Available online at www.sciencedirect.com

ScienceDirect

Information Processing Letters 105 (2008) 124–130

Information Processing Letters

www.elsevier.com/locate/ipl

On the number of Go positions on lattice graphs

Graham Farr [*], Johannes Schmidt [1]

Clayton School of Information Technology, Monash University, Clayton, Victoria 3800, Australia

Received 5 April 2006

Available online 29 August 2007

Communicated by F. Dehne

Abstract

We use transfer matrix methods to determine bounds for the numbers of legal Go positions for various numbers of players on some planar lattice graphs, including square lattice graphs such as those on which the game is normally played. We also find bounds on limiting constants that describe the behaviour of the number of legal positions on these lattice graphs as the dimensions of the lattices tend to infinity. These results amount to giving bounds for some specific evaluations of Go polynomials on these graphs.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Combinatorial problems; Combinatorial enumeration; Lattice graphs; Games; Go; Transfer matrices

1. Introduction

The game of Go is one of the oldest, most popular and intellectually challenging board games in the world [3,10]. In that game (known as *Igo* in Japan, *Weiqi* in China and *Baduk* in Korea), two players (Black and White) take turns placing stones on the vertices of a 19×19 square grid (with 19^2 vertices). Each player tries to enclose as much vacant territory as possible, and players may capture stones (and connected groups thereof) of the other colour by completely surrounding them. In the next section, we give those definitions from the game that we need. See [3,10] for further information.

In this paper we are concerned with the number of legal Go positions on grids of various types and sizes. This follows earlier work by one of us [8] on polynomials that give the numbers of legal Go positions when the game is played on general graphs with arbitrary numbers of players. Recursive schemes for computing these polynomials were given, and they were shown to be #P-hard to compute in general, even for just two players. (For the class #P, see [16,18].) In the present paper, we seek values or bounds for some particular evaluations of these polynomials for graphs of special form, namely lattice graphs similar to those on which Go is actually played. We also find estimates and bounds for certain constants that determine the limiting behaviour of the numbers of legal positions as some lattice dimension tends to infinity.

To determine these values and bounds, we use transfer matrix methods. These have been used extensively in statistical mechanics [1,6] and combinatorics (see, e.g., [5,7]) to evaluate and analyse many different functions defined on lattice graphs of various kinds. In statisti-

[*] Corresponding author.
E-mail addresses: Graham.Farr@infotech.monash.edu.au (G. Farr), academia@schmidt.net.au (J. Schmidt).
[1] The final stages of Schmidt's work were completed at NETEK Software Engineering, Oakleigh East, Victoria 3166, Australia.

0020-0190/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2007.08.018

cal mechanics, they have been used to study partition functions [1,6]. In graph theory, they have been used to count structures such as colourings [5] and self-avoiding walks [7].

There has been a number of attempts to estimate the number of legal Go positions on square lattice boards including the standard 19×19 one. A brief history of these is given in [15], where the first such estimate is attributed to Achim Flammenkamp. For the 19×19 board, he found using a computer program, doing random sampling, that about 1.2% of possible positions (with each point being empty, black or white) are legal, giving an estimate of $0.012 \times 3^{19^2} \simeq 2.1 \times 10^{170}$ legal positions. One aim of the present work was to obtain, by *deterministic* means, rigorous bounds for the numbers of legal positions on large boards. To this end, we have used transfer matrix methods. The design and implementation of algorithms for this, and the results obtained, are reported in Schmidt's dissertation [13]. The present paper describes the methods used and summarises the main results.

While preparing this paper, we learned of independent work by Tromp and Farnebäck [15]. They have obtained exact values for the number of legal positions on square boards of size up to 16×16, as well as determining the limiting behaviour of the number of legal positions on large rectangular boards of fixed width up to seven. They also find estimates for the number of legal *games* of Go. The methods used are also essentially based on transfer matrices, but with some extra techniques that enable computations to be done for larger boards, culminating in their very impressive computation for the 16×16 board. There is some overlap between the two papers, but areas of difference too. For example, in this paper we establish the existence of some important limits, and look at the different ways in which large boards can be divided into strips in order to obtain bounds for the numbers of legal positions. We also consider non-square lattice topologies and more than two players.

