f is *polynomial-time one-Turing reducible* to g, denoted by $f \leq^p_{1\text{-}T} g$, if there is a pair of polynomial-time computable functions, $R_1 : \Sigma^* \to \Sigma^*$ and $R_2 : \Sigma^* \times \mathbf{N} \to \mathbf{N}$, such that for all x it holds that $f(x) = R_2(x, g(R_1(x)))$. We consider two special cases of polynomial-time one-Turing reductions. We say that f is *polynomial-time parsimoniously reducible* to g if for all x and y the above R_2 satisfies $R_2(x,y) = y$, i.e., for all x, $f(x) = g(R_1(x))$. We say that the function f is *polynomial-time right-bit-shift reducible* to g, denoted by $f \leq^p_{r\text{-}shift} g$, if there is a polynomial-time computable function $R_3 : \Sigma^* \to \mathbf{N} - \{0\}$ such that for all x and y it holds that $R_2(x,y) = y \operatorname{div} 2^{R_3(x)}$, i.e., for all x, $f(x) = g(R_1(x)) \operatorname{div} 2^{R_3(x)}$, where div is integer division. It is easy to see that both types of reductions are transitive.

3 Counting Paths and Cycles in Planar 3-Regular Graphs

In 1976 Garey, Johnson, and Tarjan [GJT76] showed that the Hamiltonian Cycle problem is NP-complete even for the graphs restricted to planar 3-regular graphs. Let us denote the restricted version by HamCycle-Plan3 and its counting problem (that is, the problem of counting Hamiltonian cycles in planar 3-regular graphs) by #HamCycle-Plan3.

It is known for many NP-complete problems that their counting versions (which ask to compute the number of witnesses) are #P-complete. For a large number of such problems, the #P-completeness of the counting version can be shown by simply observing two properties about an existing NP-completeness proof: (1) the existing proof constructs a polynomial-time many-one reduction from an NP-complete problem whose counting version is known to be #P-complete and (2) the reduction actually preserves the number of witnesses. For example, the six standard NP-complete problems identified by Karp [K72] have this property (see page 169 of [GJ79]).

Does this property hold for HamCycle-Plan3, too? That is, does the reduction due to Garey, Johnson, and Tarjan also serve as a reduction that shows #P-completeness of #HamCycle-Plan3?

In [Pro86] Provan claims that this is indeed the case by stating that, for each 3CNF formula φ, and for each satisfying assignment α of φ, the graph produced by the Garey–Johnson–Tarjan reduction on input φ has exactly

$$(8^7 \cdot 18)^m \cdot 8^{6a} \cdot 8^b \cdot 36$$

Hamiltonian cycles corresponding to α, where m is the number of clauses of φ and a and b are quantities not depending on α. Provan uses this analysis to argue #P-completeness of #HamCycle-Plan3 and this observation is referenced by others (e.g. [HMRS98] and [Vad01]). Unfortunately, this analysis is incorrect, since two factors are missing in the formula. The correct number of Hamiltonian cycles corresponding to α is

$$2^{m_2} \cdot 3^{m_3} \cdot (8^7 \cdot 18)^m \cdot 8^{6a} \cdot 8^b \cdot 36,$$

where a and b have the same meaning as before and m_2 (respectively, m_3) is the number of clauses C in φ such that α satisfies exactly two (respectively, three) literals of C.

Now that this argument for the #P-completeness of #HamCycle-Plan3 is incorrect, can we not hope to prove the #P-completeness?

In [LOT03] it has been shown that indeed we can. The Garey–Johnson–Tarjan reduction can be modified (by restricting the type of 3CNF formulas and by replacing one of the gadgets) to prove that #HamCycle-Plan3 is #P-complete. Furthermore, a similar result has been shown for the problem of counting Hamiltonian paths.

Theorem 1 ([LOT03]) *The problem of counting Hamiltonian cycles in planar graphs of degree three and the problem of counting Hamiltonian paths in planar graphs of maximum degree three are* #P*-complete under* $\leq^{p}_{r\text{-}shift}$*-reductions.*

In Section 5 we will sketch briefly the main idea behind a correct proof of the theorem.

4 The Complete Two-Dimensional Grids

Another type of important planar graphs is the two-dimensional grid and its fragments. Interestingly, we know much less about the complexity of counting paths in this type of graphs. Evaluation of the number of length-n SAWs from the origin in the two-dimensional grid for a given n (which quantity we'll refer to by c_n) has been attracting many researchers. The term "self-avoiding walks" is often used to refer to this specific problem.

