A Halfliar's Game

Ioana Dumitriu *
Joel Spencer †

March 6, 2002

Abstract

In Ulam's game Paul tries to find one of n possibilities with q Yes-No questions, while responder Carole is allowed to lie a fixed number k of times. We consider an asymmetric variant in which Carole must say yes when that is the correct answer (whence the *halflie*). We show that this variation allows Paul to distinguish between roughly 2^k as many possibilities as in Ulam's game.

1 Introduction

The basic liar game has two players whom we call Paul and Carole and three integer parameters (n, q, k). Paul is trying to find an unknown $x \in \{1, \ldots, n\}$ by asking q questions of Carole. The questions must all be of the form "Is $x \in A$?", where A is a subset of $\{1, \ldots, n\}$. Carole, the responder, is allowed to lie; however, she may lie at most k times. Paul wins if at the end of the q questions and responses the answer x is known with certainty.

Carole is allowed to play (and *will* play) an adversary strategy. That is, she does not preselect a particular x, but rather answers questions in a manner consistent with at least one possible x. At the end of the game, if there are at least two answers x, x' still valid (i.e., for which Carole has lied at most k times) then Carole has won; otherwise Paul is the winner of the game.

We further note that Paul's questions may (and generally *will*) be adaptive. That is, Paul's choice of question depends on Carole's previous answers.

In this formulation we have a two person perfect information game; thus we know that for any given triplet (n, q, k) either Paul or Carole has a perfect strategy. The question is, which one? Due to monotonicity, it suffices to answer the following more explicit question: given q and k, what is the maximal n (which we will denote by $A_k^*(q)$) for which Paul has a winning strategy?

Much work on the basic liar game was inspired by comments in the autobiography of Stanislas Ulam [6]. For this reason we, like many other authors, refer to the liar game as Ulam's game. Other early references include work by Alfred Rényi [4] and Elwyn

*Massachusetts Institute of Technology, Dept. of Math. E-mail: dumitriu@math.mit.edu
†Courant Institute of Mathematical Sciences (New York). E-mail: spencer@cims.nyu.edu

Berlekamp [1]. Pelc [3] solved the problem completely when $k = 1$. Spencer [5] solved the problem completely for any fixed k with q sufficiently large. In particular, it is known that for any fixed k

$$A_k^*(q) \sim \frac{2^q}{\binom{q}{k}}$$

where the asymptotics are as $q \to \infty$.

In this paper we modify Carole's ability to lie: she is still allowed to lie at most k times, but she is *only* allowed to lie when the truthful answer is "No". In other words, for Paul, any "No" he hears is a truthful answer and thus completely trustworthy; and any "Yes" answer he hears is a potential lie.

We call this the *halflie* game. We shall set $A_k(q)$ equal to the maximal n such that Paul has a winning strategy in the halflie game with parameters (n, q, k). Our main result is the following:

Theorem 1.1. *For any fixed $k \in \mathbb{N}$,*

$$A_k(q) \sim 2^k \frac{2^q}{\binom{q}{k}}$$

where the asymptotics are as $q \to \infty$.

Informally, the restriction of Carole to halflies, as opposed to full lies, allows Paul to probe 2^k times as many possibilities.

Many authors have commented on the connection between the now classic liar problem and the classic coding theory problem of sending n messages through a binary channel which may make up to k errors. The two problems are equivalent if Paul is required to pose all q queries at once – i.e., if his strategy must be nonadaptive. Alternatively, the liar problem is the coding theory problem with "feedback."

We may make a similar connection to the halfliar game. Consider what is sometimes called in the coding theory literature the `Z`-channel. In this channel, a one may be accidentally transformed into a zero, but a zero is never transformed into a one. We naturally identify zero with Yes and one with No. Our halfliar game may then be considered, roughly, the coding theory problem on the `Z`-channel with feedback.

Our result for $k = 1$ (Carole is allowed 1 lie) has been proven independently by F. Cicalese and D. Mundici [2]. Indeed, a number of the key ideas of their paper have proven to be very useful in our argument for general fixed k.

2 Two Perspectives

2.1 The Vector Game

There is a natural way to describe the state of the game in a middle position, after Paul has asked and Carole has answered a certain number of questions. For each (still) valid answer α there must be a certain number of lies that Carole has already used. If that number is greater than k then α is no longer a possibility (it is not *viable*). Suppose then that for a certain still viable α, the number of lies Carole has used is $k - i$ for some $0 \leq i \leq k$.

It follows that Carole still has the opportunity to lie i more times. Let us then say that possibility α is in state i.

We can thus describe the position of the game as a vector $(x_k, x_{k-1}, \ldots, x_0)$ where x_i is the number of possibilities in state i.

We further consider a query "Is $x \in A$?" by Paul to be described by a vector $(a_k, \ldots, a_0)$ where a_i is the number of $\alpha \in A$ of state i. Now consider the two possibilities.

- Carole says "no." Paul knows this is a truthful answer. Then none of the a_i possibilities of state i that were in A are viable, while the $x_i - a_i$ possibilities of state i that were not in A are still in state i. The new position is $(x_k - a_k, \ldots, x_0 - a_0)$.

- Carole says "yes." The a_i possibilities of state i that were in A remain in state i; further, as Carole may have been lying, the $x_i - a_i$ possibilities of state i that were not in A move to state $i-1$ (when $i = 0$ these possibilities are no longer viable). The new position is then $(a_k, a_{k-1} + x_k - a_k, \ldots, a_0 + x_1 - a_1)$.

