# The Liar Game over an Arbitrary Channel

Ioana Dumitriu *

Joel Spencer †

November 15, 2002

**Abstract**

We introduce and analyze a liar game in which $t$-ary questions are asked and the responder may lie at most $k$ times. As an additional constraint, there is an arbitrary but prescribed list (the channel) of permissible types of lies. For any fixed $t$, $k$, and channel, we determine the exact asymptotics of the solution when the number of queries goes to infinity.

## 1 Introduction

This paper defines and analyzes a generalization of the well-known Rényi-Ulam liargame.

In the original Rényi-Ulam 2-player game, player 1 (whom we shall call Carole) thinks of an $x \in \{1, \ldots, n\}$ and player 2 (whom we shall call Paul) must find it by asking $q$ Yes/No questions. There is, of course, a catch (which gives the game its name): Carole is allowed to lie. However, she may only lie at most $k$ times, where $k$ is a fixed integer. The question posed by Rényi and Ulam is "for which $n, k, q$ can Paul win?"

The original game, together with many references and variants, can be found in the excellent survey article by Pelc [5]; for historical references, we recommend Rényi [6], Ulam [9] and Berlekamp [2].

*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

It is known that Carole wins (employing an adversary strategy) when

$$2^q < n \left[ 1 + \binom{q}{1} + \ldots + \binom{q}{k} \right]$$

and that for $k$ fixed and $q$ sufficiently large the converse, speaking roughly, almost holds (see [7]). In particular, with $k$ fixed the largest $n$ for which Paul wins is asymptotically (in $q$) $2^q / \binom{q}{k}$.

The (recently introduced) halflie game adds a restriction to Carole: If the honest response is Yes then Carole must say Yes.

Let $A_{Z,k}(q)$ be the maximal $n$ for which Paul can win the halflie game with $q$ questions and $k$ (fixed) lies. F. Cicalese and D. Mundici [3] showed that

$$A_{Z,1}(q) \sim \frac{2^{q+1}}{q}$$

and the current authors [4] showed that for any fixed $k$

$$A_{Z,k}(q) \sim \frac{2^{q+k}}{\binom{q}{k}}$$

In the above, we use the letter $Z$ to make the connection between the halflie game and the $Z$-channel of Coding Theory (see Figure 1).

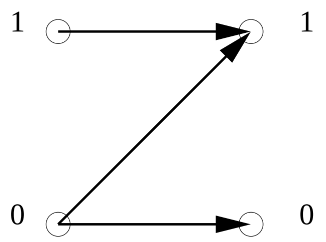


Figure 1: The $Z$-channel

Bits are sent through the channel. A 1 sent will always be received correctly but a 0 sent may be received as a 1. In medical jargon, we allow for false positives but not for false negatives.

As opposed to the original liargame where the results are extremely precise, in the halflie game we have to settle for asymptotic analysis.

In this paper, we extend our results to arbitrary channels $C$.

**Definition 1.** *A $t$-ary channel $C$ is a set of ordered pairs $(x, y)$, $1 \leq x, y \leq t$ such that for each $1 \leq x \leq t$, $(x, x) \in C$.*

In Coding Theory language, channels are used to send messages. One of the messages $1, \ldots, t$ (of course, they could be labelled differently and for the binary $t = 2$ case are generally $0, 1$ or Yes/No) can be sent through the channel. When $x$ is sent $y$ can be received only if $(x, y) \in C$. When $y \neq x$ the channel has made an error.

**Definition 2.** *We define for $1 \leq j \leq t$, $L(j)$ to be the (possibly empty) set of $x \neq j$ such that $(x, j) \in C$. The pairs $(x, y) \in C$ with $x \neq y$ are called potential errors. $E = E(C)$ denotes the number of potential errors in the channel $C$.*

For the benefit of the reader, we have included Figure 2.

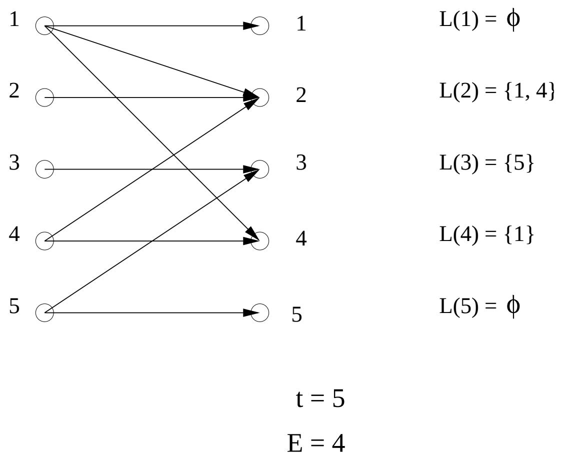


Figure 2: A 5-ary channel $C$, with $E = 4$

We now define the focus of our work, the $(n, k, C)$-liargame with $q$ questions. There are two players, Paul and Carole, and $q$ rounds. There is a set $\Omega$ of size $n$, the possibilities. Carole thinks of an $\alpha \in \Omega$. On each round Paul partitions $\Omega$ into disjoint sets $A_1, \ldots, A_t$; Carole finds that $i$ for which $\alpha \in A_i$ and responds with either $i$ or some $j \neq i$ with $(i, j) \in C$. The latter case is called a lie and it is identified with an error in the channel.

Paul's choice of partitions in later rounds can, and in general will, depend on Carole's responses — hence we say that the channel allows for feedback.

Carole can make at most $k$ lies in the course of the game. At the end of the $q$ rounds Paul has won if and only if there is only one possible $\alpha \in \Omega$ for which Carole could have made her responses.

**Definition 3.** *Let $A_{C,k}(q)$ be the maximal $n$ such that there is a winning strategy for Paul in the $(n,k,C)$ game with $q$ questions.*

A winning strategy is one that wins regardless of which $\alpha$ Carole chooses and when and how she chooses to lie. Alternatively, and equivalently, we may allow Carole to play an adversary strategy. That is, she does not actually think of a possibility $\alpha \in \Omega$ but simply answers in a way consistent (under the rules of the game) with at least one $\alpha$. If at the end of the $q$ rounds her response sequence is consistent with more than one $\alpha$ then she has won.

**Remark 1.1.** *In a classic Coding Theory problem Bob sends $x \in \{1,\ldots,t\}^q$ to Alice through channel $C$ and the channel may make as many as $k$ errors. Bob's full message is one of $n$ possibilites. Is there a protocol for which correct reception of the message by Alice is assured? The answer is yes if and only if Paul wins the $(n,k,C)$ game with $q$ questions under the additional constraint that all $q$ questions must be formulated in advance. This additional constraint – anticipative versus nonanticipative, online versus offline, feedback versus no feedback – substantially changes the question. Our strategy for Paul's choice of question depends strongly on the previous responses of Carole. Let $A^-_{C,k}(q)$ denote the maximal $n$ for which Paul wins under this additional constraint. Clearly $A^-_{C,k}(q) \leq A_{C,k}(q)$ but the asymptotics of $A^-$ appear to be difficult. Indeed, even the asymptotics of $A^-_{Z,1}(q)$ are not known.*

Now we state the main result of this work.