The question of whether there exists a formula for c_n has been extensively studied in the literature (see [MS93] and [Wel93] for a survey), but is still open. One of the major achievements in this direction is a result of Hammersley and Morton [HM54], which states that there exists some $\mu > 0$ such that $\lim_{n\to\infty} c_n^{1/n} = \mu$ and such that, for all $n \geq 0$, $\frac{c_n}{\mu^n} \geq 1$. However, the exact value of μ is unknown. Hammersley and Welsh [HW62] show that there exists some constant $a > 0$ such that $\frac{c_n}{\mu^n} = O(a^{\sqrt{n}})$. Much work has been done to improve lower and upper bounds of μ. The current best bounds are $2.62002 \leq \mu$ and $\mu \leq 2.67919$, which are respectively shown in [CG96] and in [PT00]. A conjecture in statistical physics states that for some constants A and γ it holds that

$$c_n = A\mu^n n^{\gamma-1}(1 + o(1))$$

and similar conjectures have been made for grids of higher dimension (see, e.g. [RS00]). Assuming that the conjecture for the two-dimensional grid holds, experiments seem to suggest that $\gamma = \frac{43}{32}$ (see [MS93] for more discussions).

Since the formula for c_n is unknown, much efforts has been given to calculate the exact value of c_n for as large n as possible. As of today, the value has been calculate for all n up to 51 [CG96]. Also, Randall and Sinclair [RS00] present Monte Carlo algorithms for approximating the value of c_n, and for generating SAWs of a given length almost uniformly at random.

Using NP-completeness as a yardstick for measuring computational complexity of problems, Istrail [Is00] has recently provided negative answers to conjectures about computability of some thermodynamical quantities in two- and three-dimensional square lattices. More precisely, Barahona [Ba82] shows that, for the much-studied Ising model on the three-dimensional cubic lattice, the problem of computing the ground states on finite sublattices is NP-hard, and thus, intractable. Istrail extends this result and proves that the problem remains NP-hard for every non-planar lattice, including the two-dimensional non-planar lattices.

Computational complexity of computing c_n is first addressed by Welsh [Wel93]:

Welsh's Problem *Is the exact computing of c_n complete for* $\#\mathrm{P}_1$*?*

Here $\#\mathrm{P}_1$ is the version of #P in which the inputs are over a single-letter alphabet and so the length n is specified by the input length[4]. If the answer to Welsh's question is affirmative then it means that determining the strict value for c_n may be computationally intractable and hence that no exact formula for c_n exists (see [GOR00] for hardness of $\#\mathrm{P}_1$).

[4] In [Wel93, Problem 1.7.3] Welsh asks about #P-completeness, but we believe he meant $\#\mathrm{P}_1$-completeness. See [LOT03] for detail.

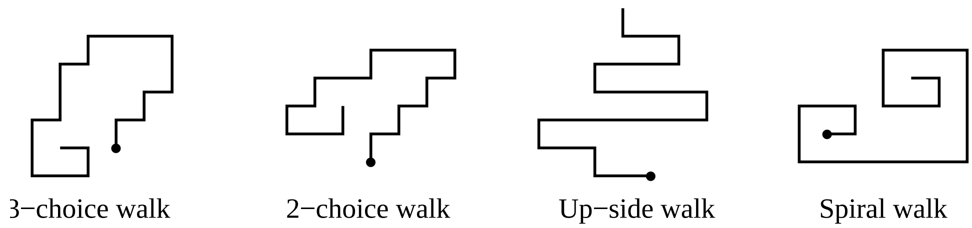

Figure 2: Various types of self-avoiding walks: the only solved models are spiral and up-side walks.

In addition to the general self-avoiding walks, some modified walks modeling various physical conditions have been subjects of considerable amount of research. Exact formulas for such modified SAWs could lead to discovery of the exact formula for the general case or better estimates of the constant μ. Unfortunately, counting is difficult even in some very restricted models. Important special cases are those in which the choices for the next move are restricted by the current move. Example of such walks are 3-choice walks, 2-choice walks, spiral SAWs, and up-side SAWs. In 3-choice walks clockwise turns are forbidden after moving horizontally, that is, no “up” move right after a “left” move and no “down” move after a “right” move. In 2-choice walks one more restriction is added to 3-choice walks: no two successive moves are vertical. In spiral walks any clockwise turns are forbidden and in up-side walks “down” moves are forbidden (see Figure 2). The only non-trivial models in which the exact formula for the number of SAWs is known are the spiral walks [BH84, GW84, Jo84, Wh84] and the up-side walks [PS89, Wil96]. In Section 6 we show that using a finite-automata-approach one can derive the exact formula for the number of the up-side SAWs. The exact computational complexity of computing the number SAWs in these models is an interesting open question.