We can describe the vector game without any reference to lying. There are q rounds. There is an initial vector $\vec{P} = (z_k, \ldots, z_0)$ with all z_j nonnegative integers. Each round Paul selects a vector $\vec{a} = (a_k, \ldots, a_0)$ with all a_j integers satisfying $0 \leq a_j \leq z_j$. Carole then resets $\vec{P}$ to either $(x_k - a_k, \ldots, x_0 - a_0)$ or $(a_k, a_{k-1} + x_k - a_k, \ldots, a_0 + x_1 - a_1)$. Paul wins if at the end of the game the sum of the coefficients of $\vec{P}$ is either zero or one. (Strictly speaking, a move by Carole that sets $\vec{P} = \vec{0}$ would be cheating. It is convenient, however, to allow this move and then declare Paul the winner.)

The halfliar game with parameters (n, q, k) is then equivalent to the q-round vector game with initial vector $\vec{P} = (n, 0, \ldots, 0)$ (of length $k+1$).

Remark 2.1. *The vector format may also be used in the full lie problem. The only distinction is that when $\vec{P} = (x_k, \ldots, x_0)$ and Paul selects $\vec{a} = (a_k, \ldots, a_0)$ then Carole may reset $\vec{P}$ to either $(a_k, a_{k-1} + x_k - a_k, \ldots, a_0 + x_1 - a_1)$ (as above) or $(x_k - a_k, x_{k-1} - a_{k-1} + a_k, \ldots, x_0 - a_0 + a_1)$.*

2.2 Packing k-Trees

In this subsection we define two concepts that proved to be crucial in our understanding of the problem, by providing a second format to the Halfliar Game.

First, we introduce the (perhaps familiar) δ function, which we use to define the two concepts mentioned above: those of k-tree and k-set.

Definition 1. *Given two points in $\{Y, N\}^q$, $w = w_1 w_2 \ldots w_q$ and $w' = w'_1 w'_2 \ldots w'_q$, we define $\delta(w, w')$ to be the smallest i for which $w_i \neq w'_i$.*

Definition 2. *A k-tree is a rooted tree of depth at most k whose vertices are points of $\{Y, N\}^q$ with the following properties:*

1. *Let the root be $r = r_1 r_2 \ldots r_q$. For each $1 \leq i \leq q$ such that $r_i = N$, there exists exactly one child r' of r with $\delta(r, r') = i$. Moreover, these are all the children of r;*

2. *Let r' be a nonroot point, with parent r^*, and of depth less than k. For each $i > \delta(r', r^*)$ for which $r'_i = N$, there exists exactly one child $\tilde{r}$ of r' such that $\delta(\tilde{r}, r') = i$. Moreover, these are all the children of r'.*

Definition 3. *We call the set of nodes of a k-tree a k-set, and we call the sequence at the root of the tree the* stem.

To better illustrate the definitions above, we have inserted Figure 1.

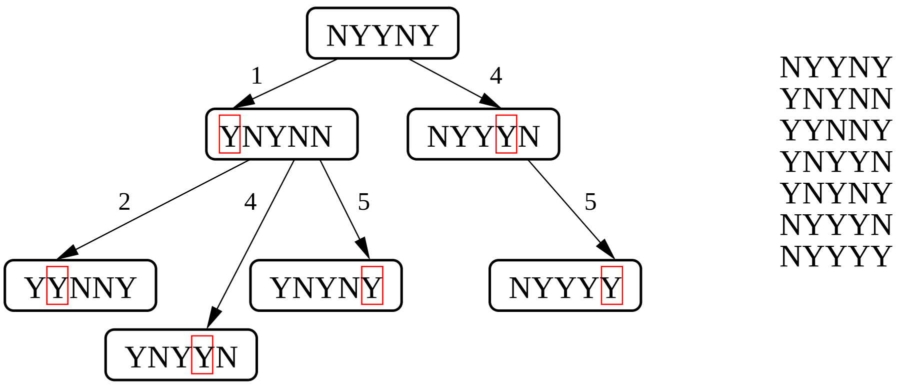

Figure 1: A 2-tree (left) and the corresponding 2-set (list, right) when $q = 5$; the boxed letters and the numbers on the arrows the represent δ(parent, child) and the place where N has turned into Y in the child sequence

Remark 2.2. *Note that any point in $\{Y, N\}^q$ is a 0-set; critically, the set of paths leading to a given value α in the decision tree for the Halflie game with k lies form an k-set (and this would be an equivalent alternative definition of a k-set).*

Claim 2.2.1. *Let A be a k-set in $\{Y, N\}^q$ with stem v.*

(i). Suppose v starts with an Y. Then $\{w : Yw \in A\}$ is a k-set (in $\{Y, N\}^{q-1}$) and $\{w : Nw \in A\} = \emptyset$.

(ii). Suppose v begins with an N. Then $\{w : Yw \in A\}$ is a $(k-1)$-set (in $\{Y, N\}^{q-1}$) and $\{w : Nw \in A\}$ is a k-set.

Proof. The proof is immediate from the definition. □

The k-sets prove to be essential for proving both upper and lower asymptotic bounds for $A_k(q)$; we will state and prove here two results that will be used in the next sections.