One of the early papers in the mathematics and computer science literature on Go was by Benson [2], who gave an essentially graph-theoretic characterisation of stones that could not possibly be captured. (Note that this differs from our focus, which is on stones that remain uncaptured as they stand, regardless of whether or not they may be able to be captured later in the game.) Benson's characterisation could be used as one component (among very many) of programs for playing Go. Such programs have been a subject of research in artificial intelligence [12]. Go has also been studied in complexity theory, where various computational problems associated with trying to win the game have been shown to be PSPACE-hard [11]. There is also research on Go using the theory of games developed by Berlekamp, Conway and others [4]. Although these various research efforts have been very fruitful, we do not discuss them further here as they are not directly related to the topic of this paper.

In the next section, we give definitions. In Section 3, we establish the existence of certain limits associated with numbers of Go positions on square lattice graphs of increasing size, answering a question raised in [15, § 4.9]. We describe our transfer matrix approach in detail in Section 4, and briefly describe adapting it to other lattice graphs in Section 5. Some details on our computations done using these methods are given in Section 6. The results obtained are presented in Section 7, and suggestions for further work are given in Section 8.

2. Definitions

Let $G = (V, E)$ be a graph. A *partial λ-assignment* is a function $f : V \to \{0, 1, \ldots, \lambda\}$. The values $1, \ldots, \lambda$ here are *colours*, and the value 0 represents absence of a colour, so that f is thought of as giving colours to some subset of V. Note that f does not have to be a proper colouring in the usual graph theoretic sense: it is permissible for adjacent vertices to receive the same colour. Each colour class induces a subgraph of G, and the connected components of these subgraphs are *groups*. (This term is not always used so precisely in writings on Go. For us, the term *group* is a synonym for *monochromatic component*.)

A partial λ-assignment is *free* if every group has a vertex that is adjacent to an uncoloured vertex. The legal Go positions, for λ players (each with her own colour) using G as the board, are precisely the free partial λ-assignments. The number of such positions (or assignments) is denoted by $\mathrm{Go}^{\#}(G; \lambda)$. This is one of the *Go polynomials* of [8]. In that paper, it was shown to be a polynomial in λ (for any given G), and that computation of it could be reduced to computation of chromatic polynomials of graphs constructed from G by local modifications. Unfortunately, exponentially many such modified graphs are used. A polynomial time algorithm for computing $\mathrm{Go}^{\#}(G; \lambda)$ for general graphs is unlikely to exist, since this Go polynomial is #P-hard to compute for all integers $\lambda \geqslant 2$ (proved in [8] using a result on linear forms in logarithms from transcendental number theory).

In this paper we are interested in much more constrained classes of graphs: planar lattice graphs of various kinds, especially square lattice graphs since it is on such a graph that Go is normally played. The $M \times N$ square lattice graph is denoted by $\mathrm{Sq}_{M,N}$. This has rows indexed by $\{0,\dots,M-1\}$ and columns indexed by $\{0,\dots,N-1\}$, with vertices corresponding to members of $\{0,\dots,M-1\} \times \{0,\dots,N-1\}$. If we were to add horizontal edges that "wrap around" from vertices on the left border of $\mathrm{Sq}_{M,N}$ to vertices on the right, we have the cylinder $\mathrm{Cyl}_{M,N}$.

We seek values and bounds for $\mathrm{Go}^{\#}(\mathrm{Sq}_{M,N};\lambda)$ for various values of M, N and λ. We are especially interested in the limiting behaviour of $\mathrm{Go}^{\#}(\mathrm{Sq}_{M,N};\lambda)$ as one or both of M, N tend to infinity. For brevity, we put

$$S_\lambda(M,N) := \mathrm{Go}^{\#}(\mathrm{Sq}_{M,N};\lambda), \tag{1}$$

$$C_\lambda(M,N) := \mathrm{Go}^{\#}(\mathrm{Cyl}_{M,N};\lambda),$$

$$s_\lambda(M) := \lim_{N\to\infty} S_\lambda(M,N)^{1/N}, \tag{2}$$

$$s_\lambda := \lim_{N\to\infty} S_\lambda(N,N)^{1/N^2}. \tag{3}$$

We discuss the existence of these limits in the next section. We may drop the subscript λ when it is clear from the context.

3. Existence of limits

We need a lemma due to Fekete. A function f is *log-superadditive* if $f(m+n) \geqslant f(m)f(n)$ for all $m,n \in \mathbb{N}$.