**Theorem 1.2.** *Let $C$ be an arbitrary (fixed) $t$-ary channel with $E > 0$ potential errors. Then for any fixed $k$ in $\mathbb{N}$,*

$$A_{C,k}(q) \sim \frac{t^k}{E^k}\frac{t^q}{\binom{q}{k}} ,$$

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

**Remark 1.3.** *When $t \geq 3$, channels with the same value of $E$ can look very different. We have no elementary explanation for why the asymptotics of their functions $A_{C,k}(q)$ should be the same.*

**Remark 1.4.** *In very recent work, J. Spencer and C. Yan [8] have improved [4] and shown that for any fixed $k$*

$$A_{Z,k}(q) \sim \frac{2^{q+k}}{\binom{q}{k}} + \Theta\left(2^q q^{-k-\frac{1}{2}}\right) .$$

*While we make no conjectures, it is natural to wonder if similar bounds could be found for $A_{C,k}(q)$ when $C$ is an arbitrary $t$-channel.*

We conclude this section with two weak bounds on $A_{C,k}$.

**Theorem 1.5.** *For any $t$-ary channel $C$, $A_{C,k}(q) \leq t^q$.*

*Proof.* Suppose $n > t^q$. Even with no lying Carole wins by the simple adversary strategy of selecting that option which keeps the most possibilities viable. □

**Theorem 1.6.** *For any $t$-ary channel $C$ $A_{C,k}(q) > t^q q^{-O(1)}$*

*Proof.* We give a strategy for Paul in the $(n, k, C)$ game. Let $s = \lceil \log_t n \rceil$ so that $n \leq t^s$. Consider the possible answers as integers $0 \leq x < n \leq t^s$. Paul first asks for the $s$ digits of $x$ in base $t$. Carole's answers yield a unique $y$ that would be the answer if she hadn't lied. The number of still viable $x$ is at most

$$A = \sum_{i=0}^{k} \binom{s}{i} (t-1)^i$$

where $i$ is the number of lies that Carole has made, $\binom{s}{i}$ counts the number of possible placements of the lies, and $(t-1)^i$ bounds the number of ways to lie. Paul now starts afresh with these $A$ possibilities, rewriting them as integers $x$ with $0 \leq x < A \leq t^u$ where $u = \lceil \log_t A \rceil$. Paul asks for the $u$ digits of $x$ in base $t$, but he asks each question $2k+1$ times. Since Carole can only lie $k$ times, she must give the correct answer at least $k+1$ times. Thus Paul will know with certainty the correct answers and therefore will know $x$.

Paul's strategy has taken a total of $s + (2k+1)u$ questions. Asymptotically (with $k, t$ fixed and $n$, and hence $s$ approaching infinity), $A = O(s^k) = O(\log_t^k s)$ so the number of questions may be expressed in the form $q = \log_t n + O(\log_t \log_t n)$. We invert this function to state Theorem 1.6 in a form consistent with that of our main result, Theorem 1.2. $\square$

**Remark 1.7.** *In some research in this area the number of questions $q$ is treated as a function of the number of possibilities $n$. We could certainly set $A^*_{C,k}(n)$ to be the minimal $q$ such that Paul wins. Total knowledge of the function $A_{C,k}(q)$ yeilds total knowledge of the function $A^*_{C,k}(n)$, and conversely. Still, we note that our weak bounds give*

$$A^*_{C,k}(n) = \log_t n + O(\log_t \log_t n) = (1 + o(1)) \log_t n$$

*which is an asymptotic formula for $A^*_{C,k}(n)$. This is very different from, and much weaker than, the asymptotic formula for $A_{C,k}(q)$ that we present!*

**Remark 1.8.** *The name "Paul" honors the great questioner, Paul Erdős. "Carole" is an anagram for "Oracle".*

# 2 Setting up the problem

There are two perspectives from which this problem can be viewed, and we will need to use both of them in order to construct our asymptotic bounds.

## 2.1 Vector format and relaxation

The first perspective is a very natural one. We can describe any intermediary state in the game (after a certain number of questions have been asked and answers have been given) by a $(k+1)$-dimensional vector.

For any possibility $\alpha \in \Omega$ we compute how many times Carole has already lied if $\alpha$ is her answer. For $0 \leq i \leq k$ let $x_i$ be the number of $\alpha$ for which the number of lies is $i$. The state is then given by the vector $\vec{x} = (x_0, \ldots, x_k)$.

Paul's question is a partition of $\Omega$ into $A_1 \cup \ldots \cup A_t$. For $1 \leq j \leq t$ and $0 \leq i \leq k$ let $a_i^j$ denote the number of $\alpha \in A_j$ for which Carole has already lied $i$ times. These $a_i^j$ (up to a permutation of the possibilities) determine the question.

In the vector format a question then becomes an ordered set of $t$ $(k+1)$-dimensional vectors

$$\big((a_0^1,\ldots,a_k^1),\ldots,(a_0^t,\ldots,a_k^t)\big)$$

which constitutes a partition of the state vector $(x_0,\ldots,x_k)$. By this we mean that

$$\sum_{j=1}^{t} a_i^j = x_i,\ \forall i = 0,\ldots,k\ .$$

To fulfill all conditions, one must have that each $a_i^j$ is a non-negative integer.

Once asked this "question", Carole answers by picking one of the $t$ vectors (say, vector $l$). We can now determine the new state.

Let $0 \le i \le k$ and suppose that she has, including this last answer, told exactly $i$ lies. There are two cases, depending on the veracity of her last answer.

- If her last answer was truthful then she had told exactly $i$ lies up to that point and there are $a_i^l$ such possibilities.

- If her last answer was a lie, then one has to consider all the $x_{i-1} = \sum_j a_{i-1}^j$ for which there had been precisely $i-1$ lies. However, due to the channel's properties, not all will still be valid after she makes her choice. Namely, only those possibilities $p$ for which $(p,l)$ is an edge in the channel will be allowable. Hence there are $\sum\limits_{p\in L(l)} a_{i-1}^p$ such possibilities.

This implies that once she picks vector $l$, the state position should be reset to

$$(a_0^l,\ a_1^l + \sum_{p\in L(l)} a_0^p,\ a_2^l + \sum_{p\in L(l)} a_1^p,\ldots,\ a_k^l + \sum_{p\in L(l)} a_{k-1}^p)\ . \tag{1}$$

The $(n,k,C)$ liargame with $q$ questions can then be described in vector format, without reference to lying. The initial state is $(n,0,\ldots,0)$, a vector with $(k+1)$ coordinates. Each round Paul gives a partition of the state $\vec{a}$, Carole gives an $l \in \{1,\ldots,t\}$, and $\vec{a}$ is reset according to Equation 1. Carole may not select $l$ so that the reset $\vec{a} = \vec{0}$.

The game takes $q$ rounds, and Paul wins if the final position $\vec{a}$ has one coordinate one and all the others zero.

We now introduce a slightly more general way of playing the game.

