More Games of No Chance
MSRI Publications
Volume **42**, 2002

Go Endgames Are PSPACE-Hard

DAVID WOLFE

Abstract. In a Go *endgame*, each local area of play has a polynomial size canonical game tree and is insulated from all other local areas by live stones. Using techniques from combinatorial game theory, we prove that the Go endgame is PSPACE-hard.

1. Introduction

Go is an ancient game which has been played for several millennia throughout Asia. Although *playable* rules are relatively simple (tournament rules can include many technicalities), the game is strategically very challenging. Go is replacing Chess as the pure strategy game of choice to serve as a test-bed for artificial intelligence ideas [9] [7] [6].

Go was proved PSPACE-hard by Lichtenstein and Sipser [8], and was later proved EXPTIME-complete (Japanese rules) by Robson [12]. More recently, Crâşmaru and Tromp [5] proved that Go positions called ladders are PSPACE-complete. Yedwab conjectured that the Go endgame is hard [14]. The *endgame* occurs when each local area of play has a polynomial size canonical game tree and is insulated from all other local areas by live stones. A combinatorial game theorist will recognize this as a *sum* of simple local positions.[1]

Morris succeeded in proving that sums of small local game trees are PSPACE-complete [11]. Yedwab restricted the games to be of depth 2 [14], and Moews showed that sums of games with only three branches are NP-hard [10] [1, p. 109]. (Each of Moew's games is of the form $\{a \mid\mid b \mid c\}$ where a, b and c are integers.) Since Yedwab's and Moews' Go-like game trees depend upon scores which are exponential in magnitude (yet polynomial in the number of bits of the scores), they did not translate to polynomial sized Go positions.

Key words and phrases. Go, PSPACE, endgame, games.

[1]Another possible definition of endgame is when each play gains at most a constant number of points; we don't address that notion here. Go endgames with ko's might be outside PSPACE.

Berlekamp and Wolfe show how to analyze certain one-point Go endgames [1]. Some of these endgame positions have *values* which are linearly related to *dyadic rationals* of the form $x = \frac{m}{2^n}$. Since these endgame positions are polynomial in size in the number of *bits of* the numerator and denominator of x, their techniques can be combined with those of Yedwab and Moews to prove that the Go endgame is PSPACE-hard.

Robson observed that in Japanese rules (where repetitions of a recent position are forbidden), Go is easily seen to be in EXPTIME. However, he conjectures that according to Chinese rules (where any previous position is forbidden), Go is EXPSPACE-complete [13]. This paper fails to resolve his conjecture, but the techniques may be applicable.

2. Proof Sketch

We'll prove that the Go endgame is PSPACE-hard using a series of reductions, as suggested by Yedwab [14] shown in Figure 1.

3-QBF

$\Downarrow$ Yedwab

PARTITION GAME

$\Downarrow$ Yedwab, Moews

SWITCH PICK GAME

$\Downarrow$

FRACTIONAL SWITCH GAME

$\Downarrow$ (Yedwab, Moews)

GAME SUM

$\Downarrow$

GO ENDGAME

Figure 1. Proof outline. Yedwab and Moews reduced something like the SWITCH PICK GAME directly to GAME SUM. The FRACTIONAL SWITCH GAME is introduced here to to keep the GO ENDGAME polynomial in size.

This paper will introduce each of these problems and reductions in sequence.

3. The Artificial Games

We begin reducing from the canonical PSPACE-complete problem Quantified Boolean Formula (QBF) in conjunctive normal form with three literals per clause:

3-QBF

Instance: A formula of the form

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 \ldots \exists x_n : C_1 \wedge C_2 \wedge C_3 \wedge \cdots \wedge C_m$$

where each of the clauses C_i is a disjunct of three literals $(l_{i1} \vee l_{i2} \vee l_{i3})$, and each literal is either some variable, say x_k, or its complement, $\overline{x_k}$, for $1 \leq k \leq n$.

Question: Is the formula true?

We reduce 3-QBF to the following partition game:

PARTITION GAME

Instance: A collection of $2N$ non-negative integers X_i and $\overline{X_i}$, $1 \leq i \leq N$ and a *target* integer T.

