On the Complexity of Tsume-Go

Marcel Crâşmaru

SCVR Vatra Dornei 5975 SV, ROMANIA,
mi@assist.cccis.ro

Abstract. In this paper, we explain why Go is hard to be programmed. Since the strategy of the game is closely related to the concept of *alive–dead* group, it is plainly necessary to analyze this concept. For this a mathematical model is proposed. Then we turn our research to Tsume-Go problems in which one of the players has always a unique good move and the other has always only two good moves available to choose from. We show that this kind of problems are NP-*complete*.

1 Introduction

The creation of a program that plays the game of Go, even at a medium level, is known to be difficult. Our aim in this paper is to give a reasonable explanation to this problem by analyzing the fundamentals of Go, more precisely, the concept of *living group*. What territories are, is a problem that can be reduced eventually to the life and death problem (tsume-Go) of certain groups placed on the board. Strong players always look to groups to find out if they are thick or weak, that is if they can live easily or not. In fact, the core of the strategy of the game is the understanding of what is a living or a dead group.

The Go-programmer dream is to find a good pruning function that would reduce a lot the searching tree in the emerging problems during the game, problems that can be reduced, as we said before, to life and death problems. What we will show is that even if one has an exceptionally good pruning function, there are positions in which the complexity of the search is NP-complete.

In the past, results on the complexity of Go were obtained by Lichtenstein and Sipser [1], who have constructed Go positions without kos which are PSPACE-hard. Using kos, Robson [3] has shown Go is EXPTIME-complete. On the other hand, Morris [2] showed that playing sums of relatively simple combinatorial games is NP-hard and then, Yedwab [4] and Moews [5] showed that even a sum of games of the form $a||b|c$ is NP-hard. For further references see Berlekamp [6].

Let $\mathcal{G}$ be a group and $\mathcal{T}(\mathcal{G})$ a tree constructed recursively as follows: the root is the (initial) position to which $\mathcal{G}$ belongs, and the children of any node v are all the positions obtained throw a legal move from v. We also denote by $\Psi(v)$ a pruning function for $\mathcal{T}(\mathcal{G})$, that is, a set containing some descendents of the node v. Let $\mathcal{T}_\Psi(G)$ be the subtree generated by $\Psi(v)$ with the same root as $\mathcal{T}(\mathcal{G})$ and then, recursively, $\Psi(v)$ is the set of children of any node v.

Let G_{NP} be the set of all groups $\mathcal{G}$ for which there is a pruning function $\Psi(v)$, computable in polynomial time, that satisfies $|\Psi(v)| = 1$ always for one of

H.J. van den Herik, H. Iida (Eds.): CG'98, LNCS 1558, pp. 222–231, 1999.

the players and $|\Psi(v)| = 2$ for the other for all nodes v of $\mathcal{T}_\Psi(G)$ and one can determine if $\mathcal{G}$ can be removed or not from the board only by scanning in $\mathcal{T}_\Psi(G)$ at a depth which is polynomial of the size of the board.

Let us consider the problem:

(GNP) Given $\mathcal{G} \in G_{\mathrm{NP}}$, decide whether $\mathcal{G}$ is a dead group.

Our main result is:

Theorem 1. *The problem* (GNP) *is* NP−*complete.*

In Sect. 1 we present a mathematical model of the game of Go, in the second we give an algorithm that checks if a group is alive or not, while in the last we prove Theorem 1.

2 A Mathematical Model of Go without the Rule of Ko

Let $\mathcal{T}$, be a nonempty finite set, which we call the *board*, and $\mathcal{N} \subset \mathcal{T} \times \mathcal{T}$ a symmetric binary relation on $\mathcal{T}$, named the *neighboring relation*. If $(x, y) \in \mathcal{N}$ we say that x and y are *neighbors* and write $x\,\mathcal{N}\,y$. Let $\mathcal{S} = \{B, W, 0\}$ the set of *colors* (B stands for Black, W for White and 0 for an empty) and $\mathcal{S}^* = \{B, W\}$, the set of *players*. If $x \in \mathcal{S}^*$ we denote by $\overline{x}$ the other element of $\mathcal{S}^*$.