Let $\vec{x} = (x_0, \ldots, x_k)$. We define the $(\vec{x}, k, C)$ liargame with $q$ questions. It has (in the vector format) the same rules as the $(n, k, C)$ liargame with $q$ question except that the initial state is $\vec{x}$. We may also express this in the original game format. Let $\Omega_i$, $0 \leq i \leq k$, be disjoint sets with $|\Omega_i| = x_i$ and set $\Omega = \Omega_0 \cup \ldots \cup \Omega_k$. Carole thinks of an $\alpha \in \Omega$. If $\alpha \in \Omega_{k-s}$ then she may lie at most $s$ times.

We note a simple dominance principle: If $\vec{x} \leq \vec{y}$ coordinatewise and Paul wins the $(\vec{y}, k, C)$ liargame with $q$ questions then he wins the $(\vec{x}, k, C)$ liargame with $q$ questions. This is easiest to see in the game format as the $(\vec{x}, k, C)$ liargame may be thought of as derived from the $(\vec{y}, k, C)$ by eliminating some possibilities, thus making things easier for Paul.

In the course of the paper it shall be useful to consider the following relaxation of the vector format.

**Definition 4.** *The* relaxed variant *of the $(n, k, C)$-liargame has the same rules as above except that we allow the question to have coordinates $a^i_j$ which are negative integers.*

The notions of winning and losing in the relaxed variant shall not concern us. The relaxed variant shall only be an auxilliary aid in analyzing the actual game.

### 2.2 $k$-set format and maximum size of a $k$-set

In this section we provide a somewhat more abstract format to our game, by re-introducing two concepts already mentioned in our previous work, [4]. We will take the concepts of a $k$-tree and a $k$-set introduced there, and slightly modify them to adapt them to the current problem. First recall the familiar $\delta$ function.

**Definition 5.** *Given two points in $\{1, \ldots, t\}^q$, $w = w_1 w_2 \ldots w_q$ 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 6.** *We define a $k$-tree to be a rooted tree of depth at most $k$ whose vertices are points of $\{1, \ldots, t\}^q$, with the following properties:*

1. *Denote the root by $r = r_1 r_2 \dots r_q$. For each $1 \le i \le q$ and each $y$ with $(r_i, y) \in C$ there exists exactly one child $r'$ of $r$ with $\delta(r, r') = i$ and with the $i$th position of $r'$ being $y$. Moreover, these are all the children of $r$;*

2. *Let $r' = r'_1 r'_2 \dots r'_q$ be a non-root point, with parent $r^*$ and at depth less than $k$. For each $i > \delta(r', r^*)$ and each $y$ with $(r_i, y) \in C$ there exists precisely one child $\tilde{r}$ of $r'$ such that $\delta(\tilde{r}, r') = i$ and the $i$th position of $\tilde{r}$ is $y$. Moreover, these are all the children of $r'$.*

**Definition 7.** *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.

**Definition 8.** *The birthtime of a vertex $s$ in a $k$-set, denoted $B(s)$, is zero when $s$ is the stem, otherwise $\delta(s, s^*)$ where $s^*$ is the parent of $s$.*

To exemplify these definitions, we have included Figure 3.

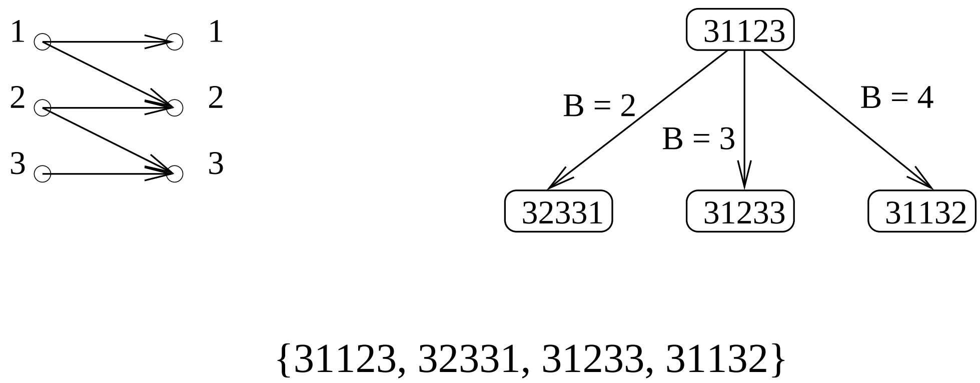


Figure 3: A 3-ary channel $C$, a 1-tree, and its corresponding 1-set. The birthtime of the vertices is recorded on the arrows from the stem.

**Remark 2.1.** *Any point in $\{1, \dots, t\}^q$ is a 0-set.*

The size of a $k$-set can vary. Suppose, for example, that $1 \le i \le t$ and there is no $j$ with $(i, j) \in C$. The stem $i \cdots i$ constitutes then a $k$-tree, for $k$, $q$ arbitrary (since there is no way to lie). The maximal size of a $k$-set depends on $C$, but here we give a useful universal upper bound.

**Lemma 2.2.** *The maximum size of a $k$-set is at most*

$$\sum_{i=0}^{k} \binom{q}{i} (t-1)^i \ .$$

*Proof.* The number of nodes at level $i$ cannot be larger than $\binom{q}{i}(t-1)^i$. Indeed, let $w$ be a node on level $i$, and let $r = r^0, r^1, \ldots, r^i = w$ be the path from the root $r$ to $w$. For $1 \le j \le i$ let $p_j$ be the birthtime of $r^j$. There are $\binom{q}{i}$ choices for the $p_j$. Fixing the $p_j$ if one knew precisely which errors (the $p_j$-th coordinate of $r^j$ for each $j$) have been committed to get to $w$, $w$ would be completely determined. But there are at most $(t-1)^i$ ways of choosing these errors. Hence there are at most $\binom{q}{i}(t-1)^i$ choices for $w$. □

## 2.3 Packing $\equiv$ winning

We begin with some technical results on $k$-trees. Let $H$ (for history) denote the set of words $r = r_1 \cdots r_m$ with $m < q$ and all $r_i \in \{1, \ldots, t\}$. This includes the null word. Let $L$ (for leaves) denote the set of words $r = r_1 \cdots r_q$.

It is useful to imagine the complete $t$-ary tree of depth $q$. The vertices are naturally associated with elements of either $L$ or of $H$, depending on whether they are leaves or interior vertices. We can further associate $r \in L$ with the path from the root to the leaf labelled $r$. The $k$-tree then has a natural picture. The root $r$ corresponds to a path in the complete $t$-ary tree. The children of $r$ are paths that break off from this path at interior points. The level at which they break off (more precisely, the first level at which they are different) is exactly one less than their birthtime (if the empty set, the root, is at level 0). See Figure 4.

The possible breakoffs at $i$ are determined by the channel $C$ and the value $r_i$. When $r'$ is a child of $r$ its children are determined by the same procedure, except that they must break off after $r'$ broke off from $r$.