Question: Players Left (L) and Right (R) alternate selecting numbers for inclusion in the set S. Left chooses X_1 or $\overline{X_1}$, Right chooses X_2 or $\overline{X_2}$ and so forth. At the end of the game, S will have N elements. Left wins if the elements sum to exactly T.

Lemma 0.1. *The PARTITION GAME is PSPACE-complete.*

Proof. An example of the reduction is shown in Figure 2.

From a 3-CNF formula, F, with n variables and m clauses, construct $n+12m$ pairs of integers each with $4m$ base b digits, where b is sufficiently large to prevent carries. (A choice of $b = 2n + m + 1$ will work.) The $4m$ digits are allocated as follows: One for each l_{ij} and one for each C_i. Denote by $D(C_i)$ (or $D(l_{ij})$) the value of an integer with the digit position C_i (or l_{ij}) set to 1, all other positions 0. $D(C_i)$ will be a power of b.

As in the figure, the $n + 12m$ pairs of integers are as follows:

(n variable pairs): One pair for each variable x_k:

$$\sum_{x_k = l_{ij}} D(l_{ij}) \quad \text{and} \quad \sum_{\overline{x_k} = l_{ij}} D(l_{ij}).$$

($6m$ clause pairs): Two pairs for each literal l_{ij}: Left's pair is 0 and $D(C_i) + D(l_{ij})$. Right's pair is 0 and 0.

($6m$ garbage collection pairs): Two pairs for each literal l_{ij}: Left's pair is $D(l_{ij})$ and 0. Right's pair is 0 and 0.

The sum T has a 1 in every digit position.

$$\text{(3-QBF)}\quad F = \exists x_1 \forall x_2 \exists x_3 : (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$$

	l_{11}	l_{12}	l_{13}	l_{21}	l_{22}	l_{23}			
	x_1	x_2	x_3	$\overline{x_1}$	$\overline{x_2}$	x_3	C_1	C_2	
$X_1 =$	**1**	0	0	0	0	0	0	0	
$\overline{X_1} =$	0	0	0	**1**	0	0	0	0	
$X_2 =$	0	**1**	0	0	0	0	0	0	
$\overline{X_2} =$	0	0	0	0	**1**	0	0	0	
$X_3 =$	0	0	**1**	0	0	**1**	0	0	
$\overline{X_3} =$	0	0	0	0	0	0	0	0	
$X_5 =$	**1**	0	0	0	0	0	**1**	0	$X_4 = \overline{X_4} = \overline{X_5} = 0$
$X_7 =$	0	**1**	0	0	0	0	**1**	0	$X_6 = \overline{X_6} = \overline{X_7} = 0$
$X_9 =$	0	0	**1**	0	0	0	**1**	0	$X_8 = \overline{X_8} = \overline{X_9} = 0$
$X_{11} =$	0	0	0	**1**	0	0	0	**1**	$X_{10} = \overline{X_{10}} = \overline{X_{11}} = 0$
$X_{13} =$	0	0	0	0	**1**	0	0	**1**	$X_{12} = \overline{X_{12}} = \overline{X_{13}} = 0$
$X_{15} =$	0	0	0	0	0	**1**	0	**1**	$X_{14} = \overline{X_{14}} = \overline{X_{15}} = 0$
$X_{17} =$	**1**	0	0	0	0	0	0	0	$X_{16} = \overline{X_{16}} = \overline{X_{17}} = 0$
$X_{19} =$	0	**1**	0	0	0	0	0	0	$X_{18} = \overline{X_{18}} = \overline{X_{19}} = 0$
$X_{21} =$	0	0	**1**	0	0	0	0	0	$X_{20} = \overline{X_{20}} = \overline{X_{21}} = 0$
$X_{23} =$	0	0	0	**1**	0	0	0	0	$X_{22} = \overline{X_{22}} = \overline{X_{23}} = 0$
$X_{25} =$	0	0	0	0	**1**	0	0	0	$X_{24} = \overline{X_{24}} = \overline{X_{25}} = 0$
$X_{27} =$	0	0	0	0	0	**1**	0	0	$X_{26} = \overline{X_{26}} = \overline{X_{27}} = 0$
$T =$	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	

Figure 2. A sample reduction of 3-QBF to PARTITION GAME. All integers are in base 9. All omitted X_i and $\overline{X_i}$ are chosen to be 0.

