# Life in the Game of Go

DAVID B. BENSON*
*Computer Science Department, Washington State University, Pullman, WA 99163*

ABSTRACT

The Oriental game of Go contains a unique method by which pieces, called stones, are captured and made safe from capture. A group of stones safe from capture is called safe, unconditionally alive, or similar terms. Life or its lack can be determined by lookahead through the game tree, at some expense. We present a graph-theoretic static analysis of the board arrangement which determines unconditional life or its lack, together with proofs of its equivalency to look ahead. An algorithm for the static evaluation is given and we argue that it is the preferable method for computer Go play. These results constitute the first realistic theorems in the theory of Go.

## 1. INTRODUCTION

The Oriental game of Go has been analyzed by Thorp and Walden [13, 14] to determine precise rules, thereby making possible computer studies of the game. There is some interest in using Go to study problems of artificial intelligence, or just to play Go on the computer for its own sake. Dowsey [1] reviews the current situation.

One of the difficulties for both human and computer players of Go is determining when a group of pieces, called stones, is impossible to capture by the opponent, although the player makes no move to save them. This situation is called *unconditional life,* and its determination is crucial for intelligent play. The rules of the game [14, 4], while completely determinate in this regard, give such latitude of play that the determination of unconditional life may tax the computational abilities of human or machine.

One method to determine unconditional life is to carry out the game tree lookahead with restrictions on the possible moves. In spite of these severe restrictions, even the simplest unconditional life situations may require looking at 30 subsequent arrangements. (The terminology follows Thorp and Walden

*Research supported in part by the National Science Foundation Grant GJ-1171.

[14]). This is a difficult task for human players, and while feasible on a computer, it is such a small part of the overall strategy and tactics of the game, although very important, that a better algorithm is called for.

The second method is to determine derived graph properties of the arrangement. If certain conditions hold, then the group has unconditional life. It is obvious from my own experience and from the literature on Go (Haruyama and and Nagahara [2], Segoe [11], and the other citations in the bibliography) that this pattern-recognition approach, as well as lookahead, is used by human players.

This note formalizes this static determination of unconditional life and proves that the conditions stated are equivalent to safety. The computational tractability of the static determination for unconditional life, together with other properties of the game obtained along the way, appear to make it preferable to lookahead in computer play.

## 2. NOTATION AND RULES

Our notation generally follows that of Thorp and Walden [14]. We have avoided using common Go terms in certain cases, since the imprecise common terminology does not exactly fit our analysis. The main difficulty is with the terms *eye* and *group* or *army*. These are used so disparately that we felt it best to avoid them. Unfortunately, no substitute for eye is obviously appropriate, so we give it a technical meaning related to the common terminology.

The aspects of Go important here will be formalized via graphs. A board is a $M \times N$ array of intersections. Each intersection is a vertex of the board graph. There is an edge between two intersections just in case either the $x$ coordinates are equal and the $y$ coordinates differ by one, or the $y$ coordinates are equal and the $x$ coordinates differ by one. For $M = 2$ and $N = 3$ the board graph is given in Fig. 1.

Fig. 1

In the usual way, two intersections are adjacent if there is a single edge between them. An *arrangement* is a function from the set of intersections $I$ to the set of possible states of each intersection, $\mathscr{S} = \{black, white, empty\}$. Denote arrangements by $a : I \to \mathscr{S}$. For each $x \in \mathscr{S}$, $a^{-1}(x)$ is the set of intersections in state $x$. Two intersections, not necessarily distinct, are state-connected if they are in the same state and there is a path between them, possibly of length zero, such that every intersection in the path is in the same state. State-connectedness is an equivalence relation partitioning $I$ into *blocks*. For each $x \in \mathscr{S}$ let $B(x)$ be the set of blocks for state $x$, i.e., every intersection in every block of $B(x)$ is in state $x$. A block of stones, that is a block in state *black* or a block in state *white*, is frequently called a *group* or *army*. However, these latter two terms also have other meanings in the Go literature and will not be used here. Figure 2 shows an arrangement of two black, one white and four empty blocks.