The following lemma follows easily from the definition.

Lemma 2.3. *For any k and q, the maximum size of a k-set in $\{Y, N\}^q$ is at most*

$$m_k(q) = 1 + \binom{q}{1} + \ldots + \binom{q}{k} .$$

Proof. We will show that the number of elements on level i of the k-tree is bounded from above by $\binom{q}{i}$.

Let w be a node at level i, and let $r = x_0, x_1, \ldots, x_i = w$ be the path from the root r to w in the tree. Let $p_1 = \delta(x_0, x_1)$, $p_2 = \delta(x_1, x_2)$, ..., $p_i = \delta(x_{i-1}, x_i)$. Note that $1 \le p_1 < p_2 < \ldots < p_i \le q$.

By the definition of the k-tree, the choice of the positions $p_1, \ldots, p_i$ determines the element w completely. Since there are at most $\binom{q}{i}$ possibilities for the choice of the p_is, it follows that there are at most $\binom{q}{i}$ elements on the ith level of the tree. □

2.3 The Synthesis

The final result that we state and prove in this section is the following crucial equivalence theorem, which connects the two formats of the Halflie game (vectorial and k-set) that we have presented here.

Theorem 2.4. *Given a number q of questions, the position $(x_k, x_{k-1}, \ldots, x_1, x_0)$ is a winning position in q moves for Paul if and only if one can find a family $\mathcal{P}$ of disjoint sets $P_{i,j}$, $0 \le i \le k$, $1 \le j \le x_i$, such that*

1. *for all i and j, $P_{i,j}$ is an i-set;*
2. *for all i and j, the elements of $P_{i,j}$ are in $\{Y, N\}^q$.*

If such a family of disjoint sets exists, we will say that we can pack *(simultaneously) x_i i-sets, $i = 0, \ldots, k$, in the full binary tree of size 2^q.*

For an example of what it means to "pack" a k-set, refer to Figure 2 below.

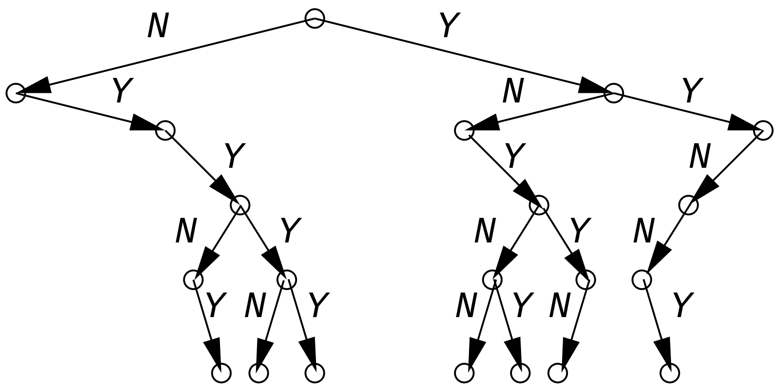

Figure 2: Packing the 2-set of Figure 1 into the Decision Tree; the rest of the Decision Tree vertices are not drawn for convenience

Proof. The "left to right" direction of the argument is easy; indeed, if Paul can win the game from $(x_k, x_{k-1}, \ldots, x_1, x_0)$ in q moves, let us examine the decision tree for the Halflie game, starting at this position. If we look at the set $P_{i,j}$ of possible paths that lead to a given answer (call it $\alpha_{i,j}$) from among the x_i possible answers in the case when Carol has already lied $k-i$ times, we can easily see that $P_{i,j}$ is an i-set. Since all such sets, for all the i's, are disjoint, it follows that they represent a packing of x_i i-sets, $i = 0, \ldots, k$, in the full binary tree of size 2^q.

For the other direction, we will show how to construct the splitting questions which lie at the nodes of the Decision Tree. We can pack the i-sets, $0 \leq i \leq k$, and we choose to label each set of leaves that correspond to an i-set by a number from 1 through $n := x_k + \ldots + x_0$. Also, we assume that the packing is done so that the "no" branches to the left, whereas the "yes" branches to the right, just like in Figure 2.

For each node z of the Decision Tree we define

$$\begin{aligned} L(z) &= \{r : \text{some leaf in the left subtree of } z \text{ has label } r\} \\ Q(z) &= \{1, 2, \ldots, n\} \setminus L(z) \ . \end{aligned}$$

At node z, we place the question "Is the answer in $Q(z)$?".

Claim 2.4.1. *The questions we indicated above represent a strategy for Paul.*

Proof. We proceed by induction on the number of questions q. The case $q = 0$ corresponds to $n = 1$ and $k = 0$, in other words Paul knows what the answer is and no questions are needed.

Assume we have proved that packing implies winning for any number of questions smaller than q, and let us examine the question at the root of the q-Decision Tree. Paul asks the question "Is the answer in $Q(\text{root})$?" and

- If Carole's answer is "no", Paul knows that she is telling the truth, and it follows that the answer *must* be one of the labels present at the leaves of the left subtree of the root. Since she answered "no", that is where he will search.

  Let
  $$A_{no}(i,j) = \{w : Nw \in P(i,j) \text{ and } P(i,j)\text{'s stem starts with a N}\};$$
  any one of these paths lead to a valid answer, and any valid answer has such a path leading to it.