If F is satisfiable, Left should be able to win a "formula game" against Right in which Left and Right alternate selecting truth values for the variables, and Left wins if the formula evaluates to True.

This yields a strategy for Left to win the PARTITION GAME as follows. Left's challenge is to assure that exactly one "1" digit has been selected for each column C_i. First, for each variable x_k which Left selects TRUE Left chooses the integer $\overline{X_k}$. For each variable x_k which Left selects FALSE, Left chooses the integer X_k. Left interprets Right's choices of variable pairs similarly. Left then has full control over the remaining clause pairs and garbage collection pairs.

Now, since F is true, each clause has some true literal l_{ij}. Left selects the integer with the two digits, one digit corresponding to the true variable, and one digit corresponding to C_i. Left uses the garbage collection pairs to include any remaining literal columns.

The converse, that a winning strategy for the PARTITION GAME yields a winning strategy for the 3-QBF *formula game*, is similar.

4. Games and Game Sums

For a more complete introduction to combinatorial game theory, refer to [3], [4] or [1, Ch. 3]. This section briefly reviews the definitions and key results needed to get through the PSPACE-hardness reduction.

A *game* $G = \{\mathsf{G}_L \mid \mathsf{G}_R\}$ is defined recursively as a pair of sets of games G_L and G_R. The two players, named Left and Right, or Black and White (respectively), play alternately. If it is Left's move, she selects an element from G_L which then becomes the current position. Right, if it is his move, would move to one of G_R. If a player cannot move (because G_L or G_R is empty), that player loses.

In a *sum* of games $G + H$, a player can move on either G or H, leaving the other unchanged. Formally,

$$G + H = \{(\mathsf{G}_L + H) \cup (G + \mathsf{H}_L) \mid (\mathsf{G}_R + H) \cup (G + \mathsf{H}_R)\}$$

Here, a game added to a set adds the game to each element of the set. I.e.,

$$\mathsf{G}_L + H = \{G_L + H : \; G_L \in \mathsf{G}_L\}$$

In order to reduce the number of braces, we often omit them, depending on the $\mid$ to separate the Left and Right options. Also, we write $||$ as a lower precedence $\mid$. So, $A \,||\, B \mid C$ means $\{\{A\} \mid \{\{B\} \mid \{C\}\}\}$

The *negative* of the game G, $-G = \{-\mathsf{G}_R \mid -\mathsf{G}_L\}$, reverses the roles of the players.

A game $G = 0$ if the player to move (whether Left or Right) is doomed to lose (under perfect play). $G > 0$ if Left wins whether she moves first or second. $G < 0$ if Right always wins. If the first player to move can force a win, we say that G is incomparable with 0 and write $G <> 0$. Observe that these are the only four possibilities.

To compare two games, $G \geq H$ if and only if $G - H \geq 0$. Under this definition, games form a group with a partial order, where the zero of the group consists of all games $G = 0$.

Certain elements of the group, defined below, are *numbers* which (in finite games) are always dyadic rational integers. Here, n, a, and b are integers.

$$\begin{array}{rcll} 0 & = & \{\,\mid\,\} & \\ n & = & \{n{-}1 \mid \,\} & n > 0 \\ n & = & \{\, \mid n{+}1\} & n < 0 \\ \frac{a}{2^b} & = & \left\{\frac{a-1}{2^b} \;\middle|\; \frac{a+1}{2^b}\right\} & a \text{ odd}, b > 0 \end{array}$$

These games add as expected, so for instance $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$. The *number avoidance theorem* states that when playing a sum of games $G + x$ where x is a number and G is not, it is best to move on the summand G. I.e., if there exists a winning move from $G + x$, then there exists a winning move to some $G^L + x$, $G^L \in \mathsf{G}_L$.