Lemma 1. *([9], see [17, Lemma 11.6, p. 85].) If a function $f:\mathbb{N}\to\mathbb{N}$ is log-superadditive and $f(n)^{1/n}$ is bounded above by a constant, then $\lim_{n\to\infty} f(n)^{1/n}$ exists and is finite.*

If we take any legal position on Sq_{M,N_1} and any legal position on Sq_{M,N_2}, we can form a position on Sq_{M,N_1+N_2} by placing the first two positions next to each other in an obvious way, adding horizontal edges to "sew" them together. The resulting position will be legal on Sq_{M,N_1+N_2}, and every legal position on Sq_{M,N_1+N_2} can arise from at most one (and sometimes no) such construction. It follows that $S_\lambda(M,N)$ is log-superadditive, as a function of N.

It is clear that $S_\lambda(M,N) \leqslant (\lambda+1)^{MN}$. Therefore, for fixed M, the quantity $S_\lambda(M,N)^{1/N}$ (as a function of N) is bounded above by the constant $(\lambda+1)^M$.

It follows from Fekete's Lemma that, for all M, the limit $s_\lambda(M)$ (defined in (2)) exists.

Now consider $S_\lambda(N,N)$ as $N\to\infty$. It is straightforward to show, using an argument similar in style to that used above (and extra vacant vertices if necessary to fill out the shape of the final board), that

$$S_\lambda(N+L,N+L) \geqslant S_\lambda(N,N)\cdot S_\lambda(L,L)^{2\lfloor N/L\rfloor+1}.$$

The upper bound $S_\lambda(N,N)^{1/N^2} \leqslant \lambda+1$ is easy, as above. We do not have log-superadditivity, but a variant of it. It is routine to adapt the proof of Fekete's Lemma to deal with this situation, and so to deduce that the limit s_λ (defined in (3)) exists.

Finally, consider $s_\lambda(N)^{1/N}$, as a function of N, and in particular its limit (if any) as $N\to\infty$. The existence of an upper bound is again easy. It is also easy to establish log-superadditivity, via the log-superadditivity (in N) of $S_\lambda(M,N)$ and taking appropriate limits. Hence $\lim_{N\to\infty} s_\lambda(N)^{1/N}$ exists. It remains to determine whether it is also s_λ.

By dividing $\mathrm{Sq}_{N,N}$ into $\lfloor N/M\rfloor$ copies of $\mathrm{Sq}_{M,N}$ (and a few empty rows as necessary), it is easy to see that $S_\lambda(N,N) \geqslant S_\lambda(M,N)^{\lfloor N/M\rfloor}$. It follows (using elementary limit arguments) that $s_\lambda \geqslant s_\lambda(M)^{1/M}$, and hence that $s_\lambda \geqslant \lim_{N\to\infty} s_\lambda(N)^{1/N}$. Conversely, by dividing $\mathrm{Sq}_{M,N}$ into $\lfloor N/M\rfloor$ copies of $\mathrm{Sq}_{M,M}$ (and some empty columns as required), we can show by similar methods that $s_\lambda \leqslant \lim_{N\to\infty} s_\lambda(N)^{1/N}$, and hence that the two limits are equal.

Regarding the graphs $\mathrm{Cyl}_{M,N}$, observe that $\mathrm{Sq}_{M,N} \leqslant \mathrm{Cyl}_{M,N}$, so $S_\lambda(M,N) \leqslant C_\lambda(M,N)$. On the other hand, any legal position on $\mathrm{Cyl}_{M,N}$ yields a legal position on $\mathrm{Sq}_{M,N+2}$: starting with the former, just delete the edges between column 0 and column $N-1$, forming a partial λ-assignment on $\mathrm{Sq}_{M,N}$, and add empty columns to the left and right sides of the board, ensuring a legal position on $\mathrm{Sq}_{M,N+2}$. Therefore $C_\lambda(M,N) \leqslant S_\lambda(M,N+2)$. These inequalities for $C_\lambda(M,N)$ imply that, as $N\to\infty$, $C_\lambda(M,N)^{1/N}$ converges to the same limit, $s_\lambda(M)$, as $S_\lambda(M,N)^{1/N}$ does. Similar reasoning establishes that $\lim_{N\to\infty} C_\lambda(N,N)^{1/N^2} = s_\lambda$.