  Using Claim 2.2.1, we get that $A_{no}(i,j)$ is an i-set. Since these i-sets are packed in the left subtree, they are packed in the Decision Tree corresponding to the $q-1$ question game. By induction, Paul can use the same strategy to win.

- If Carole's answer is "yes", Paul cannot, generally, determine the truthfulness of her answer.

  Let
  $$\begin{aligned} A_{yes}(i,j) &= \{w : Yw \in P(i,j) \text{ and } P(i,j)\text{'s stem starts with a Y}\} \\ B_{yes}(i,j) &= \{w : Yw \in P(i,j) \text{ and } P(i,j)\text{'s stem starts with a N}\} \ , \end{aligned}$$
  and again each such path leads to a valid answer and any valid answer has such a path leading to it.

Using Claim 2.2.1, we get that all $A_{yes}(i,j)$ are i-sets, and all $B_{yes}(i,j)$ are (i–1)-sets. Moreover, both $A_{yes}(i,j)$s and $B_{yes}(i,j)$s are packed in the right subtree of the root. But since Carole has answered "yes", Paul *will* be searching in the right subtree of the root! And since by induction, because these i- and $(i-1)$-sets are packed in the right subtree (and thus in the $q-1$ question Decision Tree), it follows that Paul can use the strategy to win.

□

The second implication is proved, and so is the lemma. □

3 Lower bounds

We will start with a definition. Here we are using the vector format previously described.

Definition 4. *We say that from position $(x_0, x_1, \ldots, x_k)$ we can make a perfect split if there exists an allowable question $(a_0, a_1, \ldots, a_k)$ that Paul can ask, such that the outcome in the case of an affirmative answer is the same as for a negative one.*

If they exist, the integers $a_0, \ldots, a_k$ are (uniquely) defined by the following set of equations:

$$\begin{aligned} a_0 &= \frac{x_0}{2} , \\ a_i &= \frac{x_i - x_{i-1} + a_{i-1}}{2} , \quad \forall \ 1 \le i \le k . \end{aligned}$$

Remark 3.1. *There are two ways in which a perfect split can fail to exist: the first one is due to issues of parity, and the second one is due to the fact that the question might not be allowable (for a question to be allowable we must have $0 \le a_i \le x_i$ for every $0 \le i \le k$).*

For example, from $(2,1)$ we can make a perfect split by asking the question $(1,0)$, which regardless of the answer takes Paul to position $(1,1)$.

Intuitively, whenever one is in need of perfect splits in the Halflie game, the best initial values to start from are powers of 2 (or "close" to a power of 2, like a "small" multiple of a power of 2). Unfortunately, not every number is of such a form, which is an inconvenience. However, in this game, there is a concept of dominance which will make things easier, and will allow us to reduce the study of the problem to numbers $n = a2^s$ where a will be "small" compared to s.

Definition 5. *We will say that position $p = (x_0, x_1, \ldots, x_k)$ dominates position $p' = (x'_0, x'_1, \ldots, x'_k)$ if for all $i \le k$, $\sum_{j=0}^{i} x_i \ge \sum_{j=0}^{i} x'_i$. For example, $(2,3,0,0,1)$ dominates $(2,1,1,1,1)$.*

Remark 3.2. *The essential fact about a position p dominating another position p' is that, trivially, if p is a winning position for Paul in q moves, then so is p'. In particular, if $n \le a2^s$ for some a and s, and if $(a2^s, 0, \ldots, 0)$ is a winning position for Paul when q questions are left, then so is $(n, 0, \ldots, 0)$.*

In what follows we show that for any given k, $\alpha < 2^k$, and q large enough, there exists a "small" multiple of a power of 2 between $\alpha 2^q/\binom{q}{k}$ and $2^k 2^q/\binom{q}{k}$.

Lemma 3.3. *For any $k \geq 1$ and α such that $\alpha < 2^k$, there exist integers T and q_0 such that for all $q \geq q_0$, there exists at least one $a \in (2^T, 2^{T+1}] \cap \mathbb{N}$ such that*

$$\lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil \leq a 2^s < \lfloor \frac{2^k + \alpha}{2} \frac{2^q}{\binom{q}{k}} \rfloor$$

for some integer s.

Proof. Given k and α as in the statement of the lemma, a simple calculation shows that there is a value q_0 such that for all $q \geq q_0$,

$$\frac{\lfloor \frac{2^k+\alpha}{2} \frac{2^q}{\binom{q}{k}} \rfloor - \lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil}{\lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil} > \frac{2^k - \alpha}{4\alpha} \ .$$

Thus, choose $q \geq q_0$, and T the smallest integer such that

$$\frac{1}{2^T} \leq \frac{2^k - \alpha}{4\alpha} \ ,$$

and consider all numbers of the form $a2^s$, with $a \in (2^T, 2^{T+1}] \cap \mathbb{N}$ and s an integer. Let $n_1 = (a_1 - 1)2^{s_1}$ be the largest such number smaller than or equal to $\lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil$. We will show that $a_1 2^{s_1}$ is strictly between $\lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil$ and $\lfloor \frac{2+\alpha}{2} \frac{2^q}{\binom{q}{k}} \rfloor$.