4.1 Counting in Grid Graphs Allowing Holes is Difficult

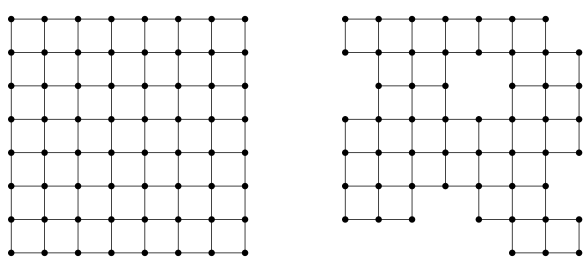

Figure 3: A fragment of complete 2D grid and a grid with holes.

It is straightforward to prove that the exact counting of SAWs for the complete two-dimensional grid belongs to $\#\mathrm{P}_1$ (consider a nondeterministic Turing machine that, on input 0^n, nondeterministically generates a sequence of n moves and then accepts if and only if the sequence is self-avoiding). On the other hand, settling Welsh’s question of whether the counting problem is $\#\mathrm{P}_1$-hard appears to be difficult. A straightforward approach to proving the hardness will be to embed the computation of a nondeterministic Turing machine in the grid, but doing this so that there are no holes and the computation paths have the same length seems quite difficult. This leads us to the question of whether the counting problem is hard for some complexity class if it is permitted to create holes, that is, if the graphs in which SAWs are counted are those composed of the nodes and the edges of the complete two-dimensional grid. Since the number of choices for locations of the holes in a finite two-dimensional grid is exponential in the number of nodes in the grid, here we should be rather thinking about $\#\mathrm{P}$-hardness than $\#\mathrm{P}_1$-hardness. Thus, we ask: *Is counting SAWs of a specified length in finite subgraphs of complete two-dimensional grids* $\#\mathrm{P}$*-complete under some* *polynomial-time function reductions?* In [LOT03] this problem has been positively resolved.

Theorem 2 ([LOT03]) *Each of the following six problems about counting SAWs in subgraphs of complete two-dimensional grids is #P-complete under $\leq^p_{r\text{-}shift}$-reductions.*

1. *counting the SAWs from the origin to a specific point having a specific length,*
2. *counting the SAWs from the origin to any point having a specific length,*
3. *counting the SAWs between any two points having a specific length,*
4. *counting the SAWs from the origin to a specific point having any length,*
5. *counting the SAWs from the origin to any point having any length, and*
6. *counting the SAWs between any two points having any length.*

Unfortunately, the above result does not resolve the question of Welsh.

A related question is whether the problem is #P-complete if the input graph is finite and without holes (that is, both the input graph and its complement are connected) (see Figure 4). We conjecture that this counting problem is #P-complete.

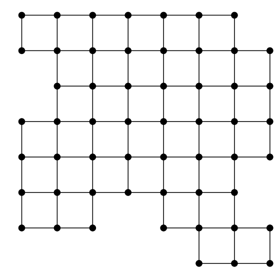

Figure 4: A solid grid.

Another related question is whether counting Hamiltonian cycles or Hamiltonian paths in finite subgraphs of the two-dimensional grid (holes permitted) is #P-complete under some polynomial-time function reductions. It is known that the decision versions of these problems are NP-complete [IPS82], but the reductions shown in [IPS82] do not possess the witness-preserving property (even with multiplicity). We conjecture that these counting problems are #P-complete.

One can also question for which subgraphs of the two-dimensional grid counting Hamiltonian cycles or Hamiltonian paths is in P. The question is unsolved for such simple subgraphs as rectangles, and only a small progress has been made toward obtaining exact formulas for the corresponding quantities. Göbel [Gö79] gives the correct number of Hamiltonian cycles for a $3 \times \ell$ grid. In [SS95] Stoyan and Strehl show that the number of Hamiltonian cycles in a $k \times \ell$ rectangular grid for fixed k has always a rational generating function, and provide generating functions for all k up to 8. In [CK97] Collins and Krompart give a generating function for the number of Hamiltonian paths from one corner of a $k \times \ell$ rectangular to another corner for all k up to 5. One can show that the number of Hamiltonian paths or Hamiltonian cycles in a $k \times \ell$ rectangular grid can be computed in polynomial time for each $k \in O(\log \ell)$. We assume here a unary encoding of both k and ℓ. To the knowledge of the authors, it is unknown whether the counting problem is polynomial-time solvable if $k \in \omega(\log \ell)$. In the case when $k \in O(\log \ell)$ we can say something stronger: both counting problems are in #L, the logarithmic-space version of #P (for a definition and properties of #L see [AO96, AJ93]).