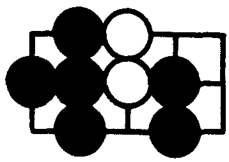

Fig. 2

A *region* is any set of intersections forming a connected subgraph of the board. For every pair of intersections in a region, there is a path between them lying entirely within the region. Clearly every block is a region. An intersection is in the *interior* of a region $R$ if every adjacent intersection is in the region, symbolized $Int(R)$. Otherwise, an intersection of $R$ is on the *border*. The immediate exterior of a region $R$, $Ext(R)$, is the set of all intersections not in $R$ which are adjacent to any intersection in $R$.

Let $E$ be the set of empty intersections, $E = a^{-1}$ (*empty*). The *liberties* of a block $b$ are the intersections in the set $L(b) = Ext(b) \cap E$. That is, a liberty is an empty intersection adjacent to the block. If the block $b$ is empty, $L(b) = \phi$.

A game is a sequence of arrangements $(a_0, a_1, \ldots, a_n)$ subject to the rules.

RULE 0 (a). For all $i \in I$, $a_0(i) = empty$, $a_0(I) = \{empty\}$.

(b). For all $k$, $a_{2k}$ and $a_{2k+1}$ differ by the addition of at most one black stone and the deletion of white stones subject to Rule 1, while $a_{2k+1}$ and $a_{2k+2}$ differ by the addition of at most one white stone and the deletion of black stones subject to Rule 1.

This rule states that the game begins with an empty board and that black and white alternate by placing stones on empty intersections or else passing. In $n$-handicap Go this rule is replaced by another in which the initial arrangement contains $n$ black stones at intersections specified by the rules while white plays the first move. See Tilley [15].

RULE 1 (CAPTURE). If a block $b$ of $x$-stones has exactly one liberty, $\# L(b) = 1$, and $\bar{x}$ (the other player) may play in the liberty, then upon doing so the block

$b$ is captured and removed from the board. Formally, if $a_k$ is the arrangement before capture, then

$$a_{k+1}(I-(b \cup L(b))) = a_k(I-(b \cup L(b))),$$

$$a_{k+1}(b) = \{empty\},$$

$$a_{k+1}(L(b)) = \{\bar{x}\}.$$

RULE 2 (NO SUICIDE). If a block $b$ of $x$-stones has exactly one liberty, $\#L(b)=1$, and there is no $\bar{x}$ block $b'$ such that $L(b')=L(b)$, then $x$ may not play at $L(b)$.

There are other rules, for which see Thorp and Walden (1972), but these three suffice for the present analysis. In particular, there is no need to consider the rules covering repetitive arrangements such as Ko, since the only concern is for unconditional safety.

## 3. SAFE BLOCKS

DEFINITION. Let $b$ be an $x$-block in arrangement $a_k$. Consider all subsequent arrangements $\mathscr{A}$ such that $x$ passes on every turn. These arrangements form a tree rooted in $a_k$. Each path from the root is denoted by $a_k, a_{k+1}, a_{k+2}, \ldots, a_n$. If block $b$ remains on the board in all arrangements in $\mathscr{A}$, then $b$ is *safe.*

This definition is intended to capture the intuitive notion of unconditional life. $x$ does not play at all, so cannot hinder attempts by $\bar{x}$ to capture $b$. By this definition we have no need to consider stupid plays by $x$ which put a safe block in danger of capture. In Fig. 3 the black block is safe, since white may not play in either of black's liberties due to the "no suicide" rule.

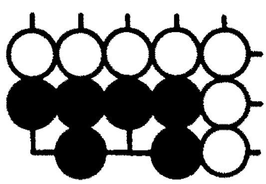

Fig. 3

The obvious algorithm for determining whether a block $b$ is safe is to attempt to play $\bar{x}$ stones on the intersections of $L(b)$. If there is no legal sequence of $\bar{x}$ plays which capture $b$, $b$ is safe. In Fig. 3 the block has only two liberties. In Fig. 4 the full tree of sequences of legal moves in $L(black)$ has 4! leaves. While the full search tree may be pruned by some elementary considerations of "inside" and "outside" liberties, in more complex situations the lookahead requirement is still large. In Fig. 5 the three black blocks are safe. The determination by lookahead requires filling three outside liberties, one inside liberty (in the corner) followed by six attempts to capture, two for each block.