To begin with, it is clear by choice of a_1 and s_1 that

$$\lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil < a_1 2^{s_1} \ .$$

The other inequality is almost as easy to prove. Indeed,

$$\begin{aligned} a_1 2^{s_1} &= \left(\frac{1}{a_1 - 1} + 1 \right) (a_1 - 1) 2^{s_1} \leq \\ &\leq \left(\frac{1}{2^T} + 1 \right) \lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil \\ &\leq \left(\frac{2^k - \alpha}{4\alpha} + 1 \right) \lceil \alpha \frac{2^q}{\binom{q}{k}} \rceil \\ &< \lfloor \frac{2^k + \alpha}{2} \frac{2^q}{\binom{q}{k}} \rfloor \ , \end{aligned}$$

by way of choosing T and q_0.

∎

Remark 3.4. *Note that if α is considerably smaller than 2^k, one does not have to go too far to find a suitable T; it is when α is close to 2^k that T gets large. However, since α is always fixed, so is T.*

The following result is a crucial splitting lemma.

Lemma 3.5. *Let $a \in \mathbb{N}$. Let $1 = m_0, m_1, \ldots, m_k$ such that $m_i \geq \sum_{j=0}^{i-1} m_j$, for all $1 \leq i \leq k$. Set $j_0 = n_0 = 1$. For $1 \leq i \leq k$ set*

$$\begin{aligned} j_i &= m_i - m_{i-1} - m_{i-2} - \ldots - m_0 \qquad \textit{and} \\ n_i = 2m_i - j_i &= m_i + m_{i-1} + m_{i-2} + \ldots + m_0 \end{aligned}$$

Let $t > k$. Then from position $(am_0 2^t, am_1 2^{t-1}, \ldots, am_k 2^{t-k})$ we can make a perfect split by asking question $(aj_0 2^{t-1}, aj_1 2^{t-2}, \ldots, aj_k 2^{t-k-1})$, with resulting position $(an_0 2^{t-1}, an_1 2^{t-2}, \ldots, an_k 2^{t-k-1})$. Furthermore, $n_i \geq \sum_{j=0}^{i-1} n_j$, for all $1 \leq i \leq k$.

Proof. Note that the requirement that $m_i \geq \sum_{j=0}^{i-1} m_j$, for all $1 \leq i \leq k$, insures that the question Paul asks for the split is an allowable one (basically, it insures that $0 \leq j_i \leq m_i$ for all i).

The formulas for j_i and n_i are easily established by inspection. Let us now examine what it means for the n_is to inherit the "growth property" of the m_is. We must show that

$$n_i \geq n_{i-1} + n_{i-2} + \ldots + n_0 ,$$

and since

$$n_i = m_i + m_{i-1} + \ldots + m_0 ,$$

it suffices to show that $m_{j+1} \geq n_j$ for all $0 \leq j \leq i-1$. Since $n_j = m_j + \ldots + m_0$, the latter inequality is just the condition we imposed on the m_is, and we are done. ☐

We now need to establish a technical result.

We define a (uniquely determined) infinite sequence of polynomials $p_0, p_1, \ldots, p_j, \ldots,$ by the recursion

$$\begin{aligned} p_0(s) &= 1 , \ \forall\, s & (1) \\ p_j(s+1) &= p_j(s) + p_{j-1}(s) + \ldots + p_0(s) , \ \forall j \geq 1 , & (2) \\ p_j(0) &= 2^{j-1} , \ \forall j \geq 1 . & (3) \end{aligned}$$

Lemma 3.6. *For every $j \geq 1$ and every $s \geq 0$*

$$p_j(s) \geq p_{j-1}(s) .$$

Furthermore, $p_j(s) = \binom{s}{j} + q_j(s)$ for some polynomial q_j with $deg(q_j) \leq j-1$.

Proof. Since the initial values (at $s = 0$) of the polynomials satisfy the same kind of recurrence as the m_is of Lemma 3.5, it follows by the argument used there that $p_j(s) \geq p_{j-1}(s)$ for all integer s and $j \geq 1$.

The second part of this technical lemma can be proved inductively. For $j = 1$, we see from the recurrence that $p_1(s) = s + 1 = \binom{s}{1} + 1$ for all s.

Assume now that we have proved the result for all $i \leq j - 1$ and let us prove it for j.

First, note that the polynomial recurrence can be replaced by the simpler 2-term recurrence

$$p_j(s+1) = p_j(s) + p_{j-1}(s+1) \ .$$

Hence, by going backwards, we obtain that

$$p_j(s+1) = \sum_{k=1}^{s+1} p_{j-1}(k) + p_j(0) = 2^{j-1} + \sum_{k=1}^{s+1} p_{j-1}(k) \ .$$

By induction, $p_{j-1}(k) = \binom{k}{j-1} + q_{j-1}(k)$, where q_{j-1} is a polynomial of degree at most $j - 2$. Thus,

$$p_j(s+1) = 2^{j-1} + \sum_{k=1}^{s+1} p_{j-1}(k) = 2^{j-1} + \sum_{k=1}^{s+1} \left(\binom{k}{j-1} + q_{j-1}(k) \right) \ .$$

By the additive property of binomial coefficients,

$$\begin{aligned} p_j(s+1) &= \binom{s+2}{j} + \sum_{k=1}^{s+1} q_{j-1}(k) + 2^{j-1} \\ &= \binom{s+1}{j} + \left(2^{j-1} + \sum_{k=1}^{s+1} q_{j-1}(k) + \binom{s+2}{j} - \binom{s+1}{j} \right) \ . \end{aligned}$$

Since the degree of q_{j-1} is at most $j - 2$, it follows that $\sum_{k=1}^{s+1} q_{j-1}(k)$ is a polynomial of degree at most $j - 1$. Thus the polynomial

$$q_j(s+1) = 2^{j-1} + \sum_{k=1}^{s+1} q_{j-1}(k) + \binom{s+2}{j} - \binom{s+1}{j}$$

has degree at most $j - 1$ (since $\binom{s+2}{j} - \binom{s+1}{j}$ has degree $j - 1$), and the lemma is proved by induction. $\square$

Remark 3.7. *In fact, one can prove that*

$$p_j(s) = \binom{s+j}{j} + \sum_{i=2}^{j} 2^{j-2} \binom{s+j-i}{j-i} \ ,$$

for all $j, s \geq 0$.