5 #P-completeness of #HamCycle-Plan3

Here we quickly sketch how the reduction of Garey, Johnson, and Tarjan [GJT76] can be modified to act as a reduction from #HamCycle-Plan3 to #3SAT. The reduction we'll construct has a special property: for each graph G produced by the reduction, there is at least one edge traversed by all the Hamiltonian cycles of G and such an edge is easy to identify. So, by simply removing one of such common edges, the Garey–Johnson–Tarjan reduction becomes a polynomial-time many-one reduction from 3SAT to the Hamiltonian Path decision problem of planar graphs having maximum

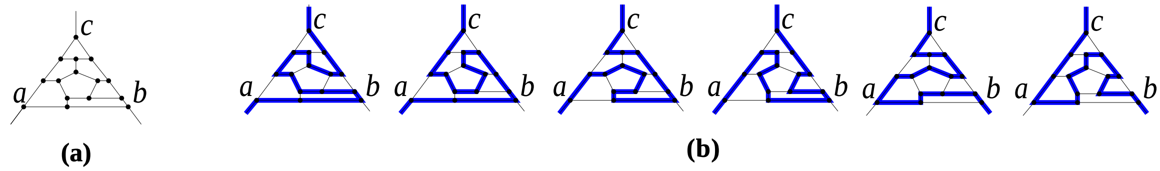

Figure 5: The Tutte-Gadget. (a) The gadget. (b) Hamiltonian traversals of the nodes in the Tutte-gadget. The top four are traversals connecting nodes a and c with b in the middle. The bottom two are traversals connecting nodes b and c with a in the middle.

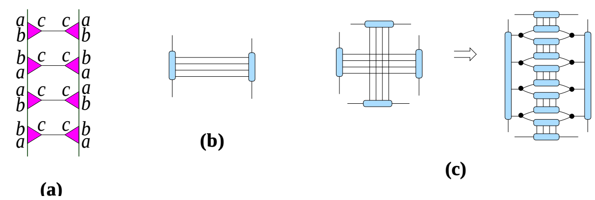

Figure 6: The XOR-Gadget. (a) The structure of the gadget: each of the eight shaded triangles represents a copy of the Tutte-gadget placed in the structure as oriented above. (b) A symbol to denote an XOR-gadget. (c) Crossing of two XOR-gadgets. On the left the four horizontal lines in one XOR-gadget and the four vertical lines in the other XOR-gadget need to be crossed. On the right the crossing of the lines are resolved by introduction of four additional XOR-gadgets.

degree three (we denote this problem by HamPath-Plan$\hat{3}$ and the corresponding counting problem by #HamPath-Plan$\hat{3}$).

As described earlier, the problem in the use of the Garey–Johnson–Tarjan reduction in showing #P-completeness of #HamCycle-Plan3 is that the number of Hamiltonian paths representing a satisfying assignment depends on how it satisfies the formula. The problem can be overcome by considering only instances of Not-All-Equal-3SAT (i.e. a problem of testing whether a given 3CNF formula can be satisfied by a not-all-equal satisfying assignment) and applying slight modifications to the Garey–Johnson–Tarjan reduction.

In [LOT03] it is shown that there is a polynomial-time reduction f with the following property: For each $n \geq 1$, for each $m \geq 1$, and for each 3CNF formula φ having n variables and m clauses, $\psi = f(\varphi)$ is a CNF formula such that:

1. The number of variables of ψ is $n + m + 1$ and the number of clauses of ψ is $8m + 1$. Of the $8m + 1$ clauses in ψ exactly one is a single-literal clause and the rest are three-literal clauses.

2. Every satisfying assignment of ψ is a not-all-equal satisfying assignment. More precisely, for every satisfying assignment α of ψ, α satisfies exactly two literals for exactly $4m$ three-literal clauses and satisfies exactly one literal for exactly $4m$ three-literal clauses.

3. $\#\text{SAT}(\varphi) = \#\text{SAT}(\psi)$.

Let φ be a 3CNF formula having n variables and m clauses for which we want to compute #SAT. Let $\psi = f(\varphi)$. We will construct a graph G from ψ by applying a modified Garey–Johnson–Tarjan reduction. Basic components of the construction are the Tutte-gadget (see Figure 5), the

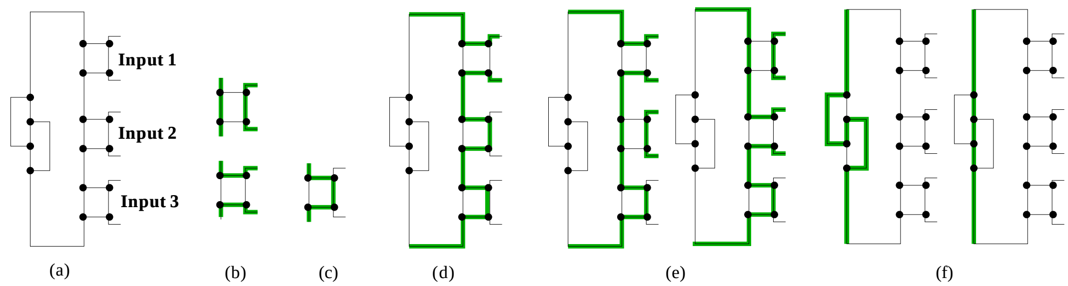

Figure 7: The OR-Gadget. (a) A three-input OR-gadget. (b) Two ways to traverse an input line whose value is true. (c) The way to traverse an input line whose value is false. (d) The traversal of input lines when only input 1 is true. (e) Two possible traversals of input lines when exactly two inputs are true. (f) Two ways to create a Hamiltonian traversal of the four nodes on the left side of the gadget.

