A Googolplex of Go Games

Matthieu Walraet John Tromp

February 11, 2017

Abstract

We establish the existence of $10^{10^{100}}$ Go games, addressing an open problem in "Combinatorics of Go" [2].

This is a preprint version. The final publication is available at Springer via `http://dx.doi.org/10.1007/978-3-319-50935-8_18`

1 Introduction

The board game of Go is well known for its combination of simple rules [3] and profound complexity. That complexity is in part due to the large boardsize, allowing for long games and hundreds of choices at every turn. Popular estimates for the number of games are based on the average length of human games, typically around 300 moves. In practice, the board gets more and more filled, until the Black and White areas are sufficiently well-defined to allow for scoring the game. Estimates on the number of 'practical' $n \times n$ games take the form b^l where b and l are estimates on the number of choices per turn (branching factor) and game length, respectively. A reasonable and minimally-arbitrary upper bound sets $b = l = n^2$, while for a lower bound, values of $b = n$ and $l = \frac{2}{3}n^2$ seem both reasonable and not too arbitrary. This gives us bounds for the ill-defined number P19 of 'practical' 19x19 games of

$$10^{306} < P19 < 10^{924}$$

Wikipedia's page on Game complexity[5] combines a somewhat high estimate of $b = 250$ with an unreasonably low estime of $l = 150$ to arrive at a not unreasonable 10^{360} games.

But the rules also allow for less sensible games where players fill in their eyes and continue capturing each other, restricted only by the superko rule that forbids repeating the whole board position. It is this precisely defined set of all possible games that we want to count.

Let's denote by $N(n)$ the number of Go games on an $n \times n$ board using the rules of [3]. Tromp and Farnebäck[2] established

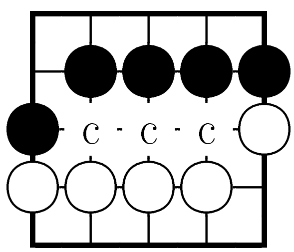
$$10^{10^{48}} < N(19) < 10^{10^{171}},$$

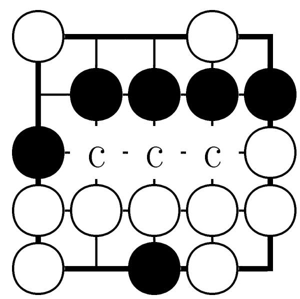
and list this rather huge gap as one of the open problems.

The challenge in proving a lower bound is to make a single game as long as possible, by visiting as many of the roughly $2 \cdot 10^{170}$ legal positions as possible. There will then turn out to be enough choices along the way to lift the game length into the exponent.

While [2] used properties of binary Grey codes to prove their lower bound, we obtain much stronger bounds by subdividing the board and iterating over all legal sub-board positions.

2 Basic scheme





For the 5×5 board on the left, consider the 25 points in row-major order from top left to bottom right. The central 3 control points marked 'c' split the board into two other symmetric sub-boards: the 11 point top and 11 point bottom.

The 5 points of the top directly preceding the control are reserved for a black border; which means they're either black or empty. The rest of the top, consisting of 6 points, can be anything that forms a legal position in combination with the black border.

Definition 1 *For some odd boardsize* $n \geq 3$, *a* legal *top position is a position on* $\frac{n^2-3}{2}$ *points, ending in* n *black border points, that is legal on the sub-board. It is called* pseudo-legal *if the position is legal on the sub-board plus an empty control. Let* H_n *be the number of legal sub-board positions (H for half).*

We have computed and manually verified that $H_5 = 323$. Similarly, the 5 points following the control are reserved for a white border, which allow for 323 corresponding legal bottom positions, which can be defined analogously. The right diagram shows a pair of legal sub-board positions.