Lemma 3.8. *For any k, a and $s \geq k$ integers, from starting position $(a2^s, a2^{s-1}, a2^{s-1}, \ldots, a2^{s-1})$, Paul can make $s-k$ perfect splits. Furthermore, after the jth split, the resulting position is $(a2^{s-j}, ap_1(j)2^{s-1-j}, \ldots, ap_k(j)2^{s-k-j})$.*

Proof. The proof follows directly from Lemmas 3.5 and 3.6. □

Lemma 3.9. *If $\sum_{i=1}^{k} x_i(1+q+\ldots+\binom{q}{i}) = 2^q - S$ with $S \geq 0$, and if Paul wins the q question game from $(x_k, \ldots, x_1, 0)$ then he wins from $(x_k, \ldots, x_1, S)$.*

Proof. By Theorem 2.4, since Paul wins we can simultaneously pack all x_i of the i-sets, for all $1 \leq i \leq k$. Because of their size limitation (see Lemma 2.3), this packing leaves at least S points of $\{Y,N\}^q$ uncovered – but each point can be a 0-set, and hence we can add another S 0-sets to the packing. Thus Paul wins from $(x_k, \ldots, x_1, S)$. □

The last lemma we need is the following.

Let now k, $\alpha < 2^k$, and T be fixed, satisfying Lemma 3.3. Let $p_0, \ldots, p_k$ the polynomials of Lemma 3.6.

Lemma 3.10. *There exists q_1 such that for all $q \geq q_1$, the following holds. Let a, s be integers such that $2^t < a \leq 2^{T+1}$ and $a2^s \leq \alpha \frac{2^q}{\binom{q}{k}}$. Set $r = q - s + k$. Then Paul can win the r-question game from position $(a2^k, ap_1(s-k)2^{k-1}, \ldots, ap_k(s-k))$.*

Proof. What we must do here is to show that it is possible to simultaneously pack $a2^k$ k-sets, $a2_{k-1}p_1(s-k)$ $(k-1)$-sets, …, and $ap_k(s-k)$ 0-sets in the full binary 2^r tree.

Since $a2^s \leq \alpha \frac{2^q}{\binom{q}{k}}$, $s \leq q - k\log_2 q + O(1)$, and hence $r = q - s + k \to \infty$ as $q \to \infty$.

We proceed by induction over k. For $k = 0$ (the no-lie case), we know that if $a2^s \leq a2^q$, then we can win the game starting at (a) with r questions, as $a \leq 2^{q-s} = 2^r$.

Assume now that we have proved the statement for all numbers smaller than $k-1$, and we will now show it for k.

Now since $(a2^k, \sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}, 0, \ldots, 0, ap_k(s-k))$ dominates $(a2^k, ap_1(s-k)2^{k-1}, \ldots, ap_k(s-k))$, it suffices to pack $a2^k$ k-sets, $\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}$ $(k-1)$-sets, and $ap_k(s-k)$ 0-sets in the full binary 2^r tree.

We will choose the $a2^k$ k-sets as follows:

$$\begin{aligned} S_1 &= \{e_1, e_2, \ldots, e_{k+1}\} \\ S_2 &= \{e_{k+2}, e_{k+3}, \ldots, e_{2k+2}\} \\ &\ldots \\ S_{a2^k} &= \{e_{(a2^k-1)(k+1)+1}, e_{(a2^k-1)(k+1)+2}, \ldots, e_{a2^k(k+1)}\}, \end{aligned}$$

where e_r is the sequence of all Ns except for the rth location which contains a Y (for example, $e_1 = YNNNN\ldots$). Here we assume that q is much larger than $a2^k(k+1)$.

It is immediate to check that these are indeed k-sets and they are disjoint.

To insure that the $(k-1)$-sets that we construct are disjoint from these, we will require that *any* point in $\{Y,N\}^q$ that goes into any one of the $(k-1)$-sets starts with $a2^k(k+1)$

Ns (which means that in effect we will be packing the $(k-1)$-sets into the full binary tree of size $2^{r-a2^k(k+1)}$).

Claim 3.10.1. *Under all the assumptions above, for q large enough, we can pack $\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}$ $(k-1)$-sets in the full binary tree of size $2^{r-a2^k(k+1)}$.*

Proof. From the fact that $a2^s \le \alpha \frac{2^q}{\binom{q}{k}}$ one easily obtains that

$$a\binom{s-k}{k} < \frac{\alpha}{2^k}2^r \ , \tag{4}$$

and hence, given some small δ such that $\alpha' = \alpha + \delta < 2^k$, for q large enough, we shall have that $ap_k(s-k) < \frac{\alpha'}{2^k}2^r$.