XOR-gadget (see Figure 6), and the OR-gadget (see Figure 7). Here the first two gadgets are taken without changes from the Garey–Johnson–Tarjan reduction while the OR-gadget is our own device.

The Tutte-gadget is used to force branching. To visit c without missing a node, one has to either enter from a and visit b on its way or enter from b and visit a on its way. There are four ways to do the former and two ways to do the latter (see Figure 5 (b)).

The XOR-gadget is a ladder built using eight copies of the Tutte-gadget (see Figure 6 (a)). To go through all the nodes in an XOR-gadget one has to enter and exit on the same vertical axis. Moreover for each of the two vertical axes there are $(4 \cdot 2)^4 = 2^{12}$ Hamiltonian paths that traverse the gadget. XOR-gadgets can be crossed without losing planarity by inflating the number of Hamiltonian paths (see Figure 6). Since four XOR-gadgets are added, the number of Hamiltonian paths is increased by a multiplicative factor of 2^{48}.

In an OR-gadget, each four-node rectangle on the right-hand side is considered to be an input. The line at the right end of an input part provides connection to the outside world. The one shown in Figure 7 is a three-input OR-gadget. Each input line will be either entirely available or entirely unavailable. If an input line is available we think of the situation as the input being assigned true as a value. Otherwise, we think of the situation as the input being assigned false. In the former case, there are two ways to traverse the line (see Figure 7 (b)). In the latter there is only one way to touch the two nodes on the line (see Figure 7 (c)). If the number of inputs that are assigned true is one, there is only one way to traverse the nodes on the input lines (see Figure 7 (d)). If the number is two, there are two possibilities (see Figure 7 (e)). Finally, there are two ways to traverse all the four nodes on the left-hand side of an OR-gadget (see Figure 7 (f)).

Another important components of the graph G are two-node cycles. A two-node cycle is represented as a pair of nodes that are vertically lined up and are connected by two edges. We refer to the two nodes by the *top node* and the *bottom node* and refer to the two arcs by the *right edge* and the *left edge*. The paths consisting solely of one of the two arcs are the Hamiltonian paths of a two-node cycle.

The graph G is essentially two vertical sequences of two-node cycles that are side-by-side. The right sequence is called Π and the left sequence is called Σ here. In each of the two sequences each neighboring cycle-pair is joined by an edge. The cycles in Σ correspond to the literals of ψ, so there are exactly $24m + 1$ cycles in it. On the other hand, the cycles in Π correspond to truth assignments, so there are $2(n + m + 1)$ cycles in it. The sequences Σ and Π are joined by an edge