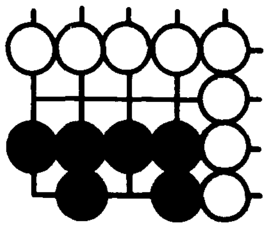

Fig. 4

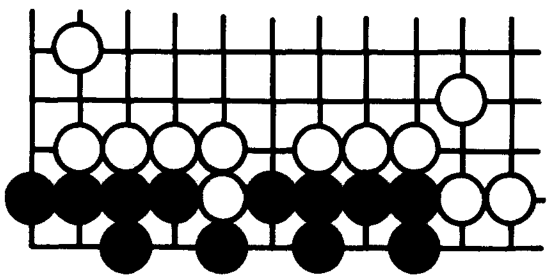

Fig. 5

There is a concept related to *safe* used in actual play, whigh might be called *almost always safe*. An $x$-block is almost always safe if in any sequence of subsequent arrangements $\bar{x}$ must play against the block $n$ times, for $n > 1$, before $x$ needs to respond in order to make his block safe. In this situation $\bar{x}$ would lose points trying to capture immediately, so the situation would be left as it is until nearer the end of the game. At some point $x$ may need to play to make his block safe, depending on the exact arrangement. Almost always safe blocks are not studied here. Our interest is in the static evaluation of safety, a considerably easier problem.

## 4. UNCONDITIONAL LIFE

The static determination of life requires additional concepts.

DEFINITION. A *small $x$-enclosed region $R$* is a region such that

(1) No $x$-stone is in $R$, $x \notin a(R)$.

(2) $R$ is surrounded by $x$-stones, $a(Ext(R)) = \{x\}$ .

(3) Each intersection of the region is on the border or contains an $\bar{x}$-stone or both, hence $a(Int(R)) \subseteq \{\bar{x}\}$ .

DEFINITION. A region $R$ is *healthy* for block $b$, symbolized $H(R, b)$, if $R$ is a small $x$-enclosed region such that every empty intersection in $R$ is a liberty of $b$, $R \cap E \subseteq L(b)$.

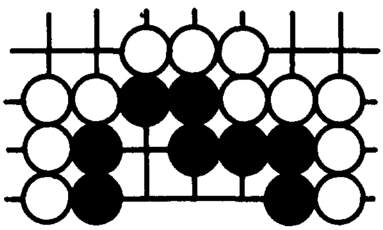

Fig. 6

The small black-enclosed region in Fig. 6 is not healthy for either black block. Provided black passes, either black block may be captured without capturing the other.

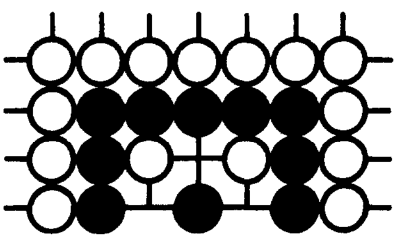

Fig. 7

The empty region in Fig. 7 is healthy for both black blocks. However, neither black block is safe since white may fill all the empty intersections, capturing both black blocks on the third move. For safety, black requires two distinct healthy regions with additional properties.

DEFINITION. Let $NB(R)$ be the set of blocks neighboring region $R$, $NB(R) = \{b \mid Ext(b) \cap R \neq \phi\}$

DEFINITION. Let $X$ be a set of $x$-blocks with $b \in X$. If $R$ is healthy for $b$, $H(R,b)$, and all blocks neighboring $R$ are in $X$, $NB(R) \subseteq X$, then $R$ is *vital* to $b$ in $X$, $V(R,b,X)$.

DEFINITION. Let $X$ be a set of $X$-blocks, $X \subseteq B(x)$, such that all $b \in X$ have two distinct vital regions, i.e., $\exists\ R_1, R_2,\ R_1 \neq R_2$ such that $V(R_i, b, x),\ i = 1, 2$. Then $X$ is *unconditionally alive.*

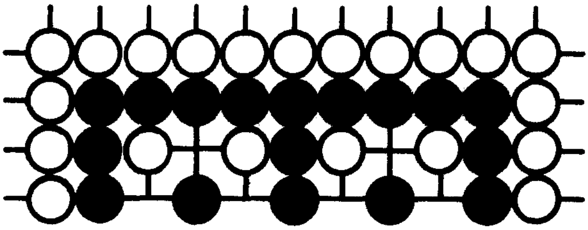

Fig. 8

The two small black-enclosed regions in Fig. 8 are healthy for all three black blocks. Let $b_0$ be the large black block, $b_1$ and $b_2$ be the other two black blocks. With $X = \{b_0, b_1, b_2\}$ then both regions are vital but since $b_1$ and $b_2$ fail to have two vital regions $X$ is not unconditionally alive. In fact, white may capture by filling all the liberties in any order.