Three more game values are relevant in this paper:

$$\begin{aligned} * &= \{0 \mid 0\} \\ \uparrow &= \{0 \mid *\} \\ \Uparrow &= \uparrow + \uparrow \end{aligned}$$

All three are *infinitesimal*, smaller than all positive numbers and larger than all negative numbers. The game $*$ is its own negative and is incomparable with 0. The game $\uparrow$ exceeds 0 but is incomparable with $*$. Lastly, $\Uparrow > *$.

5. Switch Games

A switch $\pm x$ is the game $\{x \mid -x\}$. Our games will typically involve a finale consisting of a collection of switches $\{\pm x_1, \pm x_2, \ldots, \pm x_n\}$, where $x_1 \geq x_2 \geq \cdots \geq x_n \geq 0$. Players alternately choose the largest available switch $\pm x_i$, so that after play the final score will be $x_1 - x_2 + x_3 - x_4 \cdots \pm x_n$. This final score is the *outcome* of the switch game.

Fact 1. A player can successfully choose a switch which is not the largest (and still achieve the outcome) only if the highest occurs with even multiplicity. Similarly, if the switch values are each a multiple of ε, then each time a player bypasses a large switch of odd multiplicity in lieu of a smaller switch, the player will lose at least ε but at most x_1 relative to the outcome.

SWITCH PICK GAME

INSTANCE: A target T and a set X of n pairs of switches,

$$X = \{(\pm X_1, \pm \overline{X_1}), (\pm X_2, \pm \overline{X_2}), \ldots, (\pm X_n, \pm \overline{X_n})\}.$$

All values are integers.

QUESTION: Can Left guarantee a win the following game? Left begins by selecting either $\pm X_1$ or $\pm \overline{X_1}$. Right then selects $\pm X_2$ or $\pm \overline{X_2}$. Players alternate until one from each of the n pairs of switches has been selected. Let Z be this set of n selected switches. Left wins if the switch game Z has outcome T.

Lemma 0.2. *SWITCH PICK GAME is PSPACE-complete.*

Sketch of proof (*by example*). An example of the reduction is shown in Figure 3. The reader can verify that plays on one game correspond exactly to plays on the other.

We can normalize all switches by dividing by the smallest power of 2 which exceeds all X_i, $\overline{X_i}$ and T. Letting this power be 2^β we arrive at the following PSPACE-complete variant:

PARTITION GAME

$X_1 = 3$	$\overline{X_1} = 4$
$X_2 = 2$	$\overline{X_2} = 7$
$X_3 = 1$	$\overline{X_3} = 4$
$X_4 = 0$	$\overline{X_4} = 0$
$X_5 = 1$	$\overline{X_5} = 5$

$T = 14$

SWITCH PICK GAME

$X_1 = \pm(2^{t+5} + 3)$	$\overline{X_1} = \pm(2^{t+5} + 4)$
$X_2 = \pm(2^{t+4} - 2)$	$\overline{X_2} = \pm(2^{t+4} - 7)$
$X_3 = \pm(2^{t+3} + 1)$	$\overline{X_3} = \pm(2^{t+3} + 4)$
$X_4 = \pm(2^{t+2} - 0)$	$\overline{X_4} = \pm(2^{t+2} - 0)$
$X_5 = \pm(2^{t+1} + 1)$	$\overline{X_5} = \pm(2^{t+1} + 5)$

$T = 14 + 2^{t+5} - 2^{t+4} + 2^{t+3} - 2^{t+2} + 2^{t+1}$

Figure 3. Reducing PARTITION to SWITCH PICK GAME. The quantity t is chosen to be large enough to dictate the order of play in the selected switch game, e.g., so 2^t exceeds all the PARTITION GAME's X_i and $\overline{X_i}$. Here, $t = 3$ suffices.

FRACTIONAL SWITCH GAME

Instance: The same input as SWITCH PICK GAME, but all values X_i, $\overline{X_i}$, T are dyadic rationals less than 1 and all are an even multiple of $2^{-\beta}$.

Question: Can Left win this game as in the SWITCH PICK GAME?

We will now show that the following problem, GAME-SUM, is PSPACE-hard by reduction from FRACTIONAL SWITCH GAME. The quantity $k = 6\beta + 2$ is chosen to be sufficiently large to admit construction of the game sum on a Go board.