Let $\mathcal{F}$ be the set of *configurations* which are defined to be the set of all functions from $\mathcal{T}$ to $\mathcal{S}$. (For example, the fact that $f(x) = W$ means that in x a white stone is placed). For each configuration f, it is natural to associate a binary relation $\mathcal{N}_f \subset \mathcal{T} \times \mathcal{T}$. Thus, we write $x\,\mathcal{N}_f\,y$ iff $x\,\mathcal{N}\,y$ and $f(x) = f(y)$. It is easy to see that the reflexive and transitive closure of $\mathcal{N}_f$ is an equivalence relation and its classes of equivalence will be called *groups*. We write by $\mathcal{G}_f^x$ the group that contains x. The color of the group $\mathcal{G}_f^x$ is $Color(\mathcal{G}_f^x) = f(x)$.

For any Black or White group we define the set of *liberties* by

$$\mathcal{L}_f^x = \{y \in \mathcal{T} \colon f(y) = 0 \ \text{ and } \ y\,\mathcal{N}\,z \ \text{ for some } \ z \in \mathcal{G}_f^x\} \ ,$$

that is the empty neighbors of $\mathcal{G}_f^x$. The number of liberties of $\mathcal{G}_f^x$ is the cardinal of $\mathcal{L}_f^x$ and will be denoted by $\|\mathcal{G}_f^x\|$.

During the game, any configuration is paired with a player whose turn is make the next move, thus we need to introduce the set of *positions* which is defined to be $\mathcal{P} = \mathcal{F} \times \mathcal{S}^*$. The fact that $p = (f, a) \in \mathcal{P}$ means that in the configuration f is a's turn to move and for a group that contains x in position p, the notation $\mathcal{G}_p^x$ will be used rather than $\mathcal{G}_f^x$.

Now we are ready to define the *moving function*, which we denote by $\phi \colon \mathcal{P} \times \mathcal{T}^* \to \mathcal{P}$. Let $\mathcal{T}^* = \mathcal{T} \cup \{pass\}$. If $p, p' \in \mathcal{P}$ and $x \in \mathcal{T}^*$ then $p' = \phi(p, x) = \phi\big((f, a), x\big)$ will mean that in the position p the player a has placed his stone in x and the outcome is the position p'.

There are two cases.

1. $x = pass$ or $f(x) \neq 0$. We define

$$\phi\big((f,a),x\big) = (f,\overline{a}) \ ,$$

which means that if one of the players moves *pass* then we get the same configuration with the other player's turn to move and playing on an occupied place is the same as playing *pass*.

Let us note that this also says that to play *pass* is allowed in the initial position $p_0 = (f_0, B)$, where $f_0(x) = 0$ for any $x \in \mathcal{T}$.

2. $f(x) = 0$. In this case, we may obtain or not a different configuration if the move captures or not some stones.

Let

$$g(z) = \begin{cases} f(z) & \text{if} \quad z \neq x \\ a & \text{if} \quad z = x \end{cases}$$

be the configuration obtained by placing a stone of color a on x. Let

$$Capture(p,x) = \{\mathcal{G}_g^z\colon \ g(z) = \overline{a}, \ \ z \mathcal{N}\, x \ \text{ and } \ \|\mathcal{G}_g^z\| = 0\}$$

be the groups captured by the move, that is the opposite color groups which are neighbors to x and have no liberties.

If $Capture(p,x) \neq \emptyset$, then let

$$h(z) = \begin{cases} g(z) & \text{if} \quad z \notin Capture(p,x) \\ 0 & \text{if} \quad z \in Capture(p,x) \end{cases}$$

be the configuration obtained from g after removing the groups with no liberties.

