A Generalized Framework for Analyzing Capturing Races in Go

Martin Müller
Department of Computing Science, University of Alberta
Edmonton, Canada T6G 2E8
mmueller@cs.ualberta.ca

Abstract

Capturing races or *semeai* are an important element of Go strategy and tactics. We extend previous work on semeai [1] by introducing a more general framework for analyzing semeai, based on the new concepts of *conditional combinatorial games* and *liberty count games*. We show how this framework encompasses earlier concepts such as plain liberty regions and plain eye regions. Furthermore, we discuss how to use upper and lower bounds on such games in a semeai solver.

1 Capturing Races in Go

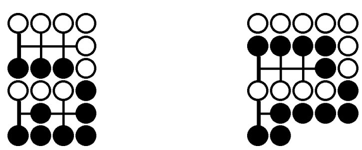

Figure 1: Two simple semeai

A *semeai* in the game of Go can be defined informally as "a race to capture between two adjacent groups that cannot both live". Figure 1 shows two simple cases. In earlier work [1, 2], we gave more formal definitions of semeai, and described nine different classes of semeai. Semeai of classes 0, 1 and 2 can be detected and evaluated statically, without search. The other classes cover semeai that can be resolved by search, potential semeai, and unclear situations which might end up as a race to capture. This paper contains the following contributions:

1. Section 2 develops a general framework for analyzing semeai in terms of *conditional combinatorial games* and *liberty count games*. This framework provides a new basis for the previous model introduced in [1] that used eye and liberty regions.
2. Section 3 extends the semeai analysis framework for cases where an exact game may be difficult to compute, but an easier to obtain upper or lower bound can lead to a resolution of the semeai problem.

We only seek to determine the win/loss/seki outcome of a semeai. We do not consider other issues here, such as maximizing the score, computing the combinatorial game value, or determining whether winning a semeai is beneficial at all [1]. In the remainder of this paper we will use the following terms that were defined in [1]: *Essential and nonessential block, outside liberty, plain outside liberty, shared liberty, eye, plain eye, nakade, class n semeai*.

2 Conditional Combinatorial Games

Conditional combinatorial games (ccg) are an abstraction of play in a local region that is part of a semeai. In some states nonlocal information is required to determine whether a move is possible. For example, the last liberty in an eye can be taken only as the last overall liberty of a block surrounding the eye.

2.1 Conditional Combinatorial Games and their Context

A conditional combinatorial game (ccg) is defined recursively just like a combinatorial game in terms of sets of left (Black) and right (White) move options, which are again ccg's. The game $\{|\}$ where no player can move is identified with the symbol 0. However, ccg's are not pure combinatorial games because the legality of some moves depends on a nonlocal context. We indicate a conditional move by using the context condition as a subscript to the option as in the following example: In the game $A = \{\ldots | \ldots, B_C\}$, White's move from A to B is possible if and only if condition C is currently true. The most important context for semeai, which we will call $L0$, is the context where a specific block has a total of 0 liberties elsewhere. For example, a White one point eye of a single block can be described by the following game: $E1 = \{0_{L0}|\}$. This can be read as follows: White does not have a sensible move, so the set of right options is empty. Black can move to 0 if and only if condition $L0$ is true, that is if White does not have any liberties elsewhere. Another frequent kind of nonlocal context is ko. A simple ko such as $A = \{1|B\}, B = \{A|0\}$ can be described by a ccg by indicating the context K in which the move from A to B is legal, and the context K' in which the move from B to A is legal, as follows: $A = \{1|B_K\}, B = \{A_{K'}|0\}$.

A very important point to note is that while we have chosen a notation that looks similar to classical combinatorial game theory, none of the usual rules for simplifying combinatorial games apply to ccg because the context conditions fundamentally alter the way that a sum of such games is played. Usually, a ccg will be played within the context of a sum of other ccg, which together make up the whole semeai position. If play switches back and forth between different ccg, then the truth status of conditions will typically change during play.

2.2 Monotone Conditions

We call a condition c monotone with respect to a ccg G if it has the following property: if c becomes true at some stage during play of G, then it remains true for the remainder of the game. For example, in semeai $L0$ is often a monotone condition: once all out-of-region liberties have been taken, there will never be any new liberties created there. Of course, it is easy to give counterexamples where $L0$ is true temporarily but later violated because new outside liberties are created by a capture.

2.3 Pruning Moves with Dominated Context in a Ccg

We will call a context C_1 more specific than context C_2 if C_1 logically implies C_2. The empty context, which is always true, is the least specific context.