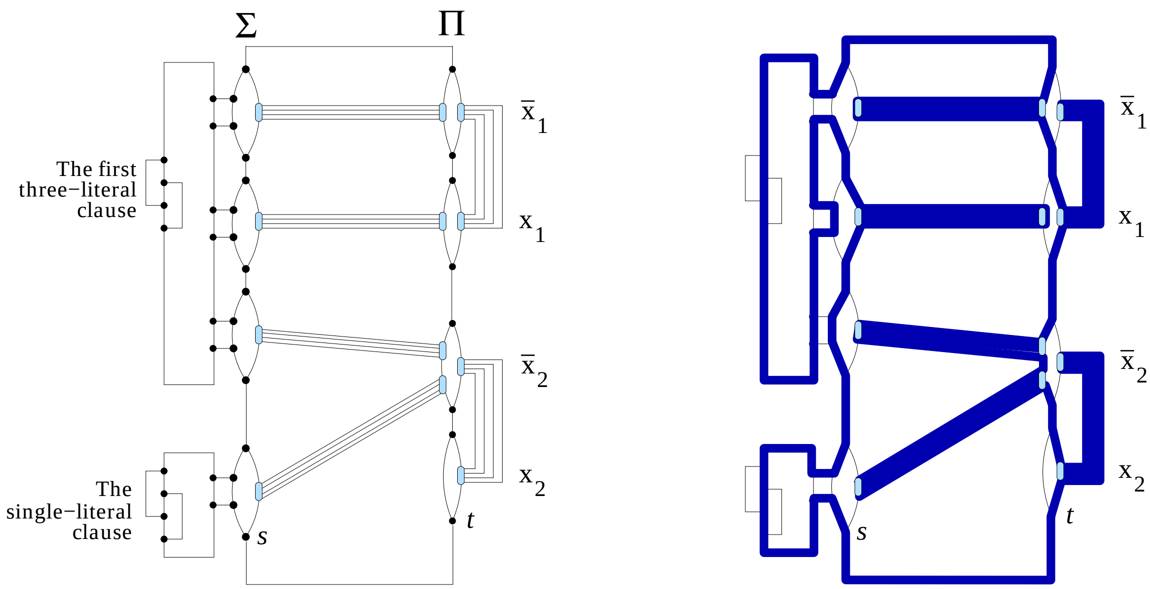

Figure 8: The reduction applied to a formula $\psi = (\overline{x}_1 \vee x_1 \vee \overline{x}_2) \wedge \overline{x}_2$. Left: the graph. Right a Hamiltonian cycle corresponding to satisfying assignment $(x_1 = 0, x_2 = 0)$. Every satisfying assignment of ψ satisfies exactly two literals of the three-literal clause. Since there are two variables and four literals and there are no crossing XOR-Gadgets, each satisfying assignment of ψ corresponds to exactly $2 \cdot 4 \cdot 2^{12 \cdot 2} \cdot 2^{12 \cdot 4}$ Hamiltonian cycles.

connecting the very top nodes of Σ and Π and by an edge connecting the very bottom nodes of Σ and Π, called s and t respectively. The first $24m$ cycles of Σ are divided into $8m$ three-cycle blocks, where for each i, $1 \leq i \leq 8m$, the ith block corresponds to the ith three-literal clause of ψ, i.e., the first of the three cycles corresponds to the first literal of the clause, the second cycle to the second literal, and the third cycle to the third literal (see Figure 8). The three cycles in each three-cycle block are connected to each other by an OR-gadget attached on their left edges. The last cycle corresponds to the unique single-literal clause of ψ. To the left edge of the two-node cycle we attach a one-input OR-gadget. The sequence Π is divided into $n + m + 1$ blocks of cycle-pairs, where for each i, $1 \leq i \leq n + m + 1$, the ith pair corresponds to x_i, the ith variable. For each pair, the top cycle corresponds to $\overline{x}_i$ and the bottom to x_i. We connect the right edges of each pair by an XOR-gadget as shown in Figure 8. This has the effect of forcing each Hamiltonian path of Π to select for each pair of two-node cycles exactly one cycle whose left edge is traversed, where the other cycle will be traversed on the right edge. Now, for each i, $1 \leq i \leq 2(n + m + 1)$, and each j, $1 \leq j \leq 24m + 1$, if the literal represented by the ith cycle of Π is the literal at the jth position in Σ, join the left edge of the ith cycle in Π and the right edge of the jth cycle in Σ by an XOR-gadget. Here, in the case where two XOR-gadgets connecting Σ and Π need to be crossed, we do so with the method for crossing XOR-gadgets described earlier.

Note that for every i, $1 \leq i \leq n + m + 1$, traversing the XOR-gadget connecting the two right edges in the ith cycle-pair in Π corresponds to selection of a value for the ith variable. Here if the right edge of the top (respectively, the bottom) cycle is used to traverse the XOR-gadget, then it frees up the left edge of the bottom (respectively, the top) cycle, enforcing that all the XOR-gadgets attached to the left edge of the bottom (respectively, the top) cycle are traversed from the Π side and that all the XOR-gadgets attached to the left edge of the top (respectively, the bottom) cycle are traversed from the Σ side. We view this situation as the ith variable assigned the value true (respectively, false). So, each Hamiltonian path of the Π side corresponds to a truth assignment of ψ. Let a Hamiltonian path of the Π side be fixed and let α be the corresponding truth-assignment of ψ. Let j, $1 \le j \le 8m$, be an integer. If α does not satisfy the jth clause, then in each of the three cycles in the jth block in Σ, the right edge needs to be taken so as to traverse the XOR-gadgets attached to it, which means that the OR-gadget attached to the block cannot be traversed. If α satisfies exactly one literal, exactly one of the three left edges of the three cycles is available, so there are two ways to traverse all the nodes in the block. If α satisfies exactly two literals, exactly two of the three left edges of the three cycles are available, so there are four ways to traverse all the nodes in the block. As to the last single-cycle with one single-input OR-gadget, if the literal is not satisfied, then there is no way to traverse the OR-gadget, and if the literal is satisfied, then there are two ways to construct a Hamiltonian path in it. The formula ψ is designed so that every satisfying assignment of ψ satisfies exactly $4m$ three-literal clauses by satisfying exactly two literals and exactly $4m$ three-literal clauses by satisfying exactly one literal. So, each Hamiltonian path of G corresponds to a satisfying assignment of ψ. Furthermore, for each satisfying assignment α of G, the number of Hamiltonian paths of G that represent α is

$$2\big(4^{4m}2^{4m}2^{12(n+m+1)}2^{12(24m+1)}2^{48r}\big) = 2^{48r+12n+312m+25},$$

where r is the number of crossings of XOR-gadgets. Note that the degree of the nodes in G is all three. Since #SAT(φ) = #SAT(ψ), #SAT(φ) can be computed from the number of Hamiltonian cycles in G by right-shifting $48r+12n+312m+25$. Thus, we have strengthened the NP-completeness result by Garey, Johnson, and Tarjan as follows:

