Go Combinatorics: The Recent Work of Dr. John Tromp and His Colleagues on the Number of Possible Go Positions, Games and their Length

By Peter Shotwell

© July 2010

This is a lightly edited version of one of the updates of my first book, Go! MoreThan a Game, *(Tuttle) that is being re-printed in 2010. After the article was finished, Doctor Tromp kindly responded to an interview request for this e-version. (Other advances in the world of go since 2003 that will appear in the book include new work on combinatorial game theory, Monte Carlo computer go, and the earliest Confucian go literature).*

Another fairyland in the world of go has recently been created by Dutch computer scientist John Tromp and his colleagues. Their work on combinatorics—the science of counting—join Monte Carlo computer methods and Elwyn Berlekamp and Bill Spight's combinatorial game theory in leaving little left of our usual thoughts about go.

For example, Dr. Tromp wrote in a private communication,

The notion of eyes only arises in sensible play. For our results, forget all that you know about go strategy and sensible play; we simply look at all possible move sequences. . . .

The result is that most of the games are similar to the Stone Counting Method that I used in *Beginning Go,* my third teaching book (aimed at children), to explain the game on a small board. In my version, the players can keep putting down stones inside of their territories until the loser has only two eyes left, while the winner has extra empty intersections besides two eyes and hence can play more stones. However, in Tromp's version, the game doesn't end at that point—the "loser" can fill in one eye and the game doesn't stop just because there are no stones left after everything is

taken off by the opponent on the next move. They keep playing and can pass at any time, which counts for a move if the opponent moves. Eventually, however, the Superko rule results in two passes that end a game.

The result is that the longest possible Go game has over 10^48 moves (1 followed by 48 zeros).

So who won?

At the end of the longest game, all moves lead to repetition of some earlier position. The board at this point may look like complete chaos however, with many groups in atari*. To ask who won is to miss the point. The scoring method is of no concern to counting games. You could even say the winner is the one who made the fewest passes; it doesn't change the number of possible move sequences. So we don't even specify any scoring method. It is of no concern to us.*

However, this game might run into time limit problems. According to one theory—the Big Freeze—if gravity fails to hold the universe together, there will only be another 1000 trillion years before all matter disappears. After the stars run out of fuel, all protons will plunge into Black Holes and evaporate, and all that will be left are electrons, positrons, and neutrinos floating aimlessly around, ready for the next stage, whatever that might be. With about 32,000,000 seconds in a year, the players would have to move very, very fast, since there would just be thirty-one sextillion seconds (32 followed by only 21 zeros) left to play in.

Since two passes at any point ends the games, the shortest possible game is 0 moves: Black passes, White passes (no resignations are permitted in these calculations). The next shortest is Black plays, White passes, Black passes. The thought that there are 361 possible positions in a 1-move game on a 19x19 board leads into the question of the number of possible Go positions on various sized boards.

On a 2x2 board, 57 out of 81 positions are 'legal'—that is, roughly 70% contain no groups without liberties:
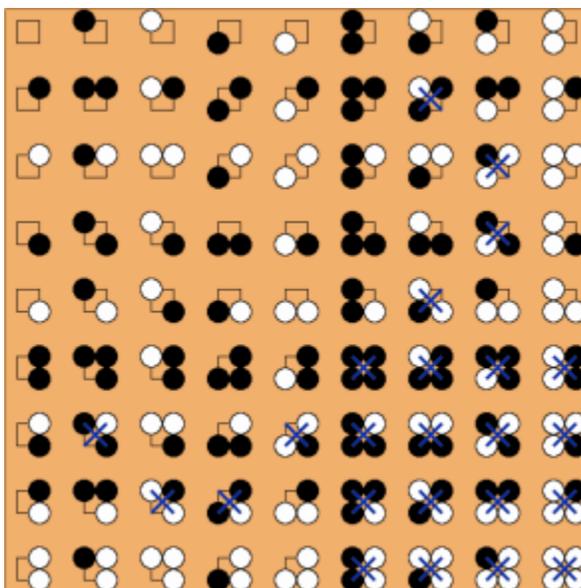
The illegal positions are X'd out

However, on 19x19 boards, only about 1.196% are legal—"about" because, constrained by resources, Dr. Tromp and colleagues have arrived at "only" 12 digits of precision.

It turns out that the old adage that "there are more possible Go games than there are atoms in the universe" vastly understates the situation. $3^{361} \times 0.011957528698 = 2.081681994 \times 10^{170}$ possible *positions*, (let alone games), can appear on a 19x19 boards and the number of *particles* in the universe is only 10^{80}.

To make dramatic comparisons with these figures and those for chess, as some have tried to do, is "nonsensical," according to Dr. Tromp.

What we do know is that the number of Go positions is many, many, many orders of magnitude larger, a 171 digit number instead of less than than a 46 digit number for chess. The number is inexact and that is an active area of research for me.