GAME-SUM

Instance: A sum of games S each of the form $a \,||\, b \mid c$ or $a \mid b \,||\, c$ for dyadic rationals $a \geq b \geq c$. Each of these numbers has a polynomial number of bits, and has an integer part which is polynomial in magnitude. Furthermore $a - b > k$ and $b - c > k$.

Question: Can Left force a win moving first on S?

The constraints on a, b and c will make translation to a Go board possible.

For a given instance of FRACTIONAL SWITCH GAME (n, X and T), we will construct a game sum with several components:

$$S = \sum G_i + \sum \overline{G_i} + \sum H_i + I + \overline{I} + \Uparrow$$

The *temperatures*[2] of all components will be sufficiently far apart to almost assure that players must play on the hottest available summand to have a hope of winning. We'd like to force the order of play to be:

[2]The terms "temperature" and "hot" are technical concepts of combinatorial game theory. Their inclusion in this paper will aid the intuition of the combinatorial game theorist. Other readers can safely ignore the terms.

$$\begin{array}{c} G_1 \text{ and } \overline{G_1} \text{ (in either order)} \\ H_1 \\ G_2 \text{ and } \overline{G_2} \\ H_2 \\ \vdots \\ G_{n-1} \text{ and } \overline{G_{n-1}} \\ H_{n-1} \\ G_n \text{ and } \overline{G_n} \\ I \\ \overline{I} \end{array}$$

To achieve this, G_i and $\overline{G_i}$ will have equal temperatures $t_i = (2n - 2i + 4)k$ which decrease with increasing i. H_i will have temperature $t_i^- = t_i - k$. In truth, the first player will be able to play H_1 before G_1 or $\overline{G_1}$, but we'll see that this will be no better than playing according to the agenda.

If i is odd,

$$\begin{aligned} G_i &= t_i \;\|\; -t_i + X_i + k \;|\; -t_i - X_i - k \\ \overline{G_i} &= t_i \;\|\; -t_i + \overline{X_i} + k \;|\; -t_i - \overline{X_i} - k \\ H_i &= 0 \;\|\; -2t_i^- \;|\; -2t_i^- \\ &= 0 \;\|\; -2t_i^- * \end{aligned}$$

If i is even,

$$\begin{aligned} G_i &= t_i + X_i + k \;|\; t_i - X_i - k \;\|\; -t_i \\ \overline{G_i} &= t_i + \overline{X_i} + k \;|\; t_i - \overline{X_i} - k \;\|\; -t_i \\ H_i &= 2t_i^- * \;\|\; 0 \quad \text{(except omit } H_n) \end{aligned}$$

Left chooses to play on either G_1 or $\overline{G_1}$ and Right is compelled to play on the other. Thus, Left decides whether $\pm X_1$ or $\pm\overline{X_1}$ is included in the switch game Z. Left is then compelled to play H_1 to 0. If Left were to play on H_1 before G_1 or $\overline{G_1}$, this is tantamount to giving Right the option as to whether $\pm X_1$ or $\pm\overline{X_1}$ is included in Z. Play on a later game such as G_2 or one of the $\pm X_i$ is too costly as a consequence of Fact 1. The amount gained by such a maneuver is at most the largest X_1, and this is less than the amount lost which is at least k.

Next, Right selects whether $\pm X_2$ or $\pm\overline{X_2}$ is included and Left and Right continue alternating until the last two plays of this phase on G_n and $\overline{G_n}$ (since we omitted H_n). It is now Right's turn, who is free to choose between the games:

$$\begin{aligned} I &= k \,|\, -T \;\|\; -2k \\ \overline{I} &= T + 2k \;\|\; * \end{aligned}$$

and Left will reply on the other. If the switch game Z is lost by Left because the alternating sum is less than T, Right will choose to play $\overline{I}$, Left will reply on I, and Right will move I to $-T$, which is sufficient to win out over the alternating sum. If, however, Z is lost by Left because the alternating sum exceeds T, Right will choose to play I to $-2k$, Left will move $\overline{I}$ to $T + 2k$, and Right will be the first to play on the alternating sum.