Lemma 5.1 *The Hamiltonian Cycle Problem of planar 3-regular graphs is* NP*-complete in the sense that there exists a polynomial-time many-one reduction f from 3SAT to the problem such that, together with another polynomial-time computable function, f acts as a polynomial-time $\le^p_{r\text{-}shift}$-reduction from* #3SAT *to* #HamCycle-Plan3.

In the above construction, the edge (s,t) is traversed by every Hamiltonian cycle of G (see Figure 8). Thus if we remove the edge, then each st-Hamiltonian path in the resulting graph corresponds to exactly one Hamiltonian cycle in G. This proves the #P-completeness of the #HamPath-Plan$\hat{3}$ problem.

6 Up-side SAWs in Complete Grids

In this section we show how a finite-automaton can be used to derive an exact formula for the number of the up-side SAWs (that is, SAWs without down moves) in two-dimensional grid. Let us denote by $L_{\text{up-side}}$ the set of all strings over $\{U, L, R\}$ that encode up-side SAWs. Obviously this language can be recognized by a deterministic 1-way finite automaton. Below we show how using this fact one can deduce a formula for the number of up-side SAWs of a given length n. Let us fix the automaton M for $L_{\text{up-side}}$ as presented in Figure 9, with the states $Q = \{1, 2, 3\}$, where 1 is the initial state and all the states are accepted states. Let A the transition matrix of M: For all i and

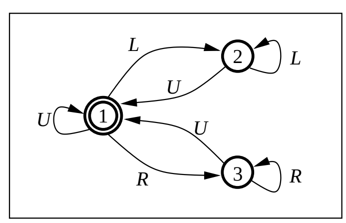

Figure 9: DFA M for up-side SAWs.

j, $1 \le i, j \le 3$, $A(i,j) = 1$ if and only if M can move from state i and state j in one step. Then

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Define sequences $\{f_n\}_{n\ge1}$, $\{g_n\}_{n\ge1}$, and $\{h_n\}_{n\ge1}$ as follows: For all $n \ge 1$,

$$\begin{bmatrix} f_n \\ g_n \\ h_n \end{bmatrix} = A^n \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Then it is easy to see that, for all integers $n \ge 1$, f_n (respectively, g_n and h_n) is equal to the number of strings having length n that are accepted by M when it is forced start in state 1 (respectively, 2 and 3). Thus, f_n is equal to the number of up-side SAWs having length n. To obtain a formula for f_n, observe that

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix}$$

and that, for all $n \ge 1$, if

$$A^n \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_n \\ b_n \\ b_n \end{bmatrix}$$

then

$$\begin{bmatrix} a_{n+1} \\ b_{n+1} \\ b_{n+1} \end{bmatrix} = A^{n+1} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_n \\ b_n \\ b_n \end{bmatrix} = \begin{bmatrix} a_n + 2b_n \\ 2b_{n-1} \\ 2b_{n-1} \end{bmatrix}.$$

It is easy to check that the integer sequences $\{a_n\}_{n\ge1}$ and $\{b_n\}_{n\ge1}$ defined above satisfy for all integer $n \ge 1$

$$\begin{aligned} b_n &= \tfrac{1}{2\sqrt{2}}\left[(1+\sqrt{2})^{n+1} - (1-\sqrt{2})^{n+1}\right] \quad \text{and} \\ a_n &= \tfrac{1}{2}\left[(1+\sqrt{2})^{n+1} + (1-\sqrt{2})^{n+1}\right] \end{aligned}$$

(for a generating function of this sequence see e.g. [SP95]). Thus, the formula for a_n gives the number of up-side SAWs having length n on the two-dimensional grid. This is an alternative method for obtaining the formula by Williams [Wil96].

Acknowledgments

The work of the first author has been partially supported by DFG research grant Re 672/3. The second author is supported in part by NSF grants EIA-0080124, DUE-9980943, and EIA-0205061, and in part by NIH grant P30-AG18254.

References

[AO96] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO - Theoretical Informatics and Applications*, 30: 1–21, 1996.