"Dramatic comparisons" can only show so many orders of magnitude in difference. Even comparing the size of the universe with the size of the nucleus of an atom, that's only a factor 10^{41}, or 41 orders of magnitude. As you can see, the 125 orders of magnitude difference that Go positions have over chess positions dwarfs that.

But when we start configuring the number of legal Go games, the numbers become even more stupefying. Despite the fact that there are only

57 legal positions on a 2x2 board, there are 386,356,909,593 possible games! And on 19x19 boards, the lower bound is "only" 10^(10^48), while the upper bound is a whopping 10^(10^171).

With the number of possible games, the situation with chess vs. go is incomparably worse. Why incomparably? Because even the number of digits in the number of 19x19 Go games is larger than the number of chess games. You really have to let that sink in. . . .

Dr. Tromp's preliminary 33 page paper is on his website www.cwi.nl/~tromp, where the huge numbers for all board sizes are listed. I asked him when he is going to publish. "We are up to exact figures for 17x17 and it's not final until I can do the computation for 19x19. :-)"
When he finishes, will this work have any applications or uses? "None that I can think of . . ."

* * * * *

Can you give readers an idea of your background?

My background is in theoretical computer science, which is like the mathematical underpinnings of computation. We study different models of computation, their relation, and the amount of resources (mainly time and space) needed to solve problems. To that end we study algorithms and data structures.

How did you get interested in go combinatorics and could you give an idea of the methods you used?

When I got my first home computer (a Sinclair ZX Spectrum 48K) in the early 80s, I not only took an interest in playing games, but also in programming them. After typing in a connect-4 BASIC program from a magazine, I tried making it stronger, eventually turning to Z80 assembly language.
When I needed a graduation topic near the end of my Computer Science studies at the University of Amsterdam, I thought of making an 'unbeatable' connect-4 program and asked around about previous efforts. I then found out that Victor Allis was in the last stages of actually solving the game at the Free University (also in Amsterdam). So instead I went on to write an M.Sc. thesis on wait-free shared register constructions, although I

did proceed to build a database of all solved 8-ply connect-4 positions as a hobby.

Although I've long been interested in the game of go, it only entered the focus of my research around 2000, when I joined Marcel Crasmaru in studying the computational complexity of ladders in go, which turn out to be very hard in general. rec.games.go had become one of my favorite newsgroups, and brought me into contact with several people like Bill Taylor and Ted Drange whom I joined in solving go on small board sizes. From exhaustively analyzing small boards like 2x2 to trying to count the number of possible games was a small step. Parts of these early discussions can be read at

http://groups.google.com/group/rec.games.go/browse_thread/thread/161ff6e5922e1124/c90e5b4a61ea0602?lnk=st.

Meanwhile, on the newsgroup the question of what percentage of positions is legal had surfaced, and I enjoyed writing a tiny program that could check the legality of a position as it was being randomly generated. It confirmed other people's estimates of their being about 1.2% legal positions for 19x19. This led some to push the boundaries of counting the exact number of legal positions by brute force, but this became rather infeasible around 5x5.

Several people then suggested ways of overcoming this barrier by means of dynamic programming techniques, in various levels of detail. Once implemented, the boundaries were quickly pushed forward again, to around 10x10.

The idea of dynamic programming was to count legal partial boards, such as 3x19 or 4x19 boards, or 3x19+4 boards, (which means the fourth row only has 4 points). And rather than enumerating all possible partial boards, we enumerate all possible boundary states, where the boundary state includes not just the color (white, black or empty) of points along the boundary, but also information about which stones have liberties in the partial board, and how same colored stones on the boundary are connected.

While the number of partial boards is exponential in board area, the number of border states is only exponential in board width. This is what makes the problem feasible for boards larger than 5x5.

Another important ingredient of our results was a very concise encoding of border states, using only 3 bits per point, which comes to 57 bits for N=19, fitting comfortably in a machine register. This was my invention.

We also make use of something called the Chinese Remainder Theorem. To identify any number up to 1000, it suffices to know its remainders when divided by 7, 11, or 13. For instance, if a number between 0 and 1000 is 3 mod 7 (meaning 3 more than a multiple of 7), and 2 mod 11, and 6 mod 13, then it must be 409. This is also easy to compute. The use of CRT for go counting was suggested by Michal Koucky.

We similarly count the number of go positions modulo several big prime numbers, which still fit in a machine register, and combine them in the end. This allows for more efficient use of memory and disk space.

For analyzing the asymptotic behavior of the number of legal positions, we made good use of the fact that the logarithm of this function is super-additive: if you take four legal positions, and join them together in a square, then the resulting position is also legal. This led us to an independently discovered generalization of Fekete's lemma, which in a sense guarantees convergence.

We also establish, for the first time, double exponential upper and lower bounds on the number of possible games.

This article will be updated and an announcement will be made in the AGA e-Journal when Dr. Tromp and his colleagues work is completed.