Then, since the suicide is forbidden, we define:

$$\phi\big((f,a),x\big) = \begin{cases} (f,\overline{a}) & \text{if} \quad Capture(p,x) = \emptyset \ \text{ and } \ \|\mathcal{G}_g^x\| = 0 \\ (h,\overline{a}) & \text{if} \quad Capture(p,x) \neq \emptyset \\ (g,\overline{a}) & \text{if} \quad Capture(p,x) = \emptyset \ \text{ and } \ \|\mathcal{G}_g^x\| \neq 0 \end{cases} .$$

Given $p_1, \ldots, p_n \in \mathcal{P}$, let $\Psi(\{p_1, \ldots, p_n\})$ be the set of all positions that can be obtained from $p_1, \ldots, p_n$ by playing any possible move. Thus, $\Psi\colon 2^{\mathcal{P}} \to 2^{\mathcal{P}}$ and

$$\Psi\big(\{p_1, \ldots, p_n\}\big) = \bigcup_{i=1}^{n} \bigcup_{x \in \mathcal{T}^*} \phi(p_i, x) \ ,$$

where $2^{\mathcal{P}}$ denotes the set of all subsets of $\mathcal{P}$. In particular, $\Psi^{(k)}\big(\{p_0\}\big)$ is the set of all positions that can be obtained from the initial position p_0 in exactly k moves. If the position p' is obtained from p in k moves we write shortly $p \overset{k}{\Longrightarrow} p'$.

The following theorem shows that the "history" that created a position plays no role in the study of a legal position of the game.

Theorem 2. *Let $G(p_0)$ be the set of all positions generated by p_0. Then, $p = (f, a) \in G(p_0)$ iff $\|\mathcal{G}_p^x\| \neq 0$ for any $x \in \mathcal{T}$ with $f(x) \neq 0$.*

Proof. Clearly p_0 has the property that $\|\mathcal{G}_{p_0}^x\| \neq 0$ for any $x \in \mathcal{T}$ with $f_0(x) \neq 0$. Suppose now that for the position $p = (f, a)$ we have that $\|\mathcal{G}_p^x\| \neq 0$ for any $x \in \mathcal{T}$ with $f(x) \neq 0$. Then, for any position p' with $p \stackrel{1}{\Longrightarrow} p'$, we have that $\|\mathcal{G}_{p'}^x\| \neq 0$ because Ψ removes the groups without liberties. By induction, we obtain that for any $p \in G(p_0)$ we have that $\|\mathcal{G}_p^x\| \neq 0$ for any $x \in \mathcal{T}$ with $f(x) \neq 0$.

To prove the other implication, let $p = (f, a)$ be a position and let $x_1, \ldots, x_{|\mathcal{T}|}$ be an ordering of $\mathcal{T}$ such that the sequence $f(x_1), \ldots, f(x_{|\mathcal{T}|})$ has the form

$$\underbrace{B, W, B, W, \ldots, B, W}_{l \text{ colors}}, \underbrace{X, \ldots, X}_{m \text{ colors}}, \underbrace{0, \ldots, 0}_{n \text{ colors}} \ ,$$

for some $l, m, n \geq 0$, $l + m + n = |\mathcal{T}|$, and $X \in \mathcal{S}^*$. Then $p_0 \stackrel{k}{\Longrightarrow} p$ through $x_1, \ldots, x_{|\mathcal{T}|}$ in $k \leq l + 2m + 1$ steps (the step from W to X may need a *pass* move, from X to X requires a *pass* move and from X to a may need a *pass* move). Since $\|\mathcal{G}_p^x\| \neq 0$ for any $x \in \mathcal{T}$ with $f(x) \neq 0$ it follows that the same is true for all the intermediary positions between p_0 and p, therefore all these positions are correctly constructed (without suicidal moves). Although is not necessary, note that the way in which p is obtained from p_0 takes a minimum number of steps. This completes the proof of the theorem.