The definition takes care of complexities exemplified by Figs. 5 and 8. That all safe situations are characterized is the substance of the theorems to follow.

## 5. THE EQUIVALENCE OF SAFETY AND UNCONDITIONAL LIFE

The proofs require the following notation.

DEFINITION. Let $R$ be healthy for $b$. If $\#NB(R) = 1$, then $R$ is an *eye*. Otherwise, $R$ is a *joiner.*

LEMMA 1. *Let $b \in B(x)$. If $b$ has two or more distinct eyes, then $b$ is safe.*

*Proof.* Any sequence of moves by $\bar{x}$ capturing $b$ must completely fill one eye with $\bar{x}$-stones before the others. Let $R$ be the first eye filled. We claim that $R$ is now an $\bar{x}$-block. In the original arrangement, every interior intersection of $R$ contained an $\bar{x}$-stone. Hence, $\bar{x}$ could only fill border intersections. Since $R$ is the union of its interior and its border, when filled by $\bar{x}$ it is an $\bar{x}$ block with no liberties. Yet $b$ has not been captured since there is at least one liberty of $b$ in another eye. So, by the "no suicide" rule, this arrangement is illegal. By symmetry $b$ is safe. ■

LEMMA 2. *Let $b \in B(x)$. If $b$ has an eye $R_0$ and one or more joiners, $R_i$, $1 \leq i \leq n$, $n \geq 1$, then at least one block other than $b$ neighboring each joiner $R_i$ must be captured before $b$ can be captured.*

*Proof.* By the previous analysis, the eye $R_0$ cannot be completely filled until the final capturing move. So $x$ must fill all the liberties of $b$ in all the joiners first. Since every empty intersection in each joiner $R_i$ is a liberty of $b$, the "no suicide" rule requires that upon filling the last empty intersection of $R_i$, $\bar{x}$ must capture. Since $b$ still has at least one liberty in its eye, some other $x$-block must be captured. ■

LEMMA 3. *Let $b \in B(x)$ such that $b$ has no eyes but has joiners $R_i$, $1 \leq i \leq n$, $n \geq 2$. Then at least one block other than $b$ neighboring each of $n - 1$ of the joiners must be captured before $b$ can be captured.*

*Proof.* An obvious modification of the proof of Lemma 2. ■

THEOREM 1. *If $X \subseteq B(x)$ is unconditionally alive, then every block in $X$ is safe.*

*Proof.* To arrive at a contradiction, assume $X$ is unconditionally alive and that $b_0 \in X$ is not safe. From Lemma 1 and the lack of safety for $b_0$, $b_0$ has at most one eye. Since $X$ is unconditionally alive, there is at least one joiner $R_1$ vital to $b_0$ in $X$, $V(R_1, b_0, X)$. If $b_0$ has no eyes, it has at least two vital joiners. Since $b_0$ is not safe, by Lemmas 2 and 3, it must be possible to capture a block other than $b_0$ neighboring some vital joiner, say $R_1$. Let $b_1$ be the first block neighboring $R_1$ to be captured. For the induction, assume that $b_0, \ldots, b_k$ are all distinct members of $X$ such that none of $b_0, \ldots, b_k$ are safe, and $b_{i+1}$ must be captured before $b_i$, $0 < i < k$. Further assume that there is a sequence $R_1, \ldots, R_k$ of regions such that $V(R_{i+1}, b_i, X)$ and $b_{i+1} \in NB(R_{i+1})$, $0 \leq i \leq k$, and that $b_i$ is the first block neighboring $R_i$ to be captured, $0 < i \leq k$. Consider $b_k$. Since $b_k$ is not safe, by Lemma 2 and the fact that $b_k$ is the first block neighboring $R_k$ to be captured, $b_k$ has no eyes. As $b_k \in X$, there are at least two regions vital to $b_k$ in $X$, one of which may be $R_k$. Let $R_{k+1}$ be another. To capture $b_k$ it is necessary to fill $R_{k+1}$ before $R_k$, since by hypothesis $b_k$ is the first block neighboring $R_k$ to be captured. Let $b_{k+1}$ be the first block neighboring $R_{k+1}$ to be captured. Clearly $b_{k+1}$ is distinct from $b_0, \ldots, b_k$. Thus, $b_0, \ldots, b_{k+1}$ and $R_1, \ldots, R_{k+1}$ satisfy the induction hypothesis and we conclude that $X$ is of infinite cardinality. Since $X$ is finite, the desired contradiction is obtained. ■