In summary:

Theorem 2. *The limits $s_\lambda(M)$ (for all M) and s_λ exist, and*

$$\lim_{M\to\infty} s_\lambda(M)^{1/M} = s_\lambda.$$

The limits $\lim_{N\to\infty} C_\lambda(M,N)^{1/M}$ (for all M) and $\lim_{N\to\infty} C_\lambda(N,N)^{1/N^2}$ also exist, and equal $s_\lambda(M)$ and s_λ, respectively.

4. Transfer matrix method

4.1. Column configurations and position sections

We view $\mathrm{Sq}_{M,N}$ as being built up column by column, so $\mathrm{Sq}_{M,N}$ is obtained by adding column $N-1$ (consisting of a vertical path of M vertices) to the right of $\mathrm{Sq}_{M,N-1}$, and adding edges $\{(i,N-2),(i,N-1)\}$

for each i, $0 \leqslant i \leqslant M-1$. A Go position on $\mathrm{Sq}_{M,N}$ is also regarded as being built up, column by column, in the same way.

In a legal position, each vertex is in one of $2\lambda + 1$ *states*, as follows. Firstly, it may be uncoloured, or *empty*. Next, let c be any colour ($1 \leqslant c \leqslant \lambda$). A vertex $v = (i, j)$ with colour c is in state *c-safe* if its group has an empty neighbour in at least one of columns $0, \dots, i$ (which together form $\mathrm{Sq}_{M,i+1}$). Otherwise, v is in state *c-not-yet–safe* (abbreviated *c-NYS*).

A *column configuration* is an assignment of states to the vertices of a single column of M vertices. A *legal column configuration* (*LCC*) is a column configuration such that, for each colour c, no vertex in state c-NYS is adjacent to any vertex in state c-safe or to any empty vertex. A column configuration is an LCC if and only if it can occur in a single column of some legal Go position on $\mathrm{Sq}_{M,N}$. It is easily verified that, if $\lambda = 2$, there are five LCCs if $M = 1$ and 17 LCCs if $M = 2$.

A *segment* of an LCC is a maximal path of consecutive nonempty vertices in the column all of which have the same state. In any given position, two different segments of the same state in the same column may or may not be part of the same group, depending on the rest of the position. It is important to be able to keep track of this group structure, when the column is part of some larger position.

A *position section* is an LCC with additional information to record how the segments are to be connected up in groups. This information specifies, for each pair of segments, whether the segments belong to the same group or not. A natural way to do this is to give each segment a group number, in some canonical way, so that segments are in the same group if and only if they have the same group number. The group numbers must be consistent with the planarity of $\mathrm{Sq}_{M,N}$, so that if we have segments s_1, s_2, s_3, s_4 going down the column (possibly with other segments and empty vertices intervening), and s_1 and s_3 have the same group number, then s_2 and s_4 cannot have the same group number as each other unless all four segments have the same group number.

Write $\mathrm{PS}(M) = \mathrm{PS}_\lambda(M)$ for the set of position sections. Given a legal Go position on $\mathrm{Sq}_{M,N}$, its jth *section* ($j \geqslant 0$) is the position section formed from column j of that position.

4.2. The transfer matrix and its eigenvalues

Two position sections s_1 and s_2 are *compatible* if there is a legal Go position on $\mathrm{Sq}_{M,N}$ in which s_1 and s_2 are the jth and $(j+1)$th sections, respectively, for some $j \geqslant 0$. (Note that compatibility is not a symmetric relation.) Compatibility is easily testable just by examining the two position sections themselves; there is no need to search among positions on larger boards. We put

$$t_{s_2,s_1} = \begin{cases} 1, & \text{if } s_1 \text{ and } s_2 \text{ are compatible;} \\ 0, & \text{otherwise.} \end{cases}$$

The *transfer matrix* $T = T_M$ is the square matrix with rows and columns indexed by $\mathrm{PS}(M)$ and entries t_{s_1,s_2}.

For each $j \geqslant 0$ and each position section s, let x_{js} be the number of legal positions on $\mathrm{Sq}_{M,j+2}$ in which the jth section is s and column $j+1$ consists entirely of empty vertices. These empty vertices just ensure that all groups with NYS vertices in column $j-1$ are uncaptured. Observe that x_{0s} will be 1 if s has all its segments in different groups and all safe segments have an empty neighbour (either immediately above or below them, or both); otherwise, it is 0. Set the column vector $\mathbf{x}_j := (x_{js})_{s \in \mathrm{PS}(M)}$.