In the basic scheme, we alternate setting up top and bottom positions, using the control to mark the different phases as follows:

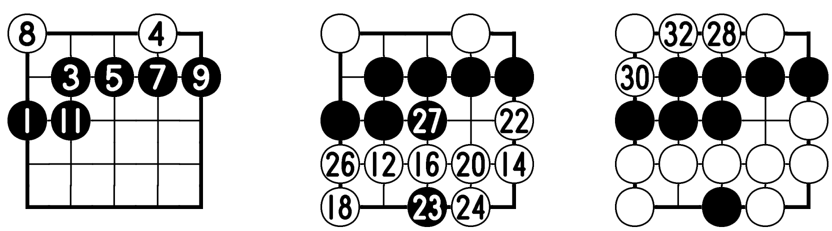
control state	top sub-board	bottom sub-board
+ + +	setup	
+ + ○	setup	frozen
+ ○ ○	frozen	fillup
● + +	frozen	setup
● ● +	fillup	frozen

To set up a position, pick any permutation of its stones (non-empty points), and play them in that order, passing in between if necessary. To complete a position setup, let's say in the black-bordered top, white plays in the control center. This "freezes" the top position, and moves the scheme into the next fill phase.

To fill up a position, say on top, first grow all white strings until they have only one liberty, possibly capturing black stones in the process. Note that the black border string is safe from capture because of its liberty in the control. Then, play black stones in any order until they fill the whole sub-board, capturing all expanded white strings in the process. The fillup phase is completed by a white play on that liberty, capturing the entire black block and clearing the sub-board.

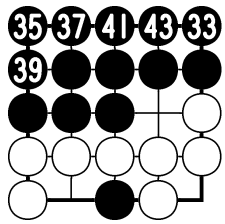
Lemma 1 *For n odd, let T (resp. B) denote the set of legal top (resp. bottom) positions. For every permutation of T, every permutation of B, every permutation of stones and every permutation of empty points in every $t \in T$, and every permutation of stones and every permutation of empty points in every $b \in B$, there is a unique game of Go.*

An example game serves to illustrate the proof.

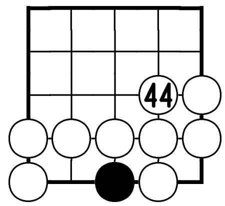


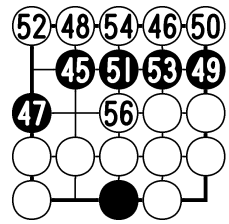
W 2,6,10 pass — B 13,15,17,19,21,25 pass — B 29,31 pass

With an empty control, the game sets up the first top position in some order, with consecutive stones of the same color requiring an intermediate pass by the other side. Move 11 changes the control, entering the next phase. The first top setup is unique not only in using an empty control, but also in skipping the bottom fillup afterwards, a fact that will be exploited later. Now the first bottom position is set up in some more arbitrary order. Move 27 changes the control again, to start a top fill. First White expands her strings until they have only 1 liberty.



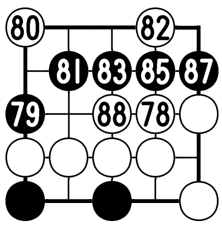
W 34,36,38,40,42 pass



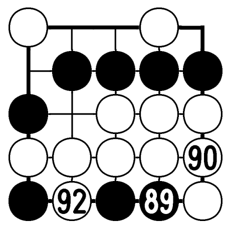


B 55 pass

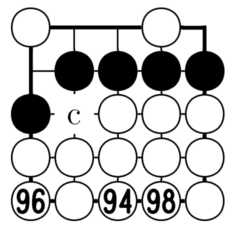
Then Black plays the originally empty points in some order, except for having to play e5 first to vacate the other points. But one of the original white stones, say at a5, can assume its place in the order. White 44 captures the entire top while changing the control yet again, preparing to set up the next top position. Let's fast-forward to the end of the game. This will be thousands of moves later, but for notational convenience we'll pretend it's move 78.



W 80,84 pass



B 91 pass



B 93,95,97 pass

After the last of the top positions has been filled up and captured at move, let's say 78 to keep things manageable, we set up the first top position again, but this time, with move 88, proceed to the fillup of the last bottom position, after which the game ends (a Black play at 'c' is prohibited by superko).