**Lemma 2.3.** *Let $S$ be a $k$-set, $r = r_1 \cdots r_q \in S$, $r$ not the stem, $i$ the birthtime of $r$. The elements of $S$ with prefix $r_1 \cdots r_i$ are precisely the descendents of $r$, including $r$ itself.*

*Proof.* The descendents of $r$ change at higher coordinates and so all have prefix $r_1 \cdots r_i$. Suppose $s \in S$ had prefix $r_1 \cdots r_i$. Let $r^0, \ldots, r^u = r$ be the path in the $k$-tree from the

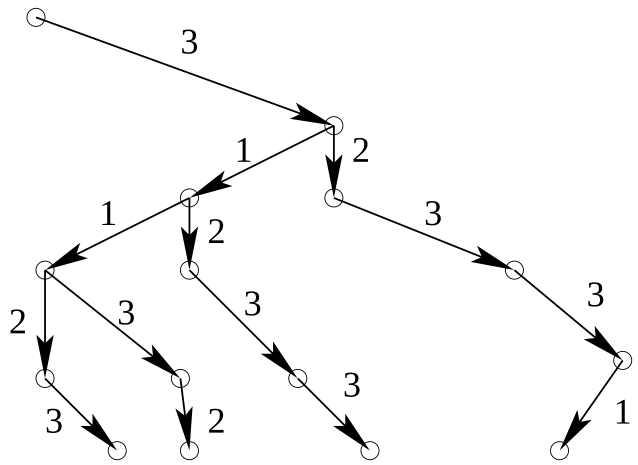


Figure 4: "Packing" the 1-set of Figure 3. Note that all 4 vertices are packed as paths, with the stem being the leftmost.

stem $r^0$ to $r$. Then $s, r$ would have a lowest common ancestor, some $r^v$ with $0 \leq v \leq u$. If $v = u$ then $s$ is a descendent of $r$. If $v < u$ let $j$ be the birthtime of $r^{v+1}$ so that $j \leq i$. Let $r^0, \ldots, r^v, s^{v+1}, \ldots, s$ be the path in the $k$-tree from the stem $r^0$ to $s$. Let $j'$ be the birthtime of $s^{v+1}$. When $j = j'$ the $j$-th coordinates of $r^{v+1}, s^{v+1}$ are different as they are different children of $r^v$. When $j < j'$ the $j$-th coordinates of $r^{v+1}, s^{v+1}$ are different since the $j$-th coordinate of $s^{v+1}$ and $r^v$ are the same. Similarly, when $j' < j$ the $j'$-th coordinates of $r^{v+1}, s^{v+1}$ are different. In all cases, setting $J = \min(j, j')$, $s^{v+1}$ does not have prefix $r_1 \cdots r_J$. All of its descendents change at higher coordinates and so $s$ cannot have prefix $r_1 \cdots r_J$. $\square$

**Lemma 2.4.** *Let $r = r_1 \cdots r_i \in H$ (including the null word with $i = 0$) and let $S$ be a $k$-set. There is at most one $s \in S$ with prefix $r$ and birthtime $B(s) \leq i$.*

*Proof.* When $i = 0$, $B(s) \leq 0$ and so $s$ must be the stem. Assume $i \neq 0$. Suppose two $s, s' \in S$ had this property. $s'$ has prefix $r_1 \cdots r_{B(s)}$. By the previous lemma $s'$ must be a descendent of $s$. (When $s$ is the stem the lemma does not apply but then $s'$ is automatically a descendent of $s$.) But, switching roles, $s$ is a descendent of $s'$ and hence they are equal. $\square$

Our next theorem connects the vector format and $k$-trees.

**Theorem 2.5.** *Let $\vec{x} = (x_0, \ldots, x_k)$. Paul wins the $(\vec{x}, k, C)$ liargame in $q$ questions if and only if there exist $x_{k-i}$ $i$-sets, $0 \leq i \leq k$, all disjoint.*

*Proof.* Let $\Omega = \Omega_0 \cup \ldots \cup \Omega_k$ denote the set of possibilities, where if $\alpha \in \Omega_s$ Carole may lie at most $k - s$ times. For Paul to win he must have a Decision Tree strategy. For each $r \in H$ Paul has a partition

$$\Omega = A_{r_1} \cup \ldots \cup A_{r_t}$$

The $r \in L$ correspond to response sequences Carole may give. For each $\alpha \in \Omega$ let $S_\alpha$ denote the set of response sequences $r \in L$ that Carole can make when her answer is $\alpha$.

Suppose $\alpha \in \Omega_{k-s}$. We claim $S_\alpha$ must form an $s$-tree. Its stem is the response sequence when Carole always answers truthfully. For $s = 0$ this is all of $S_\alpha$ and all of the 0-tree. Otherwise, let $r_1 \cdots r_q$ be the stem, $1 \leq i \leq q$, and $(r_i, y) \in C$. There must be a response sequence in which Carole responds $y$ in the $i$-th round and otherwise tells no lies. This gives a child $r'$ of $r$ with birthtime $i$ and with $i$-th position $y$. Now let $r' = r'_1 \cdots r'_q$ be any nonroot point with birthtime $j$ and at depth $s' < s$. For such response sequences (formally, by induction) Carole lies $s'$ times and the last lie is in round $j$. Let $i > j$ and $(r'_i, y) \in C$. Then there must be a response sequence identical with $r'$ in the first $i - 1$ rounds in which Carole responds $y$ in the $i$-th round and then makes no further lies. This $\tilde{r}$ is then a child of $r'$.

The $S_\alpha$, $\alpha \in \Omega$, must be disjoint for if Carole gives a response sequence $r \in S_\alpha \cap S_\beta$ then Paul cannot distinguish between the possibities $\alpha, \beta$. That is, Paul having a winning strategy implies the existence of the disjoint $s$-sets.

Conversely, suppose $S_\alpha$ are disjoint $s$-sets, defined for each $\alpha \in \Omega = \Omega_0 \cup \ldots \cup \Omega_k$. Paul now creates his Decision Tree. For $r \in H$ and $\alpha \in \Omega$ let $F(r, \alpha)$ denote that unique $y$ such that $\alpha \in A_{r_y}$. (That is, $y$ is the nonlie answer when the previous response sequence is $r$ and Carole is thinking of $\alpha$.) Fix $\alpha \in \Omega_{k-s}$ and $s$-set $S_\alpha$. Let $r_1 \cdots r_q$ be the stem of $S_\alpha$. For any proper prefix $r = r_1 \cdots r_{i-1}$ of the stem (including the null word) set $F(r, \alpha) = r_i$. This forces Carole's nonlie response sequence to be the stem. Let $r' = r'_1 \cdots r'_q$ be a nonroot point, with birthtime $j$ and depth less than $s$. For each $i > j$ set $F(r'_1 \cdots r'_{i-1}, \alpha) = r'_i$.

Lemma 2.4 insures that no value of $F$ is being set twice. When Carole's response sequence agrees with $r'$ up to and including the $j$-th round and then has no further lies this forces the response to be $r'$. All other values of $F(r,\alpha)$ may be set arbitrarily. Paul has forced $S_\alpha$ to be the set of possible responses by Carole when her answer is $\alpha$. But Paul may do this for all $\alpha \in \Omega$ simultaneously since all $F(r,\alpha)$ may be decided independently. In the game with this Decision Tree whatever $\alpha$ Carole is thinking of she must respond with some $r = r_1 \cdots r_q \in S_\alpha$. As the $S_\alpha$ are disjoint Paul can then deduce the value of $\alpha$. $\square$

# 3 Lower Bounds

Fix $\alpha < \left(\frac{t}{E}\right)^k$. Here we show that for $q$ sufficiently large and any $n < \alpha \frac{t^q}{\binom{q}{k}}$ Paul wins the $(n,k,C)$ liargame in $q$ questions. We first give an overview of Paul's strategy. First, Paul gives ground and increases $n$ to a number of the form $n = at^s$ with $a$ bounded. Now we employ the vector format. Paul further gives ground and begins at position $(ac_0t^s, \ldots, ac_kt^{s-k})$ with $c_0 = 1, \ldots, c_k$ constants given by Theorem 3.6. Paul then gives $s-k$ perfect splits, as defined below. The remaining position $(x_0, \ldots, x_k)$ with $r$ rounds remaining has $x_k = (1-\Omega(1))t^r$ and all other $x_i$ suitably small and Paul employs an endgame strategy using $k$-sets.