If we consider the contributions to $x_{(j+1),s}$ due to each of the possible position sections in column $(j-1)$, we find

$$x_{(j+1),s} = \sum_{s'} t_{s,s'} x_{j,s'}.$$

It follows that

$$\mathbf{x}_{j+1} = T\mathbf{x}_j,$$
$$\mathbf{x}_{j+n} = T^n \mathbf{x}_j.$$

In order to work out the number of legal positions when there is no final column of empty vertices adjoined, we use the vector $\mathbf{f} = (f_s)_{s \in \mathrm{PS}(M)}$, where $f_s = 1$ if s has no NYS vertices, and $f_s = 0$ otherwise. The number of legal positions in $\mathrm{Sq}_{M,N}$ is then given by

$$S(M,N) = \mathbf{f}'\mathbf{x}_{N-1} = \mathbf{f}'T^{N-1}\mathbf{x}_0. \tag{4}$$

It can be seen that the s', s-entry of T_M^n is the number of legal Go positions in $\mathrm{Sq}_{M,n+2}$ with 0th and nth position sections s and s', respectively, and column $n+1$ empty. Therefore the diagonal s, s-entry gives the number of legal positions with 0th section s in the graph formed by identifying columns 0 and n in $\mathrm{Sq}_{M,n+2}$ and leaving column $n+1$ empty. This graph is the cylinder $\mathrm{Cyl}_{M,n}$ plus those extra empty neighbours of the vertices in column 0: we denote it by $\mathrm{Cyl}'_{M,n}$.

If we sum the diagonal entries of the transfer matrix, obtaining its trace $\mathrm{tr}(T_M)$, then we obtain the total number of legal positions for $\mathrm{Cyl}'_{M,n}$, which we denote by $C'(M,n) = C'_\lambda(M,n)$. It is clear that $C(M,n) \leqslant C'(M,n)$. Furthermore, $\mathrm{Cyl}'_{M,n}$ can be obtained from $\mathrm{Cyl}_{M,n+2}$ by identifying columns $n-1$ and $n+1$, and adding the restriction that the vertices in the original column n must remain empty. It follows that $C'(M,n) \leqslant C(M,n+2)$. Using these inequalities for $C'(M,n)$, we have

$$\lim_{n\to\infty} \operatorname{tr}\bigl(T_M^n\bigr)^{1/n} = \lim_{n\to\infty} C'(M,n)^{1/n} = s(M).$$

From standard transfer matrix theory (see, e.g., [1]), this limit is the size of the largest eigenvalue of T_M.

We have calculated these transfer matrices, and their largest eigenvalues, for sizes $M \leqslant 6$. The results can be found in Section 7. The dimension of T_M grows exponentially with M, so the amount of space required to store T_M soon becomes prohibitive. For large M, we divide the board $\mathrm{Sq}_{M,N}$ into horizontal strips, each of size $\leqslant 6$, in order to obtain bounds for $S(M,N)$, and ultimately for the limiting constant s_λ. We now give the details of how we do this.

4.3. Obtaining bounds for large boards

Suppose $k_1 + \cdots + k_m = M$, and suppose the board $\mathrm{Sq}_{M,N}$ is divided into horizontal strips $\mathrm{Sq}_{k_i,N}$, $1 \leqslant i \leqslant m$, each of these being a subgraph of $\mathrm{Sq}_{M,N}$. Any m-tuple $(P_i)_{i=1}^m$, where P_i is a legal position for $\mathrm{Sq}_{k_i,N}$ for each i, yields a legal position for $\mathrm{Sq}_{M,N}$, and the mapping is one-to-one (but not onto). Hence $S(M,N) \leqslant \prod_{i=1}^m S(k_i,N)$.

