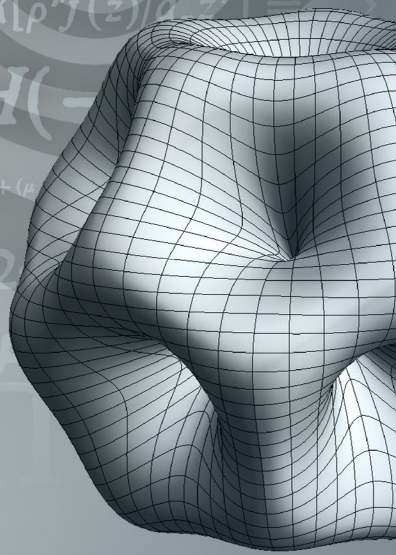
ISSN 2182-1976



Recreational Mathematics Magazine

A Random Logic Puzzle

Robert W. Smyth
Department of Mathematics, The Cooper Union
smyth@cooper.edu

Number 4
September, 2015

Ludus

CIUHCT

Ludus CIUHCT

Problems

A Random Logic Puzzle

Robert W. Smyth
Department of Mathematics, The Cooper Union
smyth@cooper.edu

Abstract: *This paper introduces and solves a challenging logic puzzle inspired by George Boolos' "Hardest Logic Puzzle Ever". The analysis hinges on a characterization of questions in terms of the relevant knowledge which may be gleaned from their answers.*

Key-words: Logic puzzle, binary trees, information theoretic bounds.

1 The Hardest Logic Puzzle Ever

In [1] George Boolos presents the following intriguing logic puzzle (together with a solution). Boolos attributes the puzzle to Raymond Smullyan and dubs it "The Hardest Logic Puzzle Ever".

(HLPE) Three gods A, B, and C are called, in some order, True, False, and Random. True always speaks truly, False always speaks falsely, but whether Random speaks truly or falsely is a completely random matter. Your task is to determine the identities of A, B, and C by asking three yes-no questions; each question must be put to exactly one god. The gods understand English, but will answer all questions in their own language, in which the words for "yes" and "no" are "da" and "ja", in some order. You do not know which word means which.

Boolos' method of solution reveals that the puzzle contains uncoupled layers of complexity that may be tackled independently. Once some god is known not to be Random, the truth value of statement S may be determined from that god's answer to "Does 'da' mean true iff (you are True iff S)?" Accordingly, in the HLPE-inspired puzzle I offer here, 'da' and 'ja' are eliminated in favor of simple 'yes' and 'no' answers, and only one species of nonrandom god is used. The puzzle focuses on the problem of separating truthful gods from random gods. This is similar to and yet, as we will see, distinctly different from the HLPE 'layer' handled by finding a god who does not answer randomly. Furthermore, to ratchet up the difficulty and inject an algorithmic analysis flavoring we will not specify an allotment of questions, but rather make the determination of an optimal strategy the heart of the puzzle.

2 The Random Logic Puzzle

Here's the random logic puzzle.

(RLP) Two of four gods are called True, and the other two are called Random. Each of the two called True always answer questions truly. The responses of the two called Random are generated randomly. Your task is to separate the Trues from the Randoms by asking as few yes-no questions as necessary. Each question must be addressed to exactly one god.

Although some clarification is in order, those puzzle enthusiasts who prefer to attack problems without even a whiff of a hint may wish to set this paper aside and take a crack at the RLP now.

3 Representing Solution Strategies

Boolos' solution to the HLPE makes essential use of the freedom to choose the second question's addressee on the basis of the answer to the first question. Indeed any solution to the HLPE must exploit this freedom. So a solution strategy is inherently more flexible (and complicated) than a simple list of questions. A strategy for the RLP may be modelled as a (finite) rooted binary tree with a question and an addressee specified at each internal node, and a partition of the gods into two pairs specified at each leaf. The root represents the opening question, each internal right child represents the question to be posed if the parent question has been answered affirmatively, and each internal left child represents the question to be posed if the parent question has been answered negatively. A strategy may be called successful if at each leaf there is no assignment of names to gods which is consistent with all the answers leading from the root to the leaf, but inconsistent with the partition given at the leaf. The RLP's task is to establish the minimum possible height of a successful strategy.

Figure 1 illustrates an unsuccessful strategy which might fittingly be called the naive strategy. The four gods have been labelled A, B, C and D arbitrarily. Notice for example that the leaf at the end of a YES, YES answer sequence is consistent with A and C as Randoms and B and D as Trues. Yet this naming scheme is inconsistent with the partition $\{\{\mathrm{A},\mathrm{B}\},\{\mathrm{C},\mathrm{D}\}\}$ at the leaf.