Proof. The existence of the permutation implied move sequence is clear from the sample game. What is left to show is that every move is legal, i.e. no position is repeated. By construction, every single setup phase is repetition free. In the fillup phase, say, on top, the first part of expanding white strings to a single liberty is repetition free, and so is the second part of forming a solid black block. Since capture of a white string in the second part removes white stones present at the fillup start, there is no repetition across the two parts either. Since every phase except the initial setup and final fillup, has half the board frozen in a sub-board position that gets set up only once, there is no repetition across phases either. •

Lemma 2 *Each combined setup and fillup of a sub-board position allows for at least* $\lfloor \frac{K}{2} \rfloor! \lceil \frac{K}{2} \rceil! \approx \pi K (\frac{K}{2e})^K$ *permutations, where* K *is the sub-board size in points.*

Proof. Let the position have $0 \leq k \leq K$ non-empty points. The setup allows for $k!$ permutations, while the fillup allows for at least $(K-k)!$ permutations. Minimizing the product of these gives the stated lower bound. •

Our Lemmas combine to prove

Theorem 1 *For n odd, $N(n) \geq (\lfloor \frac{K}{2} \rfloor ! \lceil \frac{K}{2} \rceil !)^{2H_n} H_n!^2 \approx 2\pi H_n (\pi K (\frac{K}{2e})^K \frac{H_n}{e})^{2H_n}$, where $K = \frac{n^2-3}{2}$ is the sub-board size.*

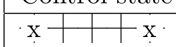
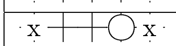
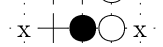
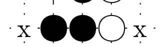
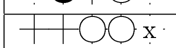
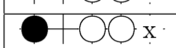
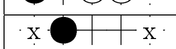
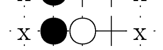
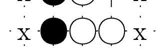
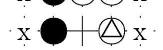
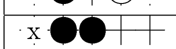
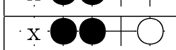
In order to apply this to $n = 19$ we need a good lower bound on H_{19}. We computed the number of legal 11×9 positions ending in an 11-stone black border as 250022411912498300328152248672940333961060, and the number of legal 8×10 positions ending in an 10-stone black border as 6838262511331611487262030859411923. Multiplying these together provides the lower bound $H_{19} > 1.7 \cdot 10^{75}$.

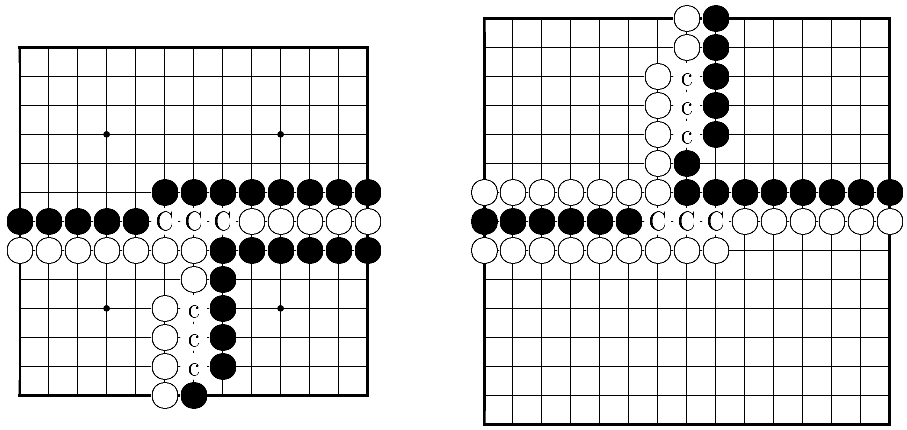
Corollary 1 *There are at least $(5!6!)^{646} 323!^2 > 10^{4314}$ Go games on 5×5, and at least $(89!90!)^{2H_{19}} H_{19}!^2 > 10^{10^{77}}$ Go games on 19×19.*

The theorem in fact applies to pseudo-legal sub-board positions as well but we refrain from formal proof, as we'll need the legal ones in the next section.