From (4) we also obtain that as q gets large,

$$a\binom{s-k}{k-1} = o(2^{r(\frac{k-1}{k}+\epsilon)}) \ , \tag{5}$$

where $\epsilon < \frac{1}{k}$ is an arbitrary (but fixed) small number.

Now since

$$\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i} = 2a\binom{s-k}{k-1}(1+o(1))$$

in asymptotic notation (because the polynomials, with the exception of p_{k-1}, are all of degree smaller than $k-1$, and the leading term in $p_{k-1}(t)$ is $\binom{t}{k-1}$), (5) implies that

$$\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i} = o(2^{r(\frac{k-1}{k}+\epsilon)}) \ .$$

Since k is fixed, but r is allowed to be very large, it follows that

$$\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i} = o(2^{r(\frac{k-1}{k}+\epsilon)}) = o\left(\frac{2^{r-a2^k(k+1)}}{\binom{r}{k-1}}\right) \ . \tag{6}$$

Consider now the k–1 lie problem; fix $\tilde{\alpha} < 2^{k-1}$, for example $\tilde{\alpha} = \frac{1}{2}$. By induction, Paul wins the s-question game from starting position $(\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}, 0, \ldots, 0)$, provided that s is large enough to have

$$\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i} < \frac{1}{2}\frac{2^s}{\binom{s}{k-1}} \ .$$

Now choose $s = r - a2^k(k+1)$. From (6), for r large enough,

$$\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i} = o\left(\frac{2^{r-a2^k(k+1)}}{\binom{r}{k-1}}\right) \le \frac{1}{2}\frac{2^s}{\binom{s}{k-1}} \ .$$

But this says that Paul wins the $r-a2^k(k+1)$ question game from position $(\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}, 0, \ldots, 0)$, provided that r is large enough.

Since as $q \to \infty$, $r \to \infty$, this holds when q is large enough. $\square$

Thus, we can pack the $a2^k$ k-sets and the $\sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}$ $(k-1)$-sets in a disjoint fashion in the full binary tree on 2^r vertices. But how much space do we have left? The k-sets take up $O(1)$ space, the $(k-1)$-sets take up at most $1 + \binom{s-k}{1} + \ldots + \binom{s-k}{k-1}$ space each, by Lemma 2.3; hence by (6), the total space taken is at most

$$O(1) + \sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}\left(1 + \binom{s-k}{1} + \ldots + \binom{s-k}{k-1}\right) = o(2^r) \ .$$

Since the singletons, all $ap_k(s-k)$ of them, take up at most a constant fraction $\alpha'/2^k$ of the total size of the tree, it follows that, by use of Lemma 3.9, Paul wins the r question game starting at $(a2^k, \sum_{i=1}^{k-1} ap_i(s-k)2^{k-i}, 0, \ldots, 0, ap_k(s-k))$, and because this position dominates $(a2^k, ap_1(s-k)2^{k-1}, \ldots, ap_k(s-k))$, he also wins with r questions from the latter position. $\square$

Remark 3.11. *An alternate proof of Lemma 3.10 may be given by proving that Paul wins the (harder) full lie r-question game from position $(a2^k, ap_1(s-k)2^{k-1}, \ldots, ap_k(s-k))$. It is shown in [5] that for all k there exists c such that for r sufficiently large Paul wins the r-question full lie game from position $(x_k, \ldots, x_0)$ whenever*

$$\sum_{i=0}^{k} x_i \left(\sum_{j=0}^{i} \binom{r}{i}\right) < 2^r - cr^k$$

A calculation shows that for r sufficiently large the initial position of Lemma 3.10 satisfies this condition.

All that is left is now to put together the results of this section.

Theorem 3.12. *For all k and $\alpha < 2^k$, there exists a q large enough so that for any $n \le \alpha \frac{2^q}{\binom{q}{k}}$, Paul can win the q-question game from position $(n, 0, 0, \ldots, 0)$.*

Proof. Using the results of Lemma 3.3, it is enough to prove the theorem for $n = a2^s$ with $a \in (2^T, 2^{T+1}] \cap \mathbb{N}$. But since $(a2^s, 0 \ldots, 0)$ is dominated by $(a2^s, a2^{s-1}, \ldots, a2^{s-1})$, it is sufficient to show that Paul can win the q-question game starting from the latter position.

From $(a2^s, a2^{s-1}, \ldots, a2^{s-1})$, Paul first makes the $s-k$ perfect splits of Lemma 3.8, resulting in position $(a2^k, ap_1(s-k)2^{k-1}, \ldots, ap_k(s-k))$. By Lemma 3.10 he then wins with $r = q - s + k$ further questions. $\square$

4 Upper Bounds

We will start with a double definition. Let x be a parameter later to be optimized.

Definition 6. *A k-set A is* normal *if all sequences $w \in A$ have at least $\frac{q}{2}-x$ Ns. Otherwise the set is* abnormal .

This definition allows for two easy lemmas.

Lemma 4.1. *The minimum size of a k-set is bounded from below by*

$$\sum_{i=1}^{k} \binom{\frac{q}{2}-x+i-1}{i} .$$

Proof. We look at the number of points on the ith level of the k-tree corresponding to the k-set. These are paths for which exactly i lies have been committed by Carole.