Now let P be any legal position for $\mathrm{Sq}_{M,N}$, and for each i let P_i be the partial λ-assignment it induces on the ith strip, $\mathrm{Sq}_{k_i,N}$. This P_i is not necessarily a legal position on $\mathrm{Sq}_{k_i,N}$, but it certainly becomes legal if a row of empty vertices is added along the top and bottom of $\mathrm{Sq}_{k_i,N}$. In fact, for $i=1$ and $i=m$, it is necessary only to add one row of empty vertices: one of the borders of $\mathrm{Sq}_{k_1,N}$ is also a border of $\mathrm{Sq}_{M,N}$, so we only add the empty row to the other border of $\mathrm{Sq}_{k_1,N}$ (and analogously for $\mathrm{Sq}_{k_m,N}$). For $b=1,2$, we write $S^{(b)}(k,N) = S_\lambda^{(b)}(k,N)$ for the number of legal positions on the modified board $\mathrm{Sq}_{k,N}$ with b extra empty rows added (one at top and one at bottom, if $b=2$). We find that

$$S(M,N) \geqslant S^{(1)}(k_1,N)S^{(1)}(k_m,N) \prod_{i=2}^{m-1} S^{(2)}(k_i,N).$$

The techniques of Section 3 are readily used to show that $u_\lambda^{(b)}(k) := \lim_{N\to\infty} S^{(b)}(k,N)^{1/N}$ exists, and furthermore that $\lim_{k\to\infty} u_\lambda^{(b)}(k)^{1/k} = s_\lambda$. It is easily seen that $u_\lambda^{(b)}(k) \geqslant s_\lambda(k)$.

The transfer matrix methods developed above can also be used for these modified boards with b extra empty rows at top and/or bottom. The set of position sections, and the resulting transfer matrices $T_k^{(b)}$, and their eigenvalues, are all different, but the underlying theory is the same. The above-mentioned limiting constant $u_\lambda^{(b)}(k)$ is just the size of the largest eigenvalue of $T_k^{(b)}$.

These methods allow one to find the following bounds on our main limiting constant:

$$s_\lambda(k)^{1/k} \leqslant s_\lambda \leqslant u_\lambda^{(2)}(k)^{1/k}. \tag{5}$$

To get the tightest bound we can, we make k as large as we can, given the available memory. The maximum feasible k decreases as λ increases. For $\lambda = 2$ we use $k=6$.

5. Other lattices

We have also applied these transfer matrix methods to a few other lattices: triangular, hexagonal and square-octagonal lattices. We view these as being constructed from square lattices as follows. For the triangular lattice, take $\mathrm{Sq}(M,N)$ and add a diagonal edge to each square face, with all these diagonal edges aligned the same way. For the hexagonal lattice, half the horizontal edges of $\mathrm{Sq}(M,N)$ are deleted: those edges $\{(i,j),(i+1,j)\}$ for which $i+j$ is odd. For the square-octagonal lattice, again, half the horizontal edges of $\mathrm{Sq}(M,N)$ are deleted: this time, we delete edges $\{(i,j),(i+1,j)\}$ for which $\lfloor (i+1)/2 \rfloor + j$ is odd.

Our results for values and bounds for numbers of Go positions on some of these lattices, and their limiting constants, are given in Section 7.

6. Computations

We developed software that, for a given column size M, number of colours λ, and lattice type (square, or see Section 5), generates position sections and uses compatibility testing to generate transfer matrices. The software was written in C and run on PowerPC 970 based computers running Mac OS X 10.4. Various output options are available, but the main form of output is in the form of scripts for the MATLAB software package [14]. These scripts, when run in MATLAB, result in a session that contains the transfer matrix T, initial vector $\mathbf{x}_0$ and vector $\mathbf{f}$, as discussed in Section 4.2. Results can then be calculated directly in MATLAB. Note that the numbers given in our tables of results in Section 7 are floating point approximations exactly as produced by MATLAB. Further information on the software, including source code, is in [13].

7. Results

The number of legal positions for square Go boards up to size 16 is known [15]. Table 1 shows our bounds for the remaining square Go boards, up to the standard 19×19 board. Observe that Flammenkamp's estimate $S(19, 19) \approx 2.1 \times 10^{170}$ lies between the given bounds for $N = 19$, with the upper bound being closer (by ratio) than the lower bound.