3 Nested scheme

With the basic scheme, we can play games visiting all legal positions of roughly half the board. To improve our lower bounds, we need to increase the fraction of the board iterated over beyond a half. While one half of the board is frozen, we have a lot of freedom in the other half. Instead of just setting up one sub-board position there, let's run a nested scheme beforehand. This requires additional main control states, to distinguish these parts. Yet we want to limit this control to 3 points. So instead, we consider the control state in context, where the context can be the color of a point horizontally adjacent to the control (marked 'x' for don't-care), or whether a stone in the control is capturable, denoted by a triangle.

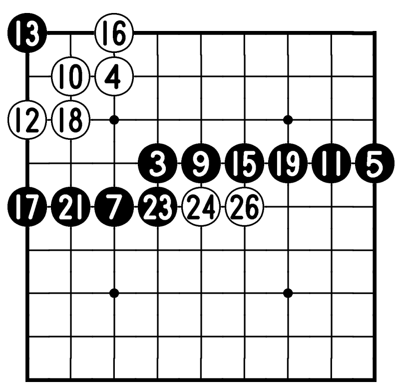
Control state	top sub-board	bottom sub-board
	setup	
	play sub	frozen
	play sub	frozen
	play sub	frozen
	last play in sub	frozen
	setup	frozen
	frozen	fillup
	frozen	play sub
	frozen	play sub
	frozen	play sub
	frozen	last play in sub
	frozen	setup
	fillup	frozen



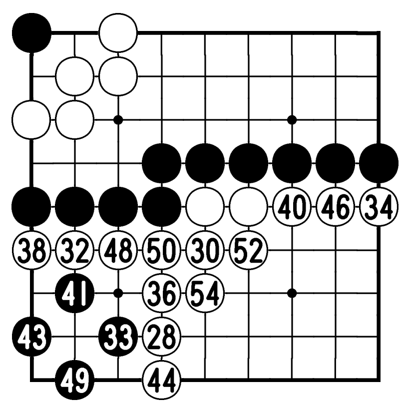
These diagrams show the nested controls on 13×13 and 15×15 boards. The main control is marked 'C', and the sub-control, situated about either halfway above or halfway below the center, is marked 'c', and acts just as the basic scheme control. For $n \equiv 1 \bmod 4$, the sub-control splits the sub-board evenly into two sub-sub-boards, but for $n \equiv 3 \bmod 4$, one side is necesarily larger by 1 point. To allow for alternating positions from the two sets of legal sub-sub-board positions, we truncate the bigger set to match the size of the smaller, which we denote Q_n (Q for quarter).

Lemma 3 *For a position $p \in T$ (resp. B), denote the possibly truncated set of legal top-left (resp. bottom-left) positions as p_L, and the possibly truncated set of legal top-right (resp. bottom-right) positions as p_R. For every Lemma 1 game, for every sub-board position p in $T \cup B$, for every permutation of p_R, every permutation of p_L, every permutation of stones and every permutation of empty points in every $r \in p_R$, and every permutation of stones and every permutation of empty points in every $l \in p_L$, there is a unique game of Go.*

Again we illustrate the proof with an example game.