[AJ93] C. Álvarez and B. Jenner. A very hard Log-space counting classes. *Theoretical Computer Science*, 107: 3–30, 1993.

[Ba82] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(3): 3241–325, 1982.

[BH84] H. Blöte and H. Hilhorst. Spiralling self-avoiding walks: an exact solution. *Journal of Physics A: Mathematical and General*, 17: L111–L115, 1984.

[CG96] A. R. Conway and A. J. Guttmann. Square lattice self-avoiding walks and corrections-to-scaling. *Physical Review Letters*, 77(26): 5284–5287, 1996.

[CK97] K. Collins and L. Krompart. The Number of Hamiltonian Paths in a Rectangular Grid, *Discrete Mathematics* 169: 29–38, 1997.

[GJ79] M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman and Company, 1979.

[GJT76] M. Garey, D. Johnson, and E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4): 704–714, 1976.

[Gö79] F. Göbel. On the number of Hamilton cycles in product graphs. *Technische Hogeschool Twente*, Memorandum Nr. 289, 1979.

[GOR00] J. Goldsmith, M. Ogihara, and J. Rothe. Tally NP sets and easy census functions. *Information and Computation*, 158: 29–52, 2000.

[GW84] A. Guttmann and N. Wormald. On the number of spiral self-avoiding walks. *Journal of Physics A: Mathematical and General*, 17: L271–L274, 1984.

[HM54] J. M. Hammersley and K. W. Morton. Poor man's Monte Carlo. *Journal of the Royal Statistical Society B*, 16: 23–38, 1954.

[HW62] J. M. Hammersley and D. J. A. Welsh. Further results on the rate of convergence to the connective constant of the hypercubical lattice. *Quarterly Journal of Mathematics, Oxford*, 13(2): 108–110, 1962.

[HMRS98] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, and R. E. Stearns. The complexity of planar counting problems. *SIAM Journal on Computing*, 27(4): 1142–1167, 1998.

[Is00] S. Istrail. Statistical mechanics, three-dimensionality and NP-completeness: I. Universality of intractability for the partition function of the Ising model across non-planar surfaces. In *Proceedings of 31st Symposium on Theory of Computing*, pages 87–96, ACM Press, 2000.

[IPS82] A. Itai, C. Papadimitriou, and J. Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4): 676–686, 1982.

[Jo84] G. Joyce. An exact formula for the number of spiral self-avoiding walks. *Journal of Physics A: Mathematical and General*, 17: L463–L467, 1984.

[K72] R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, eds., *Complexity of Computer Computations*, pages 85–103, Plenum Press, New York, 1972.

[LOT03] M. Liśkiewicz, M. Ogihara, and S. Toda. The Complexity of Counting Self-Avoiding Walks in Two-Dimensional Grid Graphs and in Hypercube Graphs. *Theoretical Computer Science*, **304**(1–3): 129–157, 2003.

[MS93] N. Madras and G. Slade. *The Self-Avoiding Walk*. Birkhäuser, Boston, MA, 1993.

[PT00] A. Pönitz and P. Tittman. Improved upper bounds for self-avoiding walks in Z^d. *The Electronic Journal of Combinatorics*, 7(1) #R21, 2000. http://www.combinatorics.org/Volume_7/v7i1toc.html.

[PS89] V. Privman and N.M. Švrakić. *Directed Models of Polymers, Interfaces, and Clusters: Scaling and Finite-Size Properties.* Lecture Notes in Physics 338, Springer-Verlag, Berlin, 1989.

[Pro86] J. S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM Journal on Computing*, 15(3): 694–702, 1986.

[RS00] D. Randall and A. Sinclair. Self-testing algorithms for self-avoiding walks. *Journal of Mathematical Physics*, 41(3): 1570–1584, 2000.

[Sch78] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of 10th Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.

[SP95] N. J. A. Sloane and S. Plouffe. *The Encyclopedia of Integer Sequences.* Academic Press, San Diego, CA, 1995.

[SS95] R. Stoyan and V. Strehl Enumeration of Hamiltonian Circuits in Rectangular Grids. *Séminaire Lotharingien de Combinatoire*, Issue 34(B34f), 21 pp., 1995.

[Vad01] S. P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM Journal on Computing*, 31(2): 398–427, 2001.

[Val79a] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2): 189–201, 1979.

[Val79b] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3): 410–421, 1979.

[Wel93] D. Welsh. *Complexity: Knots, Colourings and Counting.* Cambridge University Press, 1993.

[Wh84] S. Whittington. The asymptotic form for the number of spiral self-avoiding walks. *Journal of Physics A: Mathematical and General*, 17: L117–L119, 1984.

[Wil96] L. Williams. Enumerating up-side self-avoiding walks on integer lattices. *Electronic Journal on Combinatorics*, 3(1): Research Paper 31, 1996.