If one move requires a more specific context than another, but leads to the same result, it can be safely pruned. If B_C and B_D are both options in a game and context D is more specific than C, then move B_D can be safely pruned. An example would be the choice of taking a liberty that depends on a ko capture, or taking another liberty first, which makes the ko irrelevant.

2.4 Liberty Count Games

Ccg do not contain enough information to determine the number of liberties of blocks. However, that information is needed in order to use one region as a context of another in a semeai. Therefore we introduce *liberty count games*, which keep such information.

A liberty count game is defined over a set of blocks in a region and consists of two parts: a ccg that describes the possible moves of each player, and a *liberty counting function* $L(b, g)$ that returns the number of liberties of block b in a ccg g. A liberty counting function does not have to be defined for all blocks of a region. In semeai, typically this function will be defined only for the subset of *essential blocks* [1]. The remaining blocks are considered nonessential and not considered directly in the model, for example the stones inside a nakade shape that can be freely sacrificed. However, even those blocks are indirectly included in the model, because they affect the liberty count of essential blocks and the legal moves.

In general, the set of blocks involved in a region can change during play, by creating new blocks and by merging or capturing old ones. We assume that no new essential blocks are created during play, that merged blocks assume the identity of all constituent blocks, and that the liberty count function returns 0 for a captured block. In general, capturing an essential block finishes a semeai and all liberty count games associated with that block.

There are important special cases of liberty count games. In one case, only one player has liberties in a region. In another special case, each player has only a single essential block in the region. In the simplest case, only one player has a single essential block in the region.

2.5 Pruning Dominated Moves in a Lcg

It is possible to define a partial order of lcg by recursively testing whether the liberty counts in one game always dominate the other. Domination means that own blocks have at least the same number of liberties, while opponent blocks have the same or less. Given such a partial order, moves that lead to a worse lcg are dominated and can be pruned. Examples would be filling own liberties or eyes, or failing to extend liberties where that is possible.

2.6 Some Examples of LCG

2.6.1 Plain Outside and Shared Liberties

The game $G_n = (P_n, L)$ consisting of a single Black block b with n plain outside liberties can be defined by $P_0 = 0$, $P_{n+1} = \{|P_n\}$ and $L(b, P_n) = n$. Similarly, a plain shared liberty region between Black block b and White block w is defined by the lcg $G_n = (S_n, L)$ with $S_0 = 0$, $S_{n+1} = \{S_n|S_n\}$ and $L(b, S_n) = L(w, S_n) = n$.

2.6.2 Two-eyed Group

A two-eyed single-block group g can be described by the lcg $E2 = (0, L)$ with $L(g, 0) = 2$.

2.6.3 Large Eyes

The reason that large eyes are so valuable in semeai is their ability to provide extra liberties late in a fight, and force the opponent to fill shared liberties first. A characteristic of the different eyes is how many moves are left after the liberty count goes down to 1 for the first time. This is the crucial point since it contains the conditional move 0_{L0}. Figure 2 shows such a sequence, starting from a 7 point eye.

Figures 3 and 4 show the liberty counts during two long semeai sequences, each involving a 7 point eye. In Figure 3, initially Black has a 6 point eye containing two white stones and 5 outside liberties, while White has a 7 point eye containing six black stones and three outside liberties. There are two shared liberties. The figure shows Black's failed attempt to capture White. Up to move 6, both remove outside liberties. With moves 7 and 9, Black fills the shared liberties since there are no other liberties that can be played. White is in atari and must capture with 10, reducing the area to a six point eye. After move 13, both have no outside liberties and a six point eye containing two opponent stones. In this balanced situation the first player can win by one move. In this case it is White. Both players' liberty count sequences are in lockstep from now on, and White remains one move ahead until capture.

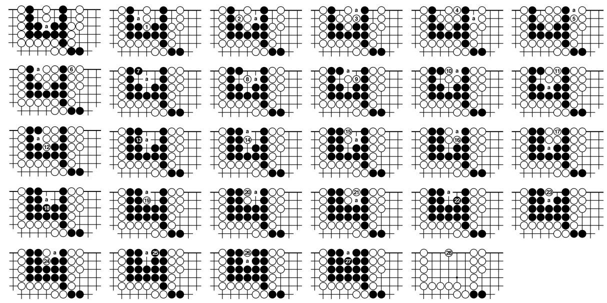

Figure 2: 7 point nakade filling sequence

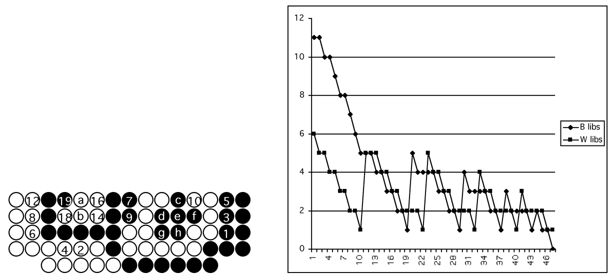

Figure 3: 7 point eye vs 6 point eye. 11 at e, 13 at d, 15 at h, 17 at g, moves from 20 at b, c, f, e, 14, h, a, d, 16, 18, a, g, c, d, 16, e, b, 14, a, g, h, d, 16, b, a, e, 16