Table 1
Bounds on # legal positions for $N \times N$ Go boards with 2 colours

N	Lower bound	Upper bound
17	$4.189686102798248 \times 10^{135}$	$3.795846030704812 \times 10^{136}$
18	$1.370980901968463 \times 10^{152}$	$1.375868183851593 \times 10^{153}$
19	$1.743685286536042 \times 10^{169}$	$6.432156129669607 \times 10^{170}$

Table 2
Bounds on c for boards of different lattices with 2 colours

Lattice	Lower bound	Upper bound
Square	2.958390334214091	2.983702175733463
Triangular	2.987746640147263	2.997980185636975
Hexagonal	2.873504100894481	2.950174880917301
Square-octagonal	2.872141227144343	2.949445944086531

Table 3
Bounds on c for square-lattice boards with λ colours, based on size-k strips

λ	Lower bound	Upper bound	k
2	2.958390334214091	2.983702175733463	6
3	3.74973384385035	3.91177363494727	4
4	4.40249502487144	4.83400900456086	3
5	5.01838431429539	5.68828042421368	3
6	5.59033171955122	6.51092688361270	3

To find the bounds shown in Table 1, various strip arrangements were tried, with the methods of Section 4.3 being used to obtain upper and lower bounds for each (and the number of positions in each strip being computed using (4)). For each value of N, the tightest bounds were chosen. It is interesting to note that evenly sized strips do not necessarily offer the best results. For example, when calculating bounds for a board of height 8, using one strip of size 5 and another of size 3 produces better bounds than using two strips of size 4. Also, for a single N, the best upper and lower bounds are not always provided by the same strip arrangement. For example, for a board of height 19, a strip arrangement 6, 4, 3, 6 provides the best lower bound, while 6, 4, 4, 5 gives the best upper bound.

Table 2 shows bounds on the limiting constant, c, calculated using (5) and strip size $k = 6$ (with appropriate adjustments for non-square boards). The order on these lattices given by their edge densities (triangular > square > hexagonal = square-octagonal) corresponds closely to that given by their limiting constants (given what these bounds tell us). We do not know which of the hexagonal and square-octagonal lattices has the greater limiting constant, or indeed whether they are unequal.

Table 4
# Legal positions for $k \times 100$ strips with λ colours

k	λ	NBC	1SBC	2SBC
1	2	$1.197615353980522 \times 10^{44}$	$5.153775207320113 \times 10^{47}$	$5.153775207320113 \times 10^{47}$
1	3	$2.34477618645844 \times 10^{52}$	$1.60693804425899 \times 10^{60}$	$1.60693804425899 \times 10^{60}$
1	4	$2.862770775344951 \times 10^{58}$	$7.888609052210114 \times 10^{69}$	$7.888609052210114 \times 10^{69}$
1	5	$2.027190260863544 \times 10^{63}$	$6.533186235000708 \times 10^{77}$	$6.533186235000708 \times 10^{77}$
1	6	$2.262096163990697 \times 10^{67}$	$3.234476509624757 \times 10^{84}$	$3.234476509624757 \times 10^{84}$
2	2	$1.007837024457852 \times 10^{93}$	$1.901030015192073 \times 10^{94}$	$2.656139888758749 \times 10^{95}$
2	3	$5.860438312661663 \times 10^{112}$	$2.21246453296292 \times 10^{116}$	$2.582249878086909 \times 10^{120}$
2	4	$7.9241768005973 \times 10^{126}$	$3.360317624581033 \times 10^{132}$	$6.223015277861129 \times 10^{139}$
2	5	$9.412149214131599 \times 10^{137}$	$2.304759172522902 \times 10^{145}$	$4.268252238120267 \times 10^{155}$
2	6	$1.321449838513509 \times 10^{147}$	$1.098994470510193 \times 10^{156}$	$1.046183829131436 \times 10^{169}$
3	2	$2.78627066316203 \times 10^{140}$	$3.954538651617351 \times 10^{141}$	$5.772518963667382 \times 10^{142}$
3	3	$1.158452405125824 \times 10^{171}$	$5.709920913833765 \times 10^{174}$	$3.621564683124062 \times 10^{178}$
3	4	$7.054118968360604 \times 10^{192}$	$9.644948950610727 \times 10^{198}$	$1.673346874227885 \times 10^{205}$
3	5	$7.426259922699532 \times 10^{209}$	$1.40471492942176 \times 10^{218}$	$2.540105154607062 \times 10^{226}$
3	6	$8.03000875084465 \times 10^{223}$	$1.116667924407958 \times 10^{234}$	$9.684656333932822 \times 10^{243}$
4	2	$6.081595154852431 \times 10^{187}$	$8.784730272847668 \times 10^{188}$	$1.271837895634509 \times 10^{190}$
4	3	$2.241163399819416 \times 10^{229}$	$1.304875387323153 \times 10^{233}$	$7.337376961041674 \times 10^{236}$
5	2	$1.341139245616684 \times 10^{235}$	$1.941095386001647 \times 10^{236}$	$2.80945174721662 \times 10^{237}$
6	2	$2.962357293902291 \times 10^{282}$	$4.287697761411524 \times 10^{283}$	$6.205926126049495 \times 10^{284}$

Table 3 shows bounds on the limiting constant, c, for square-lattice boards of λ colours. Due to computing limitations, higher λ required smaller strip sizes. The strip size used for a given λ is shown in the last column of the table.