## 3.1 Reduction to $at^s$

We will only consider $n$ to be of the form $at^s$, where $a$ will be a "small" (bounded) parameter. In this subsection, we give the reasons for this reduction.

**Lemma 3.1.** *For any $\alpha < \alpha' < \left(\frac{t}{E}\right)^k$, there exist $T$ and $q_0$ such that for any $q \geq q_0$ and for any $n < \alpha \frac{t^q}{\binom{q}{k}}$, there exists a number $at^s$ with $a \in (t^T,\ t^{T+1}] \cap \mathbb{N}$ such that $n \leq at^s < \alpha' \frac{t^q}{\binom{q}{k}}$.*

Roughly speaking, this only says that by taking $T$ sufficiently large (yet fixed!), numbers of the form $at^s$ are sufficiently dense – we refer to Lemma 3.3 of our earlier work [4] for a detailed argument for $t = 2$ and $E = 1$.

Hence it suffices for Paul to win when $n < \alpha' \frac{t^q}{\binom{q}{k}}$ and $n$ is of the form $at^s$. For convenience we replace $\alpha'$ by $\alpha$ and assume $n$ is of this form for the remainder of this section.

### 3.2 Perfect splits

In this section we define a crucial element of Paul's strategy: the perfect split. We employ the vector format.

**Definition 9.** *From position $(x_0, \dots, x_k)$ we say that we can make a perfect split if there exists an allowable question $((a_0^1, \dots, a_k^1),\ (a_0^2, \dots, a_k^2), \dots, (a_0^t, \dots, a_k^t))$ such that all $t$ possible outcomes are the same.*

In a certain sense, the perfect splits are crucial, because they provide perfect balance. To provide an example, with $k = 2$ and the channel $C$ (see Figure 5), from $(27, 9, 12)$, by asking the question set $\{(9, 0, 1), (9, 9, 10), (9, 0, 1)\}$, the outcome in every case is $(9, 9, 10)$.

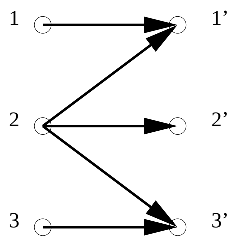


Figure 5: Channel $C$

If the perfect split exists, one can compute it algorithmically. From position $(x_0, \dots, x_k)$, one first computes $a_0^j$, with $j = 1 \dots t$. Note that $\sum_{j=1}^{t} a_0^j = x_0$. Furthermore, the zero-th coordinate of the reset position when Carole selects $j$ is $a_0^j$. Thus all $a_0^j$ must be equal and hence all $a_0^j = \frac{1}{t} x_0$.

To compute $a_i^j$, $j = 1, \dots, t$, with $i > 0$, one will rely on two things: knowledge of the previous values $a_{i-1}^j$, and knowledge of the channel $C$. We will later see that the only asymptotically essential piece of information other than $t$ is the number $E$ of potential errors.

We add an extra variable $X$ which represents the value of the $i$-th coordinate of the reset state vector. This must be the same for all $1 \leq j \leq t$. One obtains the following set of equations:

$$
\begin{aligned}
\sum_{j=1}^{t} a_i^j &= x_i \\
a_i^j + \sum_{l \in L(j)} a_{i-1}^l &= X \quad , \forall j = 1, \ldots, t \quad .
\end{aligned}
$$

The above is a system of $t+1$ equations with $t+1$ unknowns, which has a unique solution, given by

$$
X = \frac{1}{t}\left[x_i + \sum_{j=1}^{t} \sum_{l \in L(j)} a_{i-1}^l\right] \tag{2}
$$

$$
a_i^j = X - \sum_{l \in L(j)} a_{i-1}^l \quad , \forall j = 1, \ldots, t \quad . \tag{3}
$$

**Remark 3.2.** *From the system of equations (2), (3), it follows that there are two ways in which the question set can fail to be allowable: first, for reasons of integrality, and second, because one cannot ask negative questions. To be allowable, a question must have the three specified characteristics: all $a_i^j$ must be non-negative, all $a_i^j$ must be integral, and all $\sum_{j=1}^{t} a_i^j = x_i$.*

We recall Definition 4 of relaxation. The following theorem can now be easily proved.

**Theorem 3.3.** *In the relaxed variant of the game, from the initial position*

$$(c_0 t^s, c_1 t^{s-1}, c_2 t^{s-2}, \ldots, c_k t^{s-k})$$

*with $c_i \in \mathbb{N}$, $\forall i = 0, \ldots, k$, one can make a perfect split with outcome*

$$(d_0 t^{s-1}, d_1 t^{s-2}, d_2 t^{s-3}, \ldots, d_k t^{s-k-1})$$

*with $d_i \in \mathbb{N}$, $\forall i = 0, \ldots, k$. Here $s \in \mathbb{N}$, $s \geq k$.*

*Proof.* All we need to check is that the integrality condition is insured by the algorithm we presented. Clearly this is true for the first entry of the questions and the outcome. Moreover, the power of $t$ in the outcome and the questions is one less than in the original position.

To show the rest, we use induction. Our induction hypothesis (over $i$) is that all $a_{i-1}^j$, $j = 1, \ldots, t$, are integers and contain at least $s - i$ powers of $t$. For $i = 1$ this holds as all $a_0^j = c_0 t^{s-1}$. Assume it holds for some $i \leq k$. By equation (2), it follows that $X$, the $i$-th entry of the outcome vector, is an integer, and contains at least $s - i - 1$ powers of $t$. By the equations (3), it follows that the same thing is true for each $a_i^j$, and the induction is complete. We may therefore write $X = d_i t^{s-i-1}$ with $d_i \in \mathbb{N}$. $\square$

There is one more observation that will be needed.

**Lemma 3.4.** *Using the notation of Theorem 3.3, $d_i$ is a linear combination of $c_0, \ldots, c_i$, with the coefficient of $c_i$ being 1, and the coefficient of $c_{i-1}$ being $E$, for all $i = 1, \ldots, k$.*

*Proof.* We use induction on $i$. Our induction hypothesis is twofold:

- $d_i$ is an integer linear combination of $c_0, \ldots, c_i$, with the coefficient of $c_i$ being 1, and the coefficient of $c_{i-1}$ (when $i > 0$) being $E$.
- For all $1 \leq j \leq t$ we may write $a_i^j = t^{s-i-1} b_i^j$ where the $b_i^j$ are integer linear combinations of $c_0, \ldots, c_i$. The coefficient of $c_i$ is one in each $b_i^j$.