3 The Life and Death Algorithm

By the definition of the function move, one can see that if the group $\mathcal{G}_p^x$ has only one liberty (i.e. $\|\mathcal{G}_p^x\| = 1$) where $p = (f, a)$ and $Color(\mathcal{G}_p^x) = \overline{a}$ then the group $\mathcal{G}_p^x$ can be captured by taking that liberty. Let then

$$\mathcal{D}_0 = \left\{ \mathcal{G}_p^x \colon p \in G(p_0),\ p = (f, a),\ \ Color(\mathcal{G}_p^x) = \overline{a} \ \text{ and } \ \|\mathcal{G}_p^x\| = 1 \right\} \ .$$

be the set of groups which can be captured in one move. For any $i \geq 1$, we define recursively the sets $\mathcal{D}_i$ of groups that can be captured in at most $i+1$ moves. For this, let us note that Go players have different points of view on their groups. Thus, the player a who has a group $\mathcal{G}_p^x$, will convince himself that the group is dead if for all his possible moves he will get a dead group, and the opponent $\overline{a}$ needs to find only one move to kill it. This leads to the following definition

$$\begin{aligned} \mathcal{D}_i = \mathcal{D}_{i-1} \cup \big\{ \mathcal{G}_p^x \colon p \in G(p_0),\ & p = (f, a),\ \ Color(\mathcal{G}_p^x) = \overline{a} \\ & \text{and } \ \mathcal{G}_{p'}^x \in \mathcal{D}_{i-1} \text{ for some } \ p' \in \Psi\big(\{p\}\big) \big\} \\ \cup \big\{ \mathcal{G}_p^x \colon p \in G(p_0),\ & p = (f, a),\ \ Color(\mathcal{G}_p^x) = a \\ & \text{and } \ \mathcal{G}_{p'}^x \in \mathcal{D}_{i-1} \text{ for every } \ p' \in \Psi\big(\{p\}\big) \big\} \ . \end{aligned}$$

Now, we are ready to define the set of all *dead groups* by

$$\mathcal{D} = \bigcup_{i \in \mathbb{N}} \mathcal{D}_i$$

and the set of *living groups* by

$$\mathcal{V} = \left\{ \mathcal{G}_p^x \colon p \in G(p_0),\ \mathcal{G}_p^x \notin \mathcal{D} \right\} \quad .$$

Since $\mathcal{P}$ is finite and the set of groups in a certain position is also finite, it follows that $\mathcal{D}$ is finite. This implies that there is an integer, call it α, for which $\mathcal{D}_\alpha = \mathcal{D}_{\alpha+1}$, therefore, by the definition of $\mathcal{D}$ it follows that $\mathcal{D} = \mathcal{D}_\alpha$. For α we may take the trivial upper bound $2|\mathcal{T}|3^{|\mathcal{T}|}$.

The following is an algorithm that analyses if a group can be killed or not.

```
procedure Can_kill(p : G(p0), x : T, depth : N) : boolean
var w:boolean; Y ∈ 2^{G(p0)};
begin
  if G_p^x ∈ D0 then return TRUE;
  if depth > α then return FALSE;
  if (p = (f, a) AND Color(G_p^x) ≠ a)
      w =FALSE; Y = Ψ(p);
      while (Y ≠ ∅ AND NOT w)
            p' ∈ Y; Y = Y \ {p'};
            w = w OR Can_kill(p', x, depth + 1);
      endwhile
      return w;
  else
      w =TRUE; Y = Ψ(p);
      while (Y ≠ ∅ AND w)
            p' ∈ Y; Y = Y \ {p'};
            w = w AND Can_kill(p', x, depth + 1);
      endwhile
      return w;
  endif
end.
```

The next theorem proves that this algorithm is correct.