Table 4 shows the number of legal positions for $k \times 100$ boards (for $1 \leqslant k \leqslant 6$) with λ colours, with the different boundary conditions. *NBC* denotes normal boundary conditions, *1SBC* denotes single-safe boundary conditions and *2SBC* denotes double-safe boundary conditions. This table illustrates the kind of information used in applying the methods of Section 4.3, since it includes numbers of legal positions for all strips we would use to find bounds for $S_\lambda(M, 100)$, for $\lambda \leqslant 6$ and any M.

8. Further work

Further problems suggested by this work include: determine (with rigorous proof) the exact value of the constant s_λ; similarly, determine analogous constants for other infinite lattices.

Many of the computational problems discussed in this paper take just a single integer as input (e.g., determination of $S_\lambda(N, N)$, or of $S_\lambda(k, N)$, for fixed λ and k). If N is given in unary, then these problems are in Valiant's class $\#P_1$, from [16]. They join a long line of problems whose complexity within that class is not well understood. Any result establishing some reasonable kind of hardness of (e.g.) $S_\lambda(N, N)$ within that class would be very significant, if indeed such is possible. Failing that, efficient reductions to or from some better known problems in that class (e.g., counting 3-colourings of $\mathrm{Sq}_{N,N}$) would be interesting.

Acknowledgements

We thank Keith Edwards for suggesting the use of transfer matrix methods for counting legal Go positions.

References

[1] R.J. Baxter, Exactly Solved Models in Statistical Mechanics, Academic Press, London, 1982.
[2] D.B. Benson, Life in the game of Go, Inform. Sci. 10 (1976) 17–29.
[3] British Go Association, Go: The Most Challenging Board Game in the World, British Go Association, London, 1999. Available at: http://www.britgo.org/intro/booklet.pdf (accessed 23 May 2005).
[4] E.R. Berlekamp, D. Wolfe, Mathematical Go: Chilling Gets the Last Point, A.K. Peters, Wellesley, MA, 1994.
[5] N.L. Biggs, Colouring square lattice graphs, Bull. London Math. Soc. 9 (1977) 54–56.
[6] B.A. Cipra, An introduction to the Ising model, Amer. Math. Monthly 94 (1987) 937–959.
[7] K.J. Edwards, Counting self-avoiding walks in a bounded region, Ars Combin. 20B (1985) 271–281.
[8] G.E. Farr, The Go polynomials of a graph, Theoret. Comput. Sci. 306 (2003) 1–18.
[9] M. Fekete, Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten, Math. Zeitschr. 17 (1923) 228–249.
[10] K. Iwamoto, Go for Beginners, Ishi Press, 1972, and Penguin Books, 1976.
[11] D. Lichtenstein, M. Sipser, GO is polynomial-space hard, J. Assoc. Comput. Mach. 27 (1980) 393–401.
[12] M. Müller, Computer Go, Artificial Intelligence 134 (2002) 145–179.
[13] J. Schmidt, Counting legal positions in Go, BSE Hons dissertation, Clayton School of Information Technology, Monash University, December 2005; Available at: http://www.csse.monash.edu.au/hons/se-projects/2005/Johannes.Schmidt/ (accessed 6 December 2005).
[14] The MathWorks Inc., MATLAB (online), 2005. Available at: http://www.mathworks.com/access/helpdesk/help/techdoc/learn_matlab/ (accessed 1 December 2005).
[15] J. Tromp, G. Farnebäck, Combinatorics of Go, preprint, 11 October 2005; Available at: http://homepages.cwi.nl/~tromp/go/gostate.ps (accessed 10 December 2005).
[16] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (1979) 410–421.
[17] J.H. van Lint, R.M. Wilson, A Course in Combinatorics, Cambridge, 1992.
[18] D.J.A. Welsh, Complexity: Knots, Colourings and Counting, London Math. Soc. Lecture Note Series, vol. 186, Cambridge University Press, 1993.