When $i = 0$ this holds as all $a_0^j = c_0 t^{s-1}$ and $d_0 = c_0 t^{s-1}$. Suppose it holds for $i - 1$. We examine equation (2), noting $x_i = c_i t^{s-i}$. Since the outcome answer is $X$, it follows that $d_i t^{s-i-1} = X$. But

$$X = \frac{1}{t}\left[x_i + \sum_{j=1}^{t} \sum_{l \in L(j)} a_{i-1}^l\right] = t^{s-i-1}\left[c_i + \sum_{j=1}^{t} \sum_{l \in L(j)} b_{i-1}^l\right]$$

So

$$d_i = c_i + \sum_{j=1}^{t} \sum_{l \in L(j)} b_{i-1}^l$$

By induction the $b_{i-1}^l$ are all integer linear combinations of $c_0, \ldots, c_{i-1}$ hence so is the double sum. Further, for each $b_{i-1}^l$ the coefficient of $c_{i-1}$ is one and so the total coefficient of $c_{i-1}$ is $\sum_{j=1}^{t} \sum_{l \in L(j)} 1 = E$. This gives the first part of the induction hypothesis. Applying equation (3), with $a_i^j = t^{s-i-1} b_i^j$, gives

$$b_i^j = d_i - \sum_{l \in L(j)} t b_{i-1}^l$$

As $d_i$ and, by induction, all $b^l_{i-1}$ are integer linear combinations of $c_0, \ldots, c_i$ so is $b^j_i$. Moreover, the $b^l_{i-1}$ are combinations of only $c_0, \ldots, c_{i-1}$ so that the coefficient of $c_i$ in $b^j_i$ is the coefficient of $c_i$ in $d_i$ which is one. The completes the second part of the induction hypothesis. □

### 3.3 Iterating perfect splits

In the previous section we have introduced the concept of a perfect split; now we will show how one can use that concept in order to provide a strategy.

**Lemma 3.5.** *Starting from position $(c_0t^s, c_1t^{s-1}, c_2t^{s-2}, \ldots, c_kt^{s-k})$, under the relaxed variant, one can make $s-k$ perfect splits.*

*Proof.* Under the relaxation assumptions we do not worry about whether the questions we ask are positive, as long as they as integrality is verified. By Theorem 3.3 and Lemma 3.4, we can make a split and be at $(d_0t^{s-1}, d_1t^{s-2}, d_2t^{s-3}, \ldots, d_kt^{s-k-1})$, with $d_i$ being integer linear combinations of the $c_i$'s. Hence we can iterate the procedure, as long as the last (lowest) power of $t$ in the $k+1$st entry is non-zero. And since at each step that power is reduced by at most one, it follows that we can do it at least $s-k$ times. □

Of course, we can formally do these splits for as long as we wish; they will not be useful in the real game unless the questions we ask are allowable (so the positivity condition is fulfilled). Here is why the next theorem is crucial.

**Theorem 3.6.** *There exist $c_0 = 1, c_1, c_2, \ldots, c_k$, such that the positivity conditions are always fulfilled, if one starts in intial position $(c_0t^s, c_1t^{s-1}, c_2t^{s-2}, \ldots, c_kt^{s-k})$, and makes $s-k$ perfect splits. Further, the $c_i$ depend only on the channel $C$ and on $k$ and not on $s$.*

*Proof.* For $0 \leq m \leq s-k$ let

$$(d^m_0 t^{s-m}, d^m_1 t^{s-m-1}, \ldots, d^m_k t^{s-k-m})$$

denote the position in the relaxed game after $m$ perfect splits. Here $d^0_i = c_i$ for convenience.

Note that $d^m_0 = c_0$, for all $m$.

The proof is based on two observations. The first one is that the way in which the coefficients $d_i^m$ evolve over the course of the $s-k$ splits is polynomially. The second one is that choosing the $c_i$'s can be done incrementally, in other words, choosing $c_i$ will not depend on any of the $c_j$'s with $j > i$.

To prove the second observation, we go back to Lemma 3.4. We may express this by saying there is a matrix $A = (a_{ij})$ with indices $0 \leq i, j \leq k$ with all $a_{ij}$ integral so that

$$d_i = \sum_j a_{ij} c_i$$

Further, $a_{ii} = 1$, $a_{i,i-1} = E$ for $1 \leq i \leq k$ and $a_{i,i+j} = 0$ for $j > 0$. Then the $d_i^m$ are derived by application of Lemma 3.4 $m$ times. Thus

$$d_i^m = \sum_j a_{ij}^{(m)} c_i$$

where $a_{ij}^{(m)}$ denotes the $i, j$ entry of $A^m$.

**Claim 3.6.1.** *1.* $a_{i,i+j}^{(m)} = 0$ *for* $j > 0$

*2.* $a_{ii}^{(m)} = 1$ *for all* $0 \leq i \leq k$

*3.* $a_{i,i-1}^{(m)} = mE$ *for all* $1 \leq i \leq k$

*4. For all* $0 < j \leq i \leq k$, $a_{i,i-j}^{(m)}$ *is a polynomial in* $m$ *of degree* $j$ *and*

$$a_{i,i-j}^{(m)} = E^j \binom{m}{j} + O(m^{j-1})$$

*Proof.* We use only the fact that $A$ is an upper trianglular matrix with ones on the main diagonal and constant $E$ on the off-diagonal. The first two properties are immediate. The third follows immediately from the recursion

$$a_{i,i-1}^{(m)} = a_{i,i-1}^{(m-1)} a_{i-1,i-1} + a_{i,i}^{(m-1)} a_{i,i-1} = a_{i,i-1}^{(m-1)} + E$$

The final part is shown by induction on $j$, the case $j = 1$ being the third part. Writing $A^m = A^{m-1}A$ gives the recursion

$$a_{i,i-j}^{(m)} = \sum_{s=0}^{j} a_{i,i-s}^{(m-1)} a_{i-s,i-j} = a_{i,i-j}^{(m-1)} + [E a_{i,i-(j-1)}^{(m-1)} + \sum_{s=0}^{j-2} a_{i,i-s}^{(m-1)} a_{i-s,i-j}] \ .$$

The induction hypothesis gives that the bracketed term is a polynomial in $m$ which may be written $EE^{j-1}\binom{m}{j-1} + O(m^{j-2})$. The difference calculus then gives that $a^{(m)}_{i,i-j}$ is a polynomial in $m$ which may be written $E^j\binom{m}{j} + O(m^{j-1})$. $\square$

We use this information to pick the $c_i$s in such a way that the questions $a_i^j$ are positive integers at any step $m$ of the iteration of $s-k$ perfect splits.