Theorem 3. *$\mathcal{G}_p^x \in \mathcal{D}$ iff $Can\_kill(p, x, 0) =$ TRUE.*

Proof. Since trivially $|G(p_0)| < 2 \cdot 3^{|\mathcal{T}|}$, the algorithm stops eventually. By induction on i we immediately see that $\mathcal{G}_p^x \in \mathcal{D}_i$ implies that $Can\_kill(p, x, 0) =$ TRUE, and by the definition of $\mathcal{D}$ it follows that the same is true for $\mathcal{D}$.

Conversely, let $Can\_kill_n(p, x, 0)$ be the above algorithm with α replaced by n. It is not difficult to prove by induction on n that $Can\_kill_n(p, x, 0) = \texttt{TRUE}$ implies $\mathcal{G}_p^x \in \mathcal{D}_n$ for any $n \geq 0$. This completes the proof of the theorem.

Let us remark that to check that $\mathcal{G}_p^x \in \mathcal{D}_0$ takes $O(|\mathcal{T}|)$ steps. The calculation of Ψ needs $O(|\mathcal{T}|)$ steps, too. We also mention that if the depth of the calculation is d then $O(d \cdot |\mathcal{T}|)$ is the memory space needed.

We conclude this section with some observations on kos. So far no restrictions on the cycling of positions was imposed in our mathematical model. Thus, since repetitions were allowed the set $\mathcal{V}$ of living groups is larger than in an usual game with the rule of ko.

For example, in Fig. 1 and 2, if White moves then his bigger groups are alive, although in a normal game the first is dead while the second is in an undetermined position and special extra rules apply. Since our main object in this work is to prove the complexity of a certain set of groups in which, as we will next see, no kos are involved, we don't go in further detail with the ko rule.

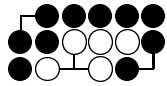

Fig. 1. A 2-kos position in which White lives.

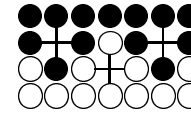

Fig. 2. In a 3-kos position both Black and White live.

4 Proof of Theorem 1

The proof has two parts. Firstly we show that GNP is a NP−problem and secondly we reduce the 3 SAT problem to the life and death problem of certain groups from G_{NP}. The theorem will then follow, since 3 SAT is NP complete, result proved by Cook in 1971.

We begin by showing that GNP belongs to NP. For this, let us consider a nondeterministic Turing machine which performs the following algorithm:

```
input G_p^x ∈ G_NP
guess p_1, ..., p_|T| with p = p_1
check that p_{i+1} ∈ Ψ(p_i) for 2 ≤ i ≤ |T| − 1
if (Color(G_p^x) = a AND |Ψ({f, a})| = 2)
      check that G_p^x ∉ D_0 for every p_i
      if this holds then return FALSE
else       // Color(G_p^x) = a AND |Ψ({f, ā})| = 2
      check that G_p^x ∈ D_0 for some p_i
      if this holds then return TRUE
endif
end
```

The computation of this algorithm takes at most $O(|\mathcal{T}|^2)$ steps.

For the second part of the proof, we show how the 3 SAT problem can be reduced to the life and death problem of certain groups. We first state the 3 SAT problem. Let us consider the formula $F = c_1 \wedge \cdots \wedge c_n$ where $c_j = \tilde{u}_{i_1} \vee \tilde{u}_{i_2} \vee \tilde{u}_{i_3}$, $\tilde{u}_{i_k} \in \{u_{i_k}, \overline{u_{i_k}}\}$ for $1 \leq i \leq n$ and $1 \leq j \leq n$ and $\mathcal{U} = \{u_1, \ldots, u_m\}$ is the set of logical variables that appear in F. Then the 3 SAT problem is:

(3 SAT) *Decide whether F evaluates to true.*

Let $p(i)$ and $q(i)$ be the number of appearances in F of u_i and $\overline{u_i}$ respectively. For each i, $(1 \leq i \leq m)$ we construct the diagrams A(i) (see Fig. 3).