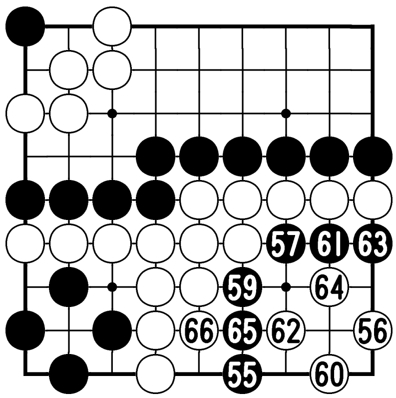


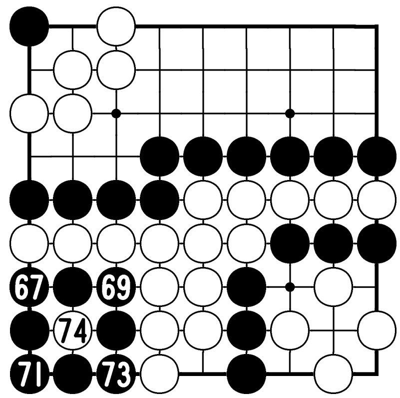
W 2,6,8,14,20,22 B 25 pass



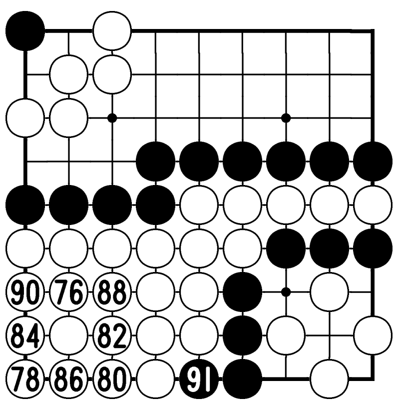
W 42 B 27,29,33,35,37,... pass

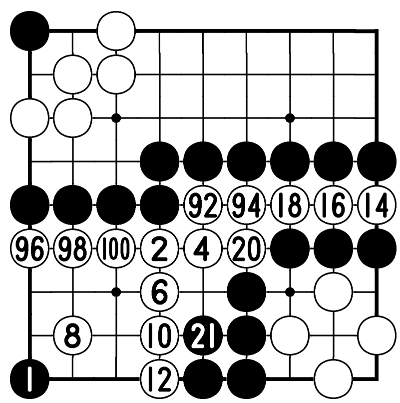
With an empty control, the game sets up the first top position in some order. Move 23 changes the control, entering bottom play. In bottom play, vacated main control point are always filled, as with moves 24 and 26, except in the final sub-game cleanup. With the sub-control empty, we then set up a bottom-left position, ending with the sub-control move at 54.



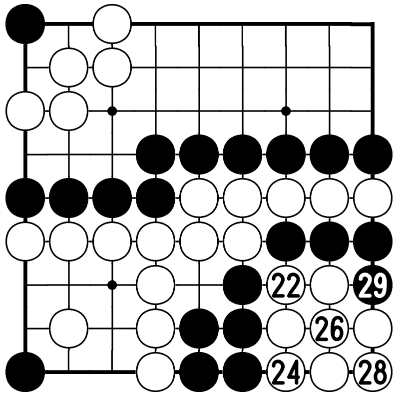


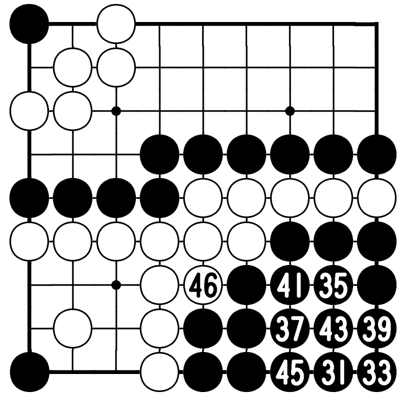
Next a bottom-right position is set up, and move 66 changes sub-control to the first quarter fill. Black expands her string to a single liberty to be captured by White 74.



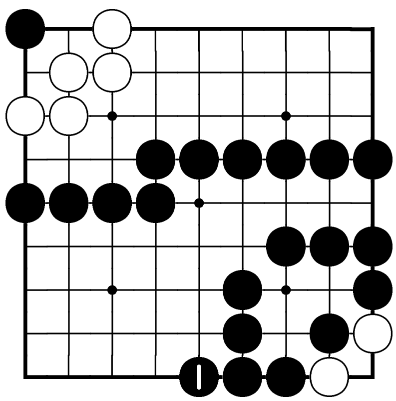


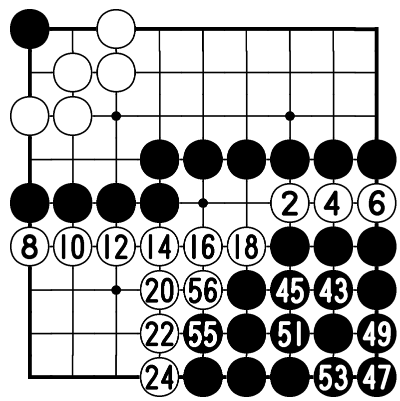
White then fills the whole bottom-left and is captured by Black 91, starting a new bottom-left setup. For clarity we show move numbers modulo 100. The setup ends with move 21 starting a bottom-right fillup phase.