**Claim 3.6.2.** *If $(x_0, \ldots, x_k)$ has $x_i \geq Etx_{i-1}$ for all $1 \leq i \leq k$ and all $x_i$ nonnegative then the $a_i^j$ given by equations (2,3) are nonnegative.*

*Proof.* As $a_0^j = \frac{1}{t}x_0$ it is nonnegative. Suppose, by induction, that all $a_{i-1}^j$, $1 \leq j \leq t$, are nonnegative. Equation (2) then gives $X \geq \frac{1}{t}x_i$. As they sum to $x_{i-1}$ all $a_{i-1}^j \leq x_{i-1}$. As $|L(j)| \leq E$ equation (3) then gives

$$a_i^j \geq \frac{1}{t}x_i - Ex_{i-1} ,$$

and the right hand side is nonnegative by hypothesis. $\square$

Now we prove Theorem 3.6. From the claims it suffices to find $c_0 = 1, \ldots, c_k$ such that $d_i^m \geq Ed_{i-1}^m$ for all $1 \leq i \leq k$ and all integers $m \geq 0$. We show by induction on $i$ that there exists $c_i$ such that $d_i^m \geq Ed_{i-1}^m$ for all integers $m \geq 0$. For $i = 1$ we require $c_1 + mc_0 \geq Ed_0^m = Ec_0$ for all $m \geq 0$. It suffices to take $c_1 \geq E$.

To complete the induction we will need the following Lemma.

**Lemma 3.7.** *Given a polynomial $p$ of degree $u$ and a polynomial $q$ of degree $v$, $u > v$, such that the coefficients of the highest order terms in both $p$ and $q$ are positive, there exists a constant $c$ such that $p(x) + c > q(x)$ for all $x \geq 0$.*

*Proof.* The proof is easy; since $p - q$ is a polynomial of degree $u$ with highest order term being positive, it has a global minimum $\mu$ on $[0, \infty)$. Taking $c > -\mu$ insures that $p(x) - q(x) + c$ is always positive on $[0, \infty)$. $\square$

Now we complete the induction. Assume constants $1 = c_0, \ldots, c_{i-1}$ have been found. With these constants, Claim 3.6.1 gives that $Ed_{i-1}^m$ is a polynomial $q(m)$ of degree $i-1$. Further, $d_i^m$ is $c_i$ plus an integer linear combination of the $c_0, \ldots, c_{i-1}$. The coefficient of

$c_0$ is $E^i\binom{m}{i}$ and so $d_i^m = c_i + p(m)$ where $p(m)$ has degree $i$. From the Lemma 3.7 we may find $c_i$ with $d_i^m \geq E d_{i-1}^m$ for all $m \geq 0$.

Thus the $c_i$'s can be found inductively, and our existence proof is complete. □

The final results of this section are two bounding lemmas.

**Lemma 3.8.** *Fix $\alpha, \alpha^*$ satisfying $\alpha < \alpha^* < \left(\frac{t}{E}\right)^k$. Fix constants $c_0 = 1, \ldots, c_k$ satsifying the conditions of Theorem 3.6. Fix a nonnegative integer $T$ and integer $a$ with $2^T < a \leq 2^{T+1}$. Let $s$ be the maximal integer with $at^s\binom{q}{k} \leq \alpha t^q$. Then the following holds for all sufficiently large $q$: Beginning at position $(ac_0t^s, \ldots, ac_kt^{s-k})$ and making $s-k$ perfect splits yields the position $(x_0, \ldots, x_k)$ with*

$$x_k \leq E^k \alpha^* t^{q-s}$$

*Proof.* Since $q$ is large $s$ can be made arbitrarily large. As $at^s \leq \alpha \frac{t^q}{\binom{q}{k}}$ it follows that

$$aE^k \binom{s}{k} < E^k \alpha^* t^{q-s} \ ,$$

and since that is the first-order term (asymptotics in $s$) in $x_k = d_k^{s-k}$, the conclusion follows. □

We set

$$r = q - s + k$$

which is the number of rounds remaining in a $q$ round game after $s-k$ perfect splits. The inequality $at^s \leq \alpha \frac{t^q}{\binom{q}{k}}$ implies

$$r \geq k + \log_t(\binom{q}{k} a/\alpha) = k \log_t q - O(1) \tag{4}$$

Thus as $q$ becomes large, $r$ becomes arbitrarily large. Rewriting the conclusion of Lemma 3.8 in terms of $r$ yields

$$x_k \leq (1-\gamma)t^r$$

where $\gamma = 1 - \alpha^*(E/t)^k$ a positive constant.

**Lemma 3.9.** *Under the same assumptions as in Lemma 3.8 and* $r = q - s + k$

$$\sum_{i=0}^{k} x_i = O(t^{r\frac{k-1}{k}})$$

*Proof.* From equation 4, one deduces that

$$aE^{k-1}\binom{s}{k-1} < \tilde{c}t^{r\frac{k-1}{k}} \ ,$$

where $\tilde{c}$ is an appropriately large constant. But $aE^{k-1}\binom{s}{k-1}$ is the largest order term in $\sum_{i=0}^{k-1} x_i = \sum_{i=0}^{k-1} d_i^{s-k}$. □

### 3.4 Endgame

After Paul plays $s - k$ perfect splits there are $r$ rounds remaining and the position $(x_0, \ldots, x_{k-1}, x_k)$ has been reached with

- $x_k < (1-\gamma)t^r$
- $\sum_{i=0}^{k-1} x_i < t^{r\frac{k-1}{k}}$

where $\gamma$ is a fixed positive constant. From bound 4 we may consider asymptotics in $r$.

Set $A = \sum_{i=0}^{k-1} x_i$. It suffices to show (as this allows Carole more or the same number of lies for each possibility) that Paul can win from $(A, 0, \ldots, 0, x_k)$. From the weak lower bound Theorem 1.6 $A < A_{C,k}(r)$ and so Paul can win from $(A, 0, \ldots, 0, 0)$. Therefore $A$ $k$-sets can be packed into $\{1, \ldots, t\}^r$. A $k$-set has size $O(r^k)$ (Theorem 2.2) and so these $A$ $k$-sets have total size $O(r^k t^{r\frac{k-1}{k}})$ which is $o(t^r)$. For Paul to win he needs to simultaneously also pack $x_k$ 0-sets but these are arbitrary singletons. There are still $(1 - o(1))t^r$ points of $\{1, \ldots, t\}^r$ not used and so Paul can do the simultaneous packing and he wins the game.

## 4 Upper Bounds

Establishing upper bounds for our main result turns out to be less complicated than establishing lower bounds; even so, we will need some additional definitions and one important probability result.

## 4.1 $M$-normal $k$-sets

First, we will define an $M$-normal sequence; then we will use this definition to define an $M$-normal set. This double definition follows below.

**Definition 10.** *Let $M$ be a large, but fixed, integer, and $\mathcal{A}$ be an alphabet with precisely $t$ letters. A sequence of $q$ letters shall be called $M$-normal if, once split in $M$ consecutive parts, as equally as possible (up to roundoff), each of its $M$ parts contains at least $\frac{q}{tM}\left(1-\frac{1}{M}\right)$ of each letter in the alphabet $\mathcal{A}$.*

**Definition 11.** *We shall call a $k$-set $M$-normal if all sequences it contains are $M$-normal. Otherwise, we shall call it $M$-abnormal.*

We now give a bound on the minimum size of an $M$-normal $k$-set.

**Lemma 4.1.** *Provided that $M$ is large enough, the minimum size of an $M$-normal $k$-set is at least $\binom{q}{k}\left(\frac{E}{t}\right)^k\left(1-\frac{1}{M}\right)^k\left(1-\frac{k(k-1)}{2M}\right)$.*

*Proof.* We shall divide each sequence in the set in $M$ (almost) equal parts. Due to the $M$-normality property, each part in each sequence will each contain at least $\frac{q}{tM}\left(1-\frac{1}{M}\right)$ of each letter of the alphabet.