At this point, the players will play the switch game $\{(\pm(k+X_i), \pm(k+\overline{X_i}))\}$. Since there are an even number of terms, the alternating sum will have the same outcome as $\{(\pm X_i, \pm\overline{X_i})\}$; the k's will cancel. If this outcome is exactly T, then however Right played I and $\overline{I}$, Left can play the switches until the whole total is 0 or $*$. The infinitesimal $\Uparrow$ included in S is then sufficiently large to assure Left's win. Thus, Left can win if and only if she can arrange that the alternating sum adds to exactly the number T.

6. Switch games in Go

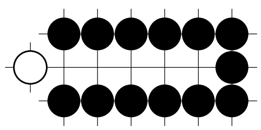
The following *warming* operator is a special case of the Norton product of two games "$g.h$" [3, p. 246], or of the overheating operator defined in [2], "$\int_s^t g$". If $g = \{g^L \mid g^R\}$, define

$$\int g \stackrel{\text{def}}{=} \int_{1*}^{1} g = g.1* = \begin{cases} g & \text{if } G \text{ is an even integer,} \\ g + * & \text{if } G \text{ is an odd integer,} \\ \{1 + \int g^L \mid -1 + \int g^R\} & \text{otherwise.} \end{cases}$$

As a consequence of being a Norton multiple, the warming operator has the following properties [3, p. 246]:

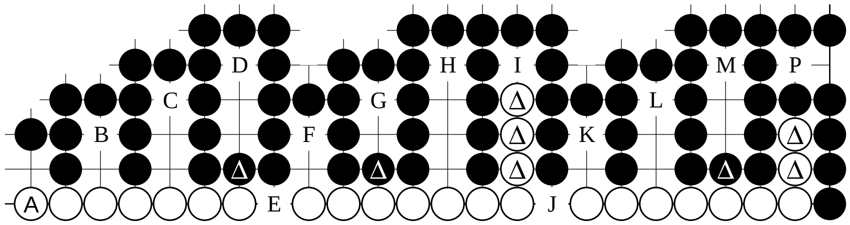
1. linearity: $\int g + \int h = \int (g + h)$,
2. order preserving: If $g \geq h$ then $\int g \geq \int h$.

The following *blocked corridor* in Go has value $n - 2 + \int 2^{1-n}$, where n is the number of empty nodes in the corridor [1]. (In this example, $n = 5$.)



$$= 3 + \int \frac{1}{16} = \{4 \mid\mid\mid\mid 3 \mid\mid\mid 2 \mid\mid 1 \mid *\}$$

Since the warming operator is linear and order preserving, it suffices to convert the sum of abstract games to Go positions of value $\int\{a \mid\mid b \mid c\}$, where $b - a \geq k$, $c - b > k$ and a, b and c are multiples of $\frac{1}{2^\beta}$. I'll describe the conversion by example. We'll first analyze the following specific position, and then argue that the position can be augmented to achieve any $\int\{a \mid\mid b \mid c\}$.



The white group including stone A is assumed to be alive, and all black stones are alive. Black has 2 sure points in region P, and there is one zero point play (or *dame*, "dah-meh") above A. Together these are worth $2*$. In addition, if Black plays at E, Black captures 56 points to the right of E. The blocked corridor at B is worth $\int \frac{1}{2}$, C is worth $1 \int \frac{1}{4}$ (i.e., $1 + \int \frac{1}{4}$) and area D is 3 points of territory. These total to $60 \int \frac{3}{4}$. If White plays at E and Black replies at J, Black nets $34 * \int \frac{11}{8}$. The $*$ represents the zero point play (or *dame*) at I. Lastly, if White plays at E and J, the resulting position is worth $12 * \int \frac{17}{8}$. Thus the original position is worth

$$2* \quad + \quad \left\{ 60 \int \frac{3}{4} \;\middle\|\; 35 * \int \frac{11}{8} \;\middle|\; 12 * \int \frac{17}{8} \right\}$$

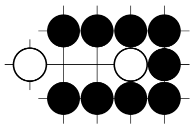
which, by applying the definition of $\int$ is

$$= 2 * + * + \int \left\{ 59\frac{3}{4} \;\middle\|\; 35\frac{3}{8} \;\middle|\; 16\frac{1}{8} \right\} = \int \left\{ 61\frac{3}{4} \;\middle\|\; 37\frac{3}{8} \;\middle|\; 18\frac{1}{8} \right\}$$