We now show that unconditional life is a property of all safe blocks. The proof involves filling the liberties of blocks, so it is advantageous to consider outside and inside liberties separately.

DEFINITION. An *inside* liberty of $b \in B(x)$ is any liberty of $b$ in a small $x$-enclosed region. Denote the set of inside liberties by $IL(b)$. The *outside* liberties of $b$ are $OL(b) = L(b) - IL(b)$.

LEMMA 4. *Let $b \in B(x)$ in arrangement $a_k$ with $IL(b) = \phi$. All outside* *liberties of b may be filled by $\bar{x}$, i.e., there is a subsequent legal arrangement $A_n$, $n \geq k$, such that $a_n(OL(b)) = \{\bar{x}.\}$*

*Proof.* Let $i$ be any outside liberty of $b$, $i \in OL(b)$. Consider the block of empty intersections containing $i$, $[i]$, together with all neighboring $\bar{x}$ blocks, to form a region,

$$R = [i] \cup \bigcup (B(\bar{x}) \cap NB\,[i])$$

If $a_k(Ext(R)) = \{x\}$ then $Int\ [i] \neq \phi$ since $R$ is not a small $x$-enclosed region. If $a_k(Ext(R)) \neq \{x\}$ then $Ext(R)$ contains an empty intersection by construction of $R$. In either case, if $\bar{x}$ fills $[i] \cap L\ (b)$, while $x$ passes, producing the arrangement $a_m$, the resulting $\bar{x}$ blocks still have at least one liberty. Since only $\bar{x}$ stones have been added, the small $x$-enclosed regions remain the same and in the arrangement $a_m$, $OL_m(b) = OL_k(b) - [i]$. The conclusion follows from a trivial induction. ■

THEOREM 2. *Let $X \subseteq B(x)$ be the set of all safe x-blocks. Then X is unconditionally alive.*

*Proof.* Suppose $X$ is not unconditionally alive. Then there exists some $b \in X$ which does not have two vital regions. This might occur in one of three ways.

(1) There are not two or more small $x$-enclosed regions neighboring $b$.
(2) There are two or more small $x$-enclosed regions neighboring $b$ but at most one is healthy.
(3) There are two or more healthy regions for $b$, but at most one is vital.

We will show that there is a sequence of legal moves by $\bar{x}$ which capture $b$. By Lemma 4 we may assume that all outside liberties of all blocks in $X$ are filled. But if case (1) holds, either $b$ has no inside liberties so is captured, or else has one small $x$-enclosed region which when filled captures $b$. Consider case 2 and let $R_i$, $1 \leq i \leq n$, be the unhealthy small $x$-enclosed regions neighboring $b$. For each $R_i$, there is an empty intersection in $R_i$ not adjacent to $b$, i.e., $(E \cap R_i) - IL(b) \neq \phi$. So $\bar{x}$ may fill $R_i \cap IL(b)$ to form a block with at least one liberty. After all such $R_i \cap IL(b)$, $1 \leq i \leq n$, have been filled, the remaining liberties of $b$, if any, are in the healthy region. This may now be filled by $\bar{x}$ since the last stone played in the healthy region captures $b$. Now consider case 3 and assume that all liberties of $b$ in unhealthy regions have been filled by $\bar{x}$. Let $R_i$, $1 \leq i \leq n$, be the healthy but not vital regions neighboring $b$. For each $R_i$, consider the blocks neighboring $R_i$ other than $b$, $NB(R_i) - \{b\}$. For each $i$, at lease one of the blocks in $NB(R_i) - \{b\}$ is not in $X$, since $R_i$ is not vital to $b$ in $X$. Consider any $b_i \in (NB(R_i) - \{b\}) - X$. By the definition of $X$, $b_i$ is not safe, so there is a sequence of moves by $\bar{x}$ capturing $b_i$. After this capture $\bar{x}$ may fill the remaining liberties of $b$ in $R_i$, $R_i \cap L(b)$, since the intersections occupied by $b_i$ supply at least one liberty for the resulting $x$ block. After all liberties for $b$ in all the $R_i$ have been filled, either $b$ is captured or else it has one vital region remaining. Since the only liberties of $b$ are in this region, it may be filled by $\bar{x}$ capturing $b$. Since $b$ has been shown to be unsafe, the contrapositive is established. ■