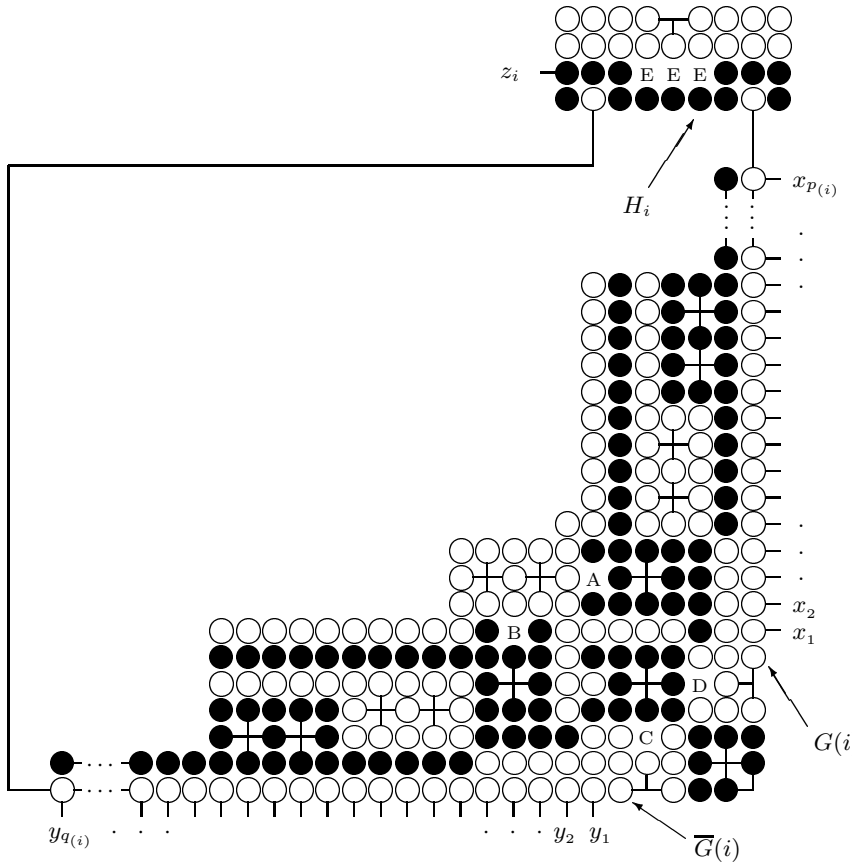

Fig. 3. Diagram A(i).

Remark 1. Suppose that in diagram A(i), White moves. Then:

1. $\|G(i)\| = 2$, $\|\overline{G}(i)\| = 2$ and $\|H_i\| = 3$.
2. If White plays A and B then H_i is dead.

3. If White plays A and Black plays C then $G(i)$ and H_i are alive and $\overline{G}(i)$ is dead.

4. If White plays B and Black plays D then $\overline{G}(i)$ and H_i are alive and $G(i)$ is dead.

For any $c_j = \tilde{u}_{i_1} \vee \tilde{u}_{i_2} \vee \tilde{u}_{i_3}$, $1 \leq j \leq n$, we associate a diagram B(j) (see Fig. 4). For any k, $1 \leq k \leq 3$, this diagram is connected with that from diagram A(i_k) as follows. If $\tilde{u}_{i_k} = u_{i_k}$ then $\tilde{G}(i_k)$ is $G(i_k)$, the connection being through x_l for some l, $1 \leq l \leq p(i_k)$. If $\tilde{u}_{i_k} = \overline{u_{i_k}}$ then $\tilde{G}(i_k)$ is $\overline{G}(i_k)$, the connection being through y_l for some l, $1 \leq l \leq q(i_k)$.

Remark 2. In diagram B(j) the following hold true:

1. $\|\tilde{G}(i_l)\| = 2$ and $\|L(i_l)\| = 3$ for $1 \leq l \leq 3$.
2. The group $K(j)$ is alive iff the group $\tilde{G}(i_l)$ alive for some l, $1 \leq l \leq 3$.