An informal recipe should convince the reader that there are enough degrees of freedom to generate any Go position $\int\{a \parallel b \mid c\}$ (where a, b and c are constrained as above):

1. A group (A) invading β corridors of increasing length. (In the example, $\beta = 3$ and the corridors are marked B, C and D.) The binary expansion of the fractional part $a - \lfloor a \rfloor$ dictates which of these β corridors are blocked. (Here, $a - \lfloor a \rfloor = .110$, so only the third corridor is blocked.)
2. A second group invading β corridors which threatens to connect to A. These corridors are blocked according to the quantity $b - a - \lfloor b - a \rfloor$. The third group's corridors should account for $c - b - a - \lfloor c - b - a \rfloor$.)
3. Adjustments to the integer differences $\lfloor b - a \rfloor$ and $\lfloor c - b \rfloor$ are made by extending the White stones marked Δ. Each additional stone adds two points; an empty node at, say, I, adds one.
4. Lastly, shift the value of the Go position by any integer by adding territory to Black or White (area P). Include a dame (say, at the point above A) as needed to adjust by $*$.

The construction of one switch $\int\{a \mid\mid b \mid c\}$ requires a number of White stones which is linear in β. The choice of $k = 6\beta + 2$ suffices. Using the fact that



$$= \int \uparrow = \int \{0 \mid *\},$$

the last games $\int \overline{I} = \int\{T + 2k \mid\mid *\}$ and $\int \Uparrow$ are also constructible on a Go board.

References

[1] Elwyn Berlekamp and David Wolfe. *Mathematical Go: Chilling Gets the Last Point*. A K Peters, Ltd., Wellesley, Massachusetts, 1994.

[2] Elwyn R. Berlekamp. Blockbusting and domineering. *Journal of Combinatorial Theory*, 49(1):67–116, September 1988.

[3] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways*. Academic Press, New York, 1982.

[4] John H. Conway. *On Numbers and Games*. Academic Press, London/New York, 1976.

[5] Marcel Crâşmaru and John Tromp. Ladders are pspace-complete. In T. A. Marsland and I. Frank, editors, *Computers and Games: Second International Conference, CG 2000, Hamamatsu, Japan, October 2000*, pages 241–249. Springer, 2001.

[6] George Johnson. To test a powerful computer, play an ancient game. *New York Times*, July 29, 1997.

[7] Anders Kierulf. *Smart Game Board: a Workbench for Game-Playing Programs, with Go and Othello as Case Studies.* PhD thesis, Swiss Federal Institute of Technology (ETH) Zürich, 1990. Nr. 9135.

[8] David Lichtenstein and Michael Sipser. Go is polynomial-space hard. *Journal of the Association for Computing Machinery*, 27(2):393–401, April 1980.

[9] J. McCarthy. Chess as the drosophila of AI. In T. A. Marsland and J. Schaeffer, editors, *Computers, Chess, and Cognition*, pages 227–237. Springer Verlag, New York, 1990. Other games replacing chess in AI.

[10] David Moews. *On Some Combinatorial Games Related to Go*. PhD thesis, University of California, Berkeley, 1993.

[11] F. L. Morris. Playing disjunctive sums is polynomial space complete. *International Journal of Game Theory*, 10:195–205, 1981.

[12] J. M. Robson. The complexity of Go. In *Information Processing; proceedings of IFIP Congress*, pages 413–417, 1983.

[13] J. M. Robson. Combinatorial games with exponential space complete decision problems. In *Lecture Notes in Computer Science. Mathematical Foundations of Computer Science 1984*, pages 498–506. Springer-Verlag, 1984.

[14] Laura Yedwab. On playing well in a sum of games. Master's thesis, M.I.T., August 1985. MIT/LCS/TR-348.

David Wolfe
Department of Mathematics and Computer Science
Gustavus Adolphus College
800 West College Avenue
Saint Peter, MN 56082
United States
wolfe@gustavus.edu
http://www.gustavus.edu/~wolfe