## 6. SAFETY IS BEST

It is of interest to consider whether $x$ would ever choose to convert a safe block into an unsafe one. The intuitive answer is no. However, whether $x$ ever obtains an advantage by this conversion depends upon the exact scoring method chosen. Unfortunately, the actual scoring scheme used by human players has so far defied accurate formalization, and we must be content with approximations. The following rule is close to that of actual human play.

RULE 3 (SCORING). At the end of the game, all unsafe blocks are removed from the board, the stones becoming prisoners of the opponent. Let $P_x$ be the total number of $\bar{x}$-stones removed from board, both due to capture by $x$ during the game and due to the previous sentence. The area for $x$, $A_x$, is the number of intersections in the empty regions completely surrounded by $x$. The score for $x$ is $A_x + P_x$. $x$ wins if $A_x + P_x > A_{\bar{x}} + P_{\bar{x}}$.

The following lemma leads directly to the conclusion that, with Rule 3 for scoring, there is never an advantage in converting safe blocks to unsafe ones.

LEMMA 5. Let $b \in B(x)$ be safe in arrangement $a_k$. $X$ must play in his own inside liberties to cause $b$ to be unsafe in subsequent arrangements.

*Proof.* Consider the set of all safe blocks, $X$. Each block in $X$ has two vital regions. To make $b$ unsafe, at least one vital region of some $b' \in X$ must be converted into a region lacking vitality or else completely removed by filling it with $x$ stones. Destroying the vitality of a region requires converting it from a healthy region to an unhealthy region. This can only be accomplished by playing in the liberties of the region. If there are $\bar{x}$-stones in the region, these may be captured by the $x$ plays, resulting in new inside liberties. This may result in an unhealthy region. If not, additional $x$ stones must be played in the resulting inside liberties until the region is removed by filling it with $x$ stones. In any case, $x$ must play on inside liberties. ■

With Rule 3 for scoring, it is clear that $x$ cannot increase his score by converting a block from safe to unsafe unless doing so enables him to capture some blocks belonging to $\bar{x}$. Since the conversion requires playing on inside liberties, the only way that these moves could increase $x$'s score is to sacrifice the block so that after $\bar{x}$ has captured it, $x$ can in turn retake more than his losses. But under best play, $\bar{x}$ will refuse the gambit, passing if necessary. Therefore, $x$ will never choose to play in such a way as to cause a safe block to be unsafe.

If Thorp and Walden's Rule 3 for scoring is used, there are situations in which $x$ chooses to play to make a safe block unsafe! The difference in scoring is that at the end of the game, no stones are removed from the board. Suppose for $b \in B(x)$ there is a healthy region containing $\bar{x}$ stones. If there are more stones than liberties in the region, it is to $x$'s advantage to fill these liberties and capture the $\bar{x}$ stones, provided this can be done without losing any of his own blocks. If there are enough $\bar{x}$ stones the arrangement after capture may contain unsafe $x$-blocks, which nonetheless cannot be captured given best play on both sides. This, together with the pathologies demonstrated in Figure 9 of Thorp and Walden's paper, combine to make our scoring rule preferable as well as more realistic. As the subsequent section shows, it is no more difficult to use in computerized Go than Thorp and Walden's scoring.