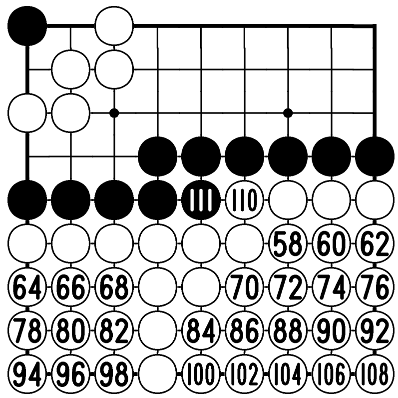


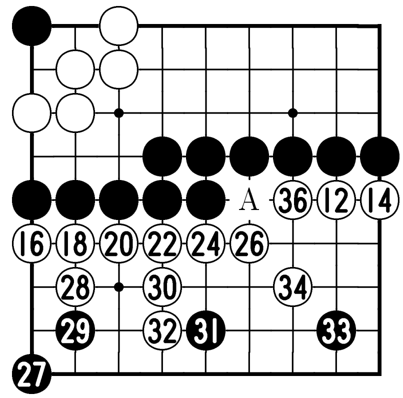
Now that the black border has one string not adjacent to the sub-control, we must take care to avoid capturing it during white string expansion (always possible due to White's multiple choice). As in the basic scheme, we can iterate through all quarter-board positions in this sub-game.





Fast forward to the capture of the last of the bottom-left positions in this sub-game with Black 1. We now add back the white border but skip adding white stones to the main Control. Next, we fill up the last of the bottom-right positions, effectively concluding the bottom sub-game. From move 55, we basically play a bottom fillup.





White 110 adds the single liberty stone to the main control, letting Black clear the whole bottom with 111. We now set up the first bottom position, freezing it with White 74. The following top fillup will be concluded by a White capture at 'A', initiating a top sub-game.

Proof. (sketch) As before, it remains to show that no position is repeated. Our previous proof of the basic scheme applies to each sub-game, up until the last sub-sub-board fillup. Then the fillup phase of sub-board is safe from repetition as the main control is left with two liberties. The last move of this fillup, its capture and the setup of next sub-board position, the fillup of the previous main position and its capture, are all protected by distinct main control codes. Thus there can be no repetition during a sub-board position freeze. Furthermore, as each of these sub-board positions gets used only once, there is no repetition across main phases. •

Lemma 4 *Each combined setup and fillup of a sub-sub-board position allows for at least* $10^{\frac{33}{4}}$ *permutations for* $n \geq 9$*, and at least* $10^{\frac{402}{4}}$ *permutations for* $n \geq 19$.

Proof. This follows from Lemma 2. The $n = 9$ sub-sub-board positions consist of at least 15 points, giving a choice of $7!8! = 203212800 > 10^{8.25}$ permutations. The $n = 19$ sub-sub-board positions consist of at least 83 points, giving a choice of $41!42! > 10^{100.5}$ permutations. •

Since each of the constructed games has $2H_n$ sub-games each consisting of $2Q_n$ combined sub-sub-board setup/fills, we immediately obtain