There are $E^k$ ways in which the liar can choose her $k$ sequence of lies. To make things harder for her, let us force her to make no more than one lie per part. That is, if she commits one lie in the $l$-th group of $q/M$ questions, she will have to answer truthfully until the game enters the $l+1$st group of $q/M$ questions.

Once she decides on the sequence of lies and the specific parts of the sequence where she will place them, she still has the following choice to make. For each change $x \to y$, due to the $M$-normality of the $k$-set, she has at least $\frac{q}{tM}\left(1-\frac{1}{M}\right)$ opportunities to make the specific change in the specific part (because there are at least that many $x$'s).

Hence the size of the $k$-set has to be at least as large as

$$\binom{M}{k}E^k\left(\frac{q}{tM}\right)^k\left(1-\frac{1}{M}\right)^k ,$$

and by taking $M$ large enough so as to have

$$\binom{M}{k}k!M^{-k} \geq 1-\frac{k(k-1)}{2M} ,$$

we obtain the desired bound. $\square$

### 4.2 Non-normality is exponentially rare

In this section we shall bound the number of $M$-abnormal sequences by looking at the probability that a random sequence of $q$ letters from the alphabet of $t$ letters we pick is $M$-abnormal.

We will make use of the following result (Corollary A.14 of [1]):

**Lemma 4.2.** *Let $Y$ be the sum of mutually independent indicator random variables, $\mu = E[Y]$. For all $\epsilon > 0$,*

$$Pr[|Y - \mu| > \epsilon\mu] < 2e^{-c_\epsilon \mu} ,$$

*where $c_\epsilon > 0$ depends only on $\epsilon$.*

Let us pick a random sequence of $q$ letters from the alphabet of size $t$, and divide it as close as possible into $M$ segments of size $q/M$. Let $x$ be a letter in the alphabet. The probability that the first of the $M$ intervals contains less than $(1-1/M)q/Mt$ $x$'s becomes smaller than $2e^{-c_{M,t}q}$, by Lemma 4.2, where $c_{M,t}$ is a constant depending only on $M$ and $t$.

Since there are $M$ intervals and $t$ letters, the probability that the random sequence is $M$-abnormal is at most $2Mte^{-c_{M,t}q}$. Thus we have proved the following:

**Lemma 4.3.** *The number of $M$-abnormal sequences of length $q$ with letters from the alphabet of size $t$ is at most $t^{q(1-\tilde{c}_{M,t})}$, where $\tilde{c}_{M,t}$ is a constant depending only on $M$ and $t$.*

### 4.3 Synthesis

In this subsection we combine the two results we have established about $M$-normal and $M$-abnormal sequences into the main result of the section, namely, the upper bound.

**Theorem 4.4.** *Let $\epsilon > 0$. There exists $q_0$ sufficiently large such that for all $q \geq q_0$, for any $n$ such that Paul wins the $(n, k, C)$ game with $q$ questions starting with position*

$(n, 0, \ldots, 0)$

$$n \leq \left( \left( \frac{t}{E} \right)^k + \epsilon \right) \frac{t^q}{\binom{q}{k}} .$$

*Proof.* First, choose $M$ large enough so that

$$\frac{1}{\left(1 - \frac{1}{M}\right)^k \left(1 - \frac{k(k-1)}{2M}\right)} < 1 + \left( \frac{E}{t} \right)^k \frac{\epsilon}{2} ,$$

in addition to being large enough to fulfill all conditions in Section 4.1. Recall that $M$ is large but fixed.

Choose now $q_0$ large enough so that

$$\binom{q}{k} t^{-\tilde{c}_{M,t} q} < \left( \frac{t}{E} \right)^k \frac{\epsilon}{2} ,$$

for all $q \geq q_0$. Here $\tilde{c}_{M,t}$ is the same constant as in the previous section. This is doable, since $M$ and $t$ are fixed, and $\tilde{c}_{M,t}$ is a constant depending only on $M$ and $t$.

As we have proved in Section 2.3, in order for Paul to win, he must be able to pack $n$ $k$-sets. Some of them will be $M$-normal; since the minimum size of an $M$-normal $k$-set is at most

$$\binom{q}{k} \left( \frac{E}{t} \right)^k \left( 1 - \frac{1}{M} \right)^k \left( 1 - \frac{k(k-1)}{2M} \right) ,$$

it follows by our choices of $M$ and $q_0$ that the total number of $M$-normal $k$-sets that Paul can pack is at most

$$\left( \left( \frac{t}{E} \right)^k + \frac{\epsilon}{2} \right) \frac{t^q}{\binom{q}{k}} .$$

On the other hand, since there are at most

$$t^{q(1 - \tilde{c}_{M,t})} \leq \frac{t^q}{\binom{q}{k}} \left( \frac{t}{E} \right)^k \frac{\epsilon}{2}$$

abnormal sequences, and any abnormal $k$-set will have to conatin at least one such sequence, it follows that Paul cannot pack more than

$$\frac{t^q}{\binom{q}{k}} \left( \frac{t}{E} \right)^k \frac{\epsilon}{2}$$

abnormal $k$-sets.

Since any $k$-set is either $M$-normal or abnormal, it follows immediately that

$$\begin{aligned} n &\leq \# M\text{-normal } k\text{-sets} + \# \text{ abnormal } k\text{-sets} \\ &\leq \left( \left(\frac{t}{E}\right)^k + \epsilon \right) \frac{t^q}{\binom{q}{k}}, \end{aligned}$$

and the theorem is proved. □

**Remark 4.5.** *The upper bound argument shows that if $n$ is too large Paul cannot win as a win leads to a packing of $k$-sets. As Paul cannot win there must be an adversary strategy for Carole to win. The above argument does not give such a strategy in an explicit form.*

## 5 Acknowledgements

The authors would like to thank the *Schloss Dagstuhl* International Conference and Research Center for Computer Science, as well as the organizers of the Seminar on Algorithmic Combinatorial Game Theory (02/17 – 02/22/2002) for providing an excellent environment for research and the sharing of mathematical ideas which utimately lead to the results in this paper. In particular, the authors would like to especially thank Elwyn Berlekamp, Tomasz Łuczak, and Uri Zwick, for many fruitful conversations.

Ioana Dumitriu would like to thank the Clay Mathematics Institute, as part of this research was done while she was a Special Project Prize Fellow of CMI, during the summer of 2002. Ioana Dumitriu's research was also partially supported from NSF grant DMS-9971591.

## References

[1] N. Alon and J. Spencer, The Probabilistic Method, 2nd ed., John Wiley, 2000.

[2] 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

[3] 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.

[4] I. Dumitriu and J. Spencer, A Halfliar's Game, *Theoretical Computer Science* (to appear)

[5] A. Pelc, Searching games with errors – fifty years of coping with liars, *Theoretical Computer Science* **270** (2002), 71–109

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

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

[8] J. Spencer and C. Yan, The HalfLie Problem (submitted for publication)

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