## 7. SUPPORTS

It is of interest to consider sets of $x$-blocks which mutually support one another in unconditional life. To do so, consider the collection of all unconditionally alive sets of $x$-blocks as ordered by the subset relation. This partially ordered set is a join-semilattice with a zero and a one since the union of any two unconditionally alive sets is unconditionally alive, while $\phi$ provides the zero and the set of all safe $x$-blocks provides the one. The partially ordered set is not a lattice, since the intersection of two unconditionally alive sets may not be unconditionally alive. Given any set $Y$ of safe $x$-blocks, consider the collection of all unconditionally alive sets $X$ such that $Y \subseteq X$, and there is no unconditionally alive set $Z$ such that $Y \subseteq Z \subseteq X$. Each $X$ in this collection of minimal covers for $Y$ is certainly a *mutually supporting* set, in that if any block is removed from $X$, the resulting set is not unconditionally alive. If there are several minimal covers, none of them is the largest set of "connected" mutually supporting $x$-blocks covering $Y$. We define the support of $Y$ to be this natural set of mutually supporting $x$-blocks after giving precision to the notion of connected $x$-blocks.

Blocks $b_1$ and $b_2$ are *joined* if there is a small $x$-enclosed region, $R$, such that both $b_1$ and $b_2$ neighbor $R$, $\{b_1, b_2\} \subseteq NB(R)$. The transitive closure of this relation is an equivalence relation called region-connectedness, denoted by $\equiv$. That is, $b_0 \equiv b_{n+1}$ if $b_1, \ldots, b_n \in B(x)$ such that $b_i$ is joined to $b_{i+1}$, $1 \leq i \leq n$, or if $b_0 = b_{n+1}$. Let the equivalence classes under $\equiv$ containing $b \in B(x)$ be denoted in the usual way as $[b]$ while for $Y \subseteq B(x)$, $[Y] = \cup \{[b] \mid b \in Y\} = \{b' \mid b' \equiv b, b \in Y\}$. For $Y \subseteq B(x)$, consider the largest unconditionally alive set $X$ such that $X \subseteq [Y]$. The *support* of $Y$ is $S(Y) = X \cap Y$. If every block in $Y$ is safe, then $S(Y) \subseteq X$, while if no block in $Y$ is safe then $S(Y) = \phi$. In any case, every safe block in $Y$ is in $S(Y)$.

## AN UNCONDITIONAL LIFE ALGORITHM

For any $Y \subseteq B(x)$ the algorithm implicit in this section finds the support of $Y$, as will be proved. We envision the algorithm being used on a single block, $Y = \{b\}$ , or on a set of blocks in some "area" of the board which other parts of a computer Go player find to be related in some way.

Given that small $x$-enclosed regions and $x$-blocks have been determined, the first stage of the algorithm computes $[Y]$ by any standard transitive closure algorithm such as in Knuth (1968). The second stage is specific to find the support. It proceeds by casting out unsafe blocks.

Let $Z_0 = [Y]$, $\mathscr{R}_0 = \{R \mid NB(R) \subseteq [Y], R \text{ a small } x\text{-enclosed region}\}$. For $i \geq 0$, let $Z_{i+1}$ be the set of all blocks in $Z_i$ such that $b$ has two healthy regions in $\mathscr{R}_i$.

$$Z_{i+1} = \{b \in Z_i \mid R_1, R_2 \in \mathscr{R}_i, R_1 \neq R_2, H(R_1, b) \,\&\, H(R_2, b)\}.$$

Further, let $\mathscr{R}_{i+1}$ be the set of all small $x$-enclosed regions neighboring $Z_{i+1}$, $\mathscr{R}_{i+1} = \{R \mid NB(R) \subseteq Z_{i+1}, R \text{ a small } x\text{-enclosed region}\}$. Let $Z$ be the minimal fixed point of the sequence of $Z_i$, that is $Z = Z_n$ for that $n$ such that $Z_n = Z_{n+1}$.

THEOREM 3. *$Z$ is the largest unconditionally alive set of X-blocks contained in* $[Y]$.