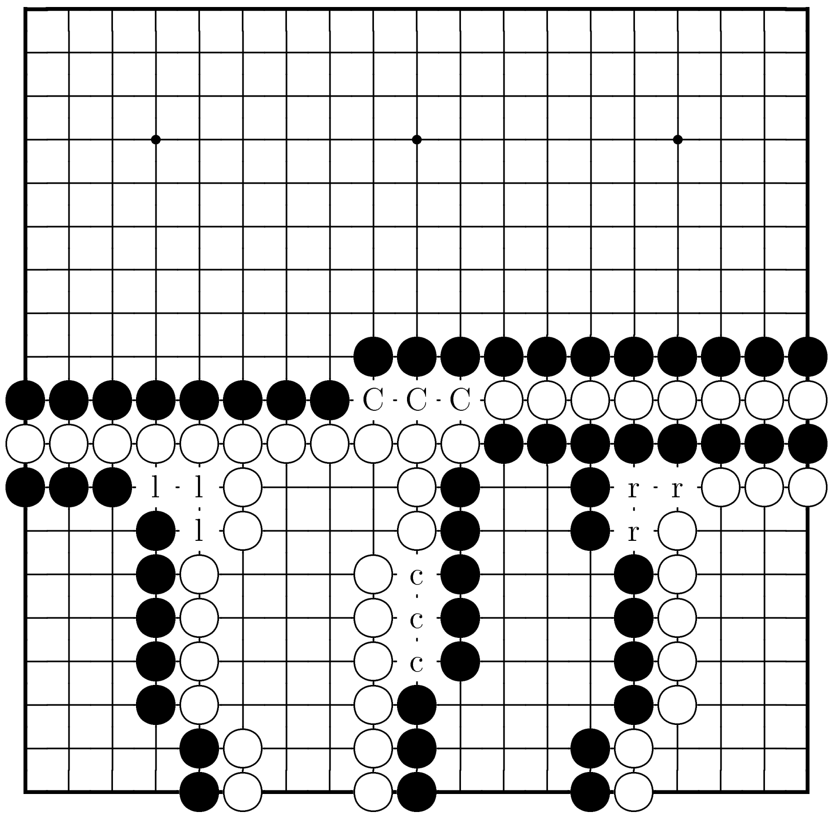
Theorem 2 *For* $n \geq 9$ *odd,* $N(n) \geq 10^{33Q_nH_n}$*, while for* $n \geq 19$ *odd,* $N(n) \geq 10^{402Q_nH_n}$.

We computed $H_9 = 95276398927407$ and $Q_9 > 10000$. We showed earlier that $H_{19} > 1.7 \cdot 10^{75}$, and computed $Q_{19} > 8.4 \cdot 10^{30}$.

Corollary 2 *There are at least* $10^{10^{19}}$ *Go games on* 9×9*, and at least* $10^{10^{108}}$ *Go games on* 19×19*, well over a googolplex.*

4 And beyond ...

We need not stop nesting at 2 levels. The diagram below shows a triple nesting, with even less uniformity in shape, and diminishing returns, of what can be estimated as $10^{10^{117}}$ games. Considering the burden of proof, and how big of a gap remains with the known upper bound of $10^{10^{171}}$, we leave that as an exercise for the reader.



5 Conclusion

The original lower bound of $10^{10^{48}}$ on the number of 19x19 games, proved in [2], uses approximately half the board to cycle through binary configurations. This paper obtains a much stronger result by improving on both aspects. The nesting subdivision construction allows a majority of the area to be used for cycling through configurations, and these can be ternary rather than binary. Combined, these improvements push the number of games beyond $10^{10^{100}}$, popularly known as a 'googolplex'.

6 Acknowledgments

Many thanks to Arnaud Knippel for comments on early versions.

References

[1] Sensei Library http://senseis.xmp.net/?NumberOfPossibleGoGames

[2] John Tromp and Gunnar Farnebäck, Combinatorics of Go, 2007, https://tromp.github.io/go/gostate.pdf

[3] http://tromp.github.io/go.html

[4] http://tromp.github.io/go/legal.html

[5] https://en.wikipedia.org/wiki/Game_complexity#Complexities_of_some_well-known_games

[6] Supporting materials http://matthieuw.github.io/go-games-number/