Finally, all the previous diagrams are interconnected also with diagram C (see Fig. 5). Thus, all the groups H_i, are connected to those from diagram A(i) through z_i for $1 \leq i \leq m$. Similarly, all the groups K_j, are connected to those from diagram B(j) through w_j for $1 \leq j \leq n$.

Remark 3. Suppose that in diagram C, White moves. Then we have:

1. $\|\Lambda\| = \|\Omega\| = 2$.
2. The group Λ is alive iff either H_i is dead for some i, $1 \leq i \leq m$ or K_j is alive for all j, $1 \leq j \leq n$.

The construction of all these diagrams needs $O(n)$ steps, so the reduction is made in a polynomial number of steps.

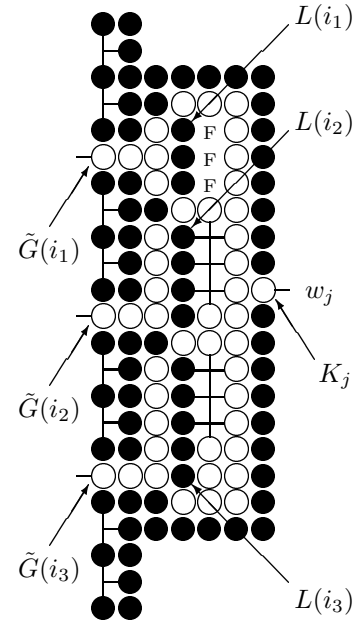

Fig. 4. Diagram B(j).

Remark 4. (Easy life). For White, an easy way to live with Λ is by moving A (or B) in diagram $A(i)$ and Black plays poorly by failing to respond with C (or D) in the same diagram. This allows White to occupy both A and B and then, by 2 of Remark 1 H_i is dead, so Λ is alive by 2 of Remark 3.

Let us now see that the 3 SAT problem is equivalent to the life and death problem of the group Λ from diagram C. In the position given by the above construction suppose that White moves. We denote by $v(u_i) \in \{\texttt{TRUE}, \texttt{FALSE}\}$ any assignation of the logical variables u_i, $1 \leq i \leq m$. The correspondence between our construction and $v(u_i)$ is given by the convention:

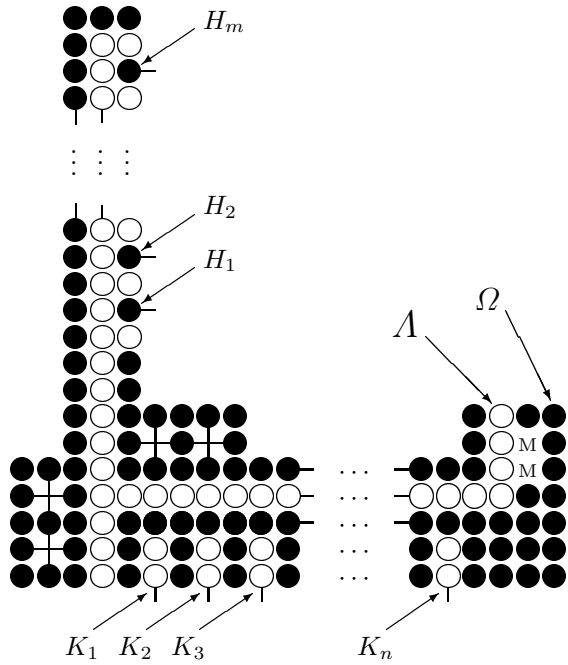

Fig. 5. Diagram C.

$$v(u_i) = \begin{cases} \texttt{TRUE} & \text{if } G(i) \text{ is alive} \\ \texttt{FALSE} & \text{if } \overline{G}(i) \text{ is alive} \end{cases} . \tag{1}$$

We claim that any solution of the 3 SAT problem makes Λ alive and the converse is also true.