Figure 4 pits a seven point eye against an eyeless group with many liberties. Up to 7, Black fills outside liberties and White fills Black's eye space. From 8 to 14, White fills the four shared liberties.

2.6.4 Protected Liberties

Protected liberties have properties halfway between outside liberties and eyes. Protected liberties can be occupied directly only if $L0$ holds, but require one or more

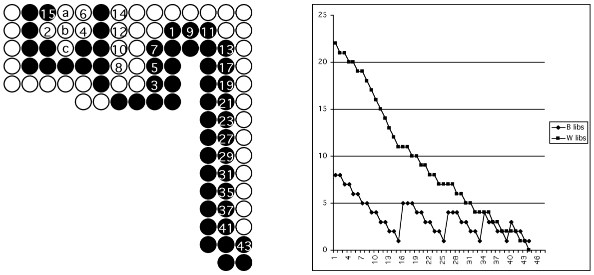

Figure 4: 7 point eye vs no eye. 16 at b, 18 at 4, 20 at a, 22 at c, 24 at 2, 25 at 6, 26 at b, 28 at c, 30 at a, 32 at 2, 33 at 4, 34 at b, 36 at c, 38 at a, 39 at 2, 40 at b, 42 at c

approach moves otherwise.

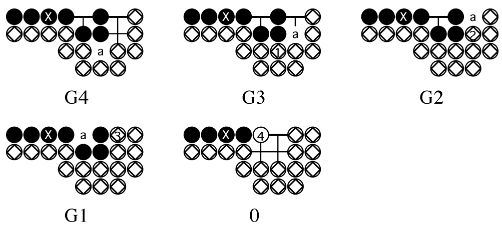

Figure 5: Protected liberty

Figure 5 shows a protected liberty of the block X. The ccg are $G4 = \{|0_{L0}, G3\}$, $G3 = \{|0_{L0}, G2\}$, $G2 = \{|0_{L0}, G1\}$, and $G1 = \{|0\}$. The liberty count function is $L(X, G) = 1$ for $G \in \{G1, \ldots, G4\}$ and $L(X, 0) = 0$.

3 Bounds

Instead of computing an exact lcg, it may be easier to determine the winner of a semeai by using bounds. As a trivial example, having any combination of n outside liberties is at least as good as having n plain liberties, but it may be better because the opponent may need extra approach moves and/or the player may have eye-making potential in the region.

Example: Figure 6 shows problem D from Figure 14 of [2], which was solved there by the search method of *partial order bounding*. We will show how to solve problem D statically by using bounds. Black has 3 outside liberties and an eye status of 5 with one opponent stone inside.

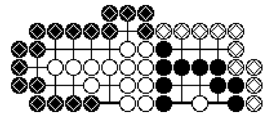

Figure 6: Semeai test problem D, from [2]

Because there are no shared liberties, this is equivalent to a plain liberty filling sequence of 10 moves. White has 8 plain liberties on the right and bottom, but White's eye space is unsettled on the left side. However, White has 2 non-plain liberties there, so White can win going first. In this case, creating an eye would be a fatal mistake for White. However, in other circumstances where Black does not have a large eye and where there are shared liberties, creating an eye would be the only good move.

In terms of partial order evaluation, we extend the evaluation of lcg by defining new games representing upper and lower bounds on real games as proposed in [2].

4 Summary and Future Work

We introduced the concepts of *conditional combinatorial games* and *liberty count games* as tools for the local analysis of semeai. We have shown how to integrate nonlocal aspects such as the total liberty count and ko status into such a framework as context conditions, and have given some examples to demonstrate that this work is a generalization of our previous work on semeai [1, 2].

Future work includes working out the details of an implementation and of the local search process, and researching the overall strategy for selecting appropriate local analyses in complex semeai situations.

References

[1] M. Müller. Race to capture: Analyzing semeai in Go. In *Game Programming Workshop in Japan '99*, volume 99(14) of *IPSJ Symposium Series*, pages 61–68, 1999.

[2] M. Müller. Partial order bounding: A new approach to evaluation in game tree search. *Artificial Intelligence*, 129(1-2):279–311, 2001.