*Proof.* Let $b \in Z$. Then $\exists R_1, R_2, R_1 \neq R_2$ such that $H(R_1, b)$ and $H(R_2, b)$. Further, $R_1, R_2 \in \{R \mid NB(R) \subseteq Z\}$ so $R_1$ and $R_2$ are vital to $b$ in $Z$. Hence $Z$ is unconditionally alive. Consider $b \in [Y] - Z$. Now $b \in Z_0$ but for some $i$, $0 \leq i < n$, $b \in Z_i$ and $b \notin Z_{i+1}$. Therefore, $b$ does not have two healthy regions in $\mathscr{R}_i$. By the construction of $\mathscr{R}_i$, every region neighboring $b$ is in $\mathscr{R}_i$. Hence $b$ is not unconditionally alive. ∎

COROLLARY. *$S(Y) = Y \cap Z$ and $Z = S[Y]$.*

Three algorithms have been mentioned, pure game tree lookahead, lookahead distinguishing between inside and outside liberties, and computation of the support. In the order given, each algorithm requires more storage than its predecessor. However, even support computation requires only a reasonable amount of storage, so the main criterion for selecting an algorithm is the time requirement. Both support and lookahead with inside-outside liberty determination require the determination of the small $x$-enclosed regions of an arrangement. Since knowledge of these regions seems to be valuable to the computer Go player for other portions of the game analysis, we will assume the regions have been determined and ignore the pure lookahead algorithm.

Support computation requires the determination of $[Y]$ and healthy regions.

If the equivalence classes of region-connected blocks are maintained throughout the game, then $[Y]$ is obtained at little cost. Again it seems to be valuable for other portions of the game analysis to maintain these equivalence classes. By storing suitable data on liberties, computing whether a region is healthy for a block just involves a linear search of the intersections of the region. Once determined, the fact may be stored as part of the data about the region, not changing until a stone of either color is played in the region. Finally, the support computation must go through the casting out process to arrive at $Z$. In almost all practical cases, $Z = Z_n$ for $n \leq 3$, with $n = 1$ or 2 the most common. The casting out process is less time-consuming than lookahead, since each move provisionally placed on the board must be tested for legality. This suggests that the two algorithms are competitive, the exact costs dependent on their computer implementations. However, much of the data collected in the computation of $Z$ is of interest for determining intelligent plays. For example, playing in one's own inside liberties is a good play only if necessary for safety. So it is bad to play in vital regions. Also, every block in $[Y] - Z$ is in some danger of capture. Once informed of this fact, the player can devise plans to save these blocks by connecting them to the support. The most defensive way to do this is to make a few plays in the small enclosed regions joining the blocks. These considerations strongly suggest that the computation of unconditional life supports is to be preferred to the lookahead determination of safety.

*The extremely prompt review by Professor Thorp is greatly appreciated. He noticed several minor errors and suggested the desirability of Section 6.*

## REFERENCES

1. S. Dowsey, (1973), Go and the computer, *Go Review,* 13 No. 3, 72-74 (Mar 1973), [In abbreviated form in the ACM SIGART Newsletter 44 (Feb 1974)].
2. Haruyama and Nagahara (1969), *Basic Techniques of Go,* Ishi Press, Tokyo.
3. I. Ishigure (1973), *In the Beginning: The Opening in the Game of Go,* Ishi Press, Tokyo.
4. K. Iwamoto (1972), *Go for Beginners,* Ishi Press, Tokyo.
5. K. Iwamoto (1973), *The 1971 Honinbo Tournament,* Ishi Press, Tokyo.
6. Y. Kawabata (1972), *The Master of Go,* Alfred A. Knopf, New York.
7. D. E. Knuth (1968), *The Art of Computer Programming, Volume 1, Fundamental Algorithms,* Addison-Wesley, Reading, Mass.
8. K. Kosugi and J. Davies, *38 Basic Joseki,* Ishi Press, Tokyo.
9. Y. Nagahara (1972), *Strategic Concepts of Go,* Ishi Press, Tokyo.
10. E. Sakata (1971), *The Middle Game of Go,* Ishi Press, Tokyo.
11. K. Segoe (1960), *Go Proverbs Illustrated,* The Nihon Ki-in, Tokyo.
12. K. Takagawa (1958), *The Vital Points of Go,* The Nihon Ki-in, Tokyo.
13. E. Thorp and W. Walden (1964), A partial analysis of Go, *Comput. J.* 7, No. 3, 203-207.
14. E. Thorp and W. Walden (1972), A computer assisted study of Go on $M \times N$ Boards, *Information Sciences* 4, No. 1, 1-33 (Jan 1972).
15. J. Tilley, Ed. (1968?), *Go: International Handbook and Dictionary,* Ishi Press, Tokyo.

*Received August 1974; revised version received December 1974*