Let the path from the root of the k-tree down to node $w \in \{Y,N\}^q$ on level i be $r_0 = root, r_1, \ldots, r_i$, and let $\delta(r_j, r_{j+1})$, for all $0 \le j \le i-1$, be the place where r_{j+1} first differs from its parent.

Let n_{j+1} = number of Ns in r_j before position $\delta(r_j, r_{j+1})$, for all $0 \le j \le i-1$. Then it follows that to each sequence

$$1 \le n_1 \le n_2 \le \ldots \le n_i ,$$

corresponds exactly one point on level i. Since we have at least $\frac{q}{2}-x$ choices for n_i (but we could in fact have much more), it follows that level i must contain at least $\binom{\frac{q}{2}-x+i-1}{i}$ different points.

By adding all these lower bounds for levels 0 through i we get the result of the lemma. □

Lemma 4.2. *The total number of abnormal sets we can pack in the Decision Tree is at most*

$$\sum_{i=0}^{\frac{q}{2}-x} \binom{q}{i} .$$

Proof. Since they must be disjoint (because of the packing), it follows that the total number of abnormal sequences cannot surpass the total number of sequences with less than $\frac{q}{2}-x$ Ns. The result follows. □

The two lemmas above allow us to give the following upper bound.

Lemma 4.3. *If Paul can win the k-lie game with q questions starting from position $(n, 0, \ldots, 0)$, then*

$$n \le \frac{2^q}{\sum_{i=1}^{k} \binom{\frac{q}{2}-x+i-1}{i}} + \sum_{i=0}^{\frac{q}{2}-x} \binom{q}{i} ,$$

for any x.

Proof. If he can win, the Paul can somehow pack n k-sets, normal and abnormal. The bound follows then immediately from Lemmas 4.1 and 4.2. $\square$

With this, we are now ready to present the main result of the section. Let $\epsilon > 0$, and k fixed.

Theorem 4.4. *There exists q_0 sufficiently large such that for all $q \geq q_0$, for any n such that Paul wins the k-lie, q question game from position $(n, 0, \ldots, 0)$,*

$$n \leq (2^k + \epsilon) \frac{2^q}{\binom{q}{k}} .$$

Proof. Let $x = c_k \sqrt{q \ln q}$, with $c_k > \sqrt{\frac{k}{2}}$.

To win the game, Paul must be able to pack n k-sets in the Decision Tree. By Lemma 4.3 we get that

$$n \leq \frac{2^q}{\sum_{i=1}^{k} \binom{\frac{q}{2}-x+i-1}{i}} + \sum_{i=0}^{\frac{q}{2}-x} \binom{q}{i} .$$

Since

$$\sum_{i=0}^{\frac{q}{2}-x} \binom{q}{i} = 2^q \ \Pr[\ S \text{ contains less than } (\frac{q}{2} - c_k \sqrt{q \ln q}) \ Ns\ ]$$

where S is a random sequence of q Ys and Ns, Chernoff-type bounds yield that

$$\Pr[\ S \text{ contains less than } (\frac{q}{2} - c_k \sqrt{q \ln q}) \ Ns\ ] \leq e^{-c_k^2 q \ln q/(q/2)} = q^{-2c_k^2} = o\left(\binom{q}{k}^{-1} \right) ,$$

by our choice of c_k.

Thus there exists a q_0 large enough such that

$$\sum_{i=0}^{\frac{q}{2}-x} \binom{q}{i} < 2^{k-1} \epsilon \frac{2^q}{\binom{q}{k}} . \tag{7}$$

Since $\sum_{i=1}^{k} \binom{\frac{q}{2}-x+i-1}{i}$ is a polynomial in q of degree k, and since $x = o(q)$, it follows that

$$\sum_{i=1}^{k} \binom{\frac{q}{2} - x + i - 1}{i} \sim \binom{\frac{q}{2}}{k} \sim 2^{-k} \binom{q}{k} .$$

Hence there must be a q large enough so as to have

$$\sum_{i=1}^{k} \binom{\frac{q}{2} - x + i - 1}{i} \geq \frac{1}{2^k + \frac{\epsilon}{2}} \binom{q}{k} . \tag{8}$$

From equations (7) and (8), it follows that for any $q \geq q_0$,

$$n \leq (2^k + \epsilon) \frac{2^q}{\binom{q}{k}} ,$$

and the theorem is proved. $\square$

References

[1] E.R. Berlekamp, Block coding for the binary symmetric channel with noiseless, delayless feedback, In *Error-correcting codes*, H.B. Mann (ed.), Wiley (1968), 61-88

[2] F. Cicalese and D. Mundici, Optimal coding with one asymmetric error: below the Sphere Packing bound, In *Proceedings of 6th Annual International Conference on Computing and Combinatorics*–COCOON'2000, *Lecture Notes in Computer Science*, **1858**, pp. 159–169, Springer–Verlag, 2000.

[3] A. Pelc, Solution to Ulam's problem on searching with a lie, *J. Combinatorial Theory*, Ser. A. **44** (1987), 129-142

[4] A. Rényi, *A Diary on Information Theory*, J. Wiley and Sons (1984), (original publication: *Napló az információelméletről*, Gondolat, Budapest, 1976).

[5] J. Spencer, Ulam's searching problem with a fixed number of lies, *Theoretical Computer Science* **95** (1992), 307-321

[6] S. M. Ulam, *Adventures of a Mathematician*, Scribners, 1976.