Let $v(u_i) \in \{\texttt{TRUE}, \texttt{FALSE}\}$ for $1 \leq i \leq m$ be an assignation which validates F. If $v(u_1) = \texttt{TRUE}$ then White moves in A in diagram $A(1)$ so Black is forced to respond C (otherwise Λ lives easily, confer Remark 4). If $v(u_1) = \texttt{FALSE}$ then White moves in B in diagram $A(1)$ so Black is forced to respond D (otherwise Λ lives easily, confer Remark 4). White will then turn to the subsequent diagrams for $i = 2, \ldots, m$ and play in the same way, giving Black only unique-choice moves. Playing in this manner yields

$$v(\tilde{u}_i) = \texttt{TRUE} \qquad \text{iff} \qquad \tilde{G} \text{ is alive for } i = 1, \ldots, m \ . \tag{2}$$

Let us also note that White moves make K_j alive for $j = 1, \ldots, n$. For if K_j is dead for some j, then by 2 of Remark 2 the groups $\tilde{G}(i_l)$ from diagram $B(j)$ are all dead. By (2) it follows that $v(\tilde{u}_{i_l}) = \texttt{FALSE}$ for $l = 1, 2, 3$, so c_j is $\texttt{FALSE}$ which implies that F is $\texttt{FALSE}$, contradicting our assumption. Since K_j are alive for $j = 1, \ldots, n$, by 2 of Remark 3 it follows that Λ is alive.

Conversely, suppose that Λ is alive. As we observed before in Remark 4, Λ can live easily if Black plays poorly. Suppose that Black plays well. Then by 2 of Remark 3, the only way for White to live with Λ is to live with all K_j for $j = 1, \ldots, n$. By 2 of Remark 2, K_j lives iff $\tilde{G}(i_l)$ lives for some $l = 1, 2, 3$. But $\tilde{G}(i_l)$ lives iff White plays in diagram $A(i_l)$ in A if $\tilde{G}(i_l) = G(i_l)$ and B if $\tilde{G}(i_l) = \overline{G}(i_l)$. Therefore, by the construction of $B(j)$ and 3, 4 of Remark 1, the only moves for White which make Λ alive are A or B in diagram $A(i)$ for $i = 1, \ldots, m$ and since Black plays well, he is forced to respond with C or D. Let us also note that the order of diagrams $A(i)$ in which White makes the moves is not important, since the result (after his m moves) is the same.

With the convention (1) White moves give values to the variables u_i for $i = 1, \ldots, m$. Now, because K_j lives for all $j = 1, \ldots, n$, it means that c_j is true for $j = 1, \ldots, n$, which implies that F is `TRUE`. This concludes the proof of our claim.

It remains to show that Λ is in G_{NP}. In the above discussion, we saw that if White wants to live with Λ he has to move A or B in diagram $A(i)$ for $i = 1, \ldots, m$ and we observed that the order of i's is not important. Moreover, trying to kill Λ, Black had unique choice-responses. Thus, a pruning function with the required properties exists. Note also that to check that Λ lives after White has played his first m moves takes only $O(n)$ steps (in every diagram $B(j)$ it is enough to check if one of the groups $\tilde{G}(i_l)$ for $l = 1, 2, 3$ has 2 liberties), and the theorem is proved.

References

1. Lichtenstein, D. and Sipser, M., GO is Polynomial-Space Hard, Journal ACM, Vol. **27**, No. 2, (April 1980) 393-401.
2. Morris, F.L., Playing Disjunctive Sums is Polynomial Space Complete, Int. Journal Game Theory, Vol. **10**, No.3-4, (1981) 195-205.
3. Robson, J., The Complexity of Go, Proc. IFIP (International Federation of Information Processing), (1983) 413-417.
4. Yedwab, L., On Playing Well in a Sum of Games, Master Thesis, MIT/LCS/TR-348, 1985.
5. Moews, D.J., On Some Combinatorial Games Connected with Go, Ph.D. Thesis, University of California at Berkeley, 1993.
6. Berlekamp, E., Wolfe, D. Mathematical Go: Chilling Gets the Last Point, A. K. Peters, 1994.