4 Clarifications

In [3] Rabern and Rabern propose a solution to the HLPE using only two yes-no questions. They argue that the question 'Are you going to answer this question with the word that means *no* in your language?' can not be answered truthfully

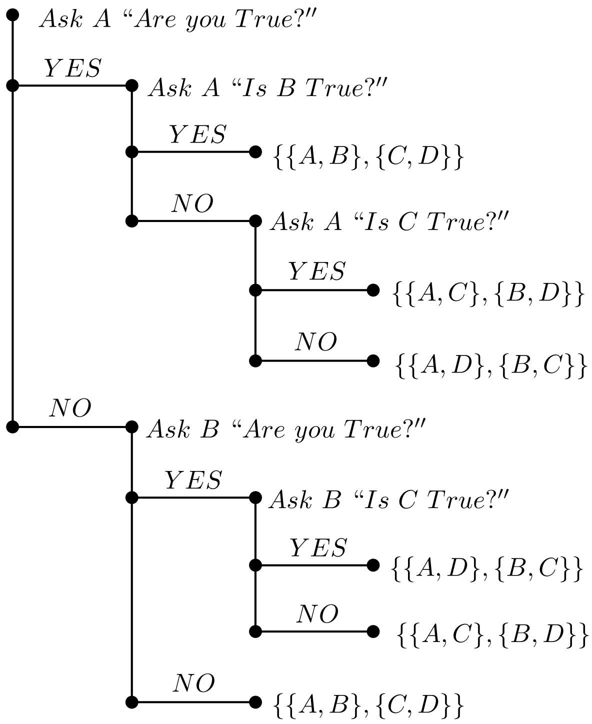


Figure 1: An unsuccessful strategy

by True, and thus, if he is the addressee of this question his head must explode revealing his identity. Notice that our model of a strategy implicitly precludes such questions. Admissible questions can not have any outcome other than a yes or no answer. (From here on this requirement on yes-no questions will be in force not just in our treatment of the RLP, but also in our continuing discussion of the HLPE.) Furthermore, it should be understood that

> Each time one of the Randoms is asked a question, the response is determined at random independently of any prior responses from that god or the other Random.
>
> Each god knows his own identity as well as the identities of the other gods.
>
> Once the gods have been separated into two pairs it is not required to specify which pair is which, and
>
> The gods are distinguishable in a manner that permits questions to reference a specific god or subset of the gods other than the addressee, and enables the questioner (and the gods themselves) to know at all times which god was addressed in any previous question, and which god(s) were referenced. (This assumption was already applied in our use of the labels A, B, C and D for the four gods.)

Those readers who haven't taken a crack at the puzzle yet are encouraged to do so now.

5 Information Theoretic Bounds

Since there's no limit to the number of questions which can be contrived, proving the optimality of a strategy may appear to be a daunting task. In the case of the HLPE, however, a simple information theoretic observation proves the optimality of any correct three question strategy. In that puzzle there are six possible assignments of names to gods that must be discriminated amongst. The answers to two yes-no questions (two bits of information) can discriminate among at most four cases. (cf. [2]) So what's the information theoretic lower bound on the number of questions required by the RLP? Since there are only three ways to partition a set of four elements into two pairs, only two bits of information are needed. Here, however, the simple information theoretic lower bound can not be achieved. We'll need a more careful accounting of what can and can not be learned from arbitrary yes-no questions.

6 The Tree of Knowledge

Let's track information by annotating each node in a strategy tree with the set of assignments of names to gods which are consistent with the answers that lead to the node. We'll call this annotated tree the strategy's tree of knowledge. (In effect we're characterizing each question by the relevant knowledge which may be gleaned from its answer. This characterization transforms the unbounded class of allowed questions into a finite, and thus readily analyzed, set.) Notice that for a successful strategy tree, the annotation at each leaf must be a subset (not necessarily proper) of the complementary pair of name assignments which are consistent with the partition specified at the leaf.

Figure 2 shows the tree of knowledge for the unsuccessful naive strategy of Figure 1. The one leaf with a "correct" partition is marked with a ✓, the others with ✗. Let's adopt a shorthand for assignments of names to gods. A string with two 'T's and two 'R's represents an assignment which gives the name True to A (resp. B, C, D) if the first (resp. second, third, fourth) character in the string is 'T', and gives the name Random to A (resp. B, C, D) if the first (resp. second, third, fourth) character is 'R'. Let $U = \{\text{TTRR}, \text{TRTR}, \text{TRRT}, \text{RTTR}, \text{RTRT}, \text{RRTT}\}$ be the collection of all possible assignments. This collection U decomposes into three mutually disjoint pairs of complementary assignments $\{\text{TTRR}, \text{RRTT}\}$, $\{\text{TRTR}, \text{RTRT}\}$ and $\{\text{TRRT}, \text{RTTR}\}$ each of which corresponds to a possible separation of the Randoms from the Trues. For convenience introduce the following notation. Let $\mathcal{N}$ be the set of all nodes in a strategy's knowledge tree. At each node $P \in \mathcal{N}$, $k(P)$ equals the subset of U stored at P. At each internal node P, $k_l(P) = k(P\text{'s left child})$ and $k_r(P) = k(P\text{'s right child})$. So in particular the tree of knowledge for any strategy satisfies $k(\text{root}) = U$. Also we clearly have $k_l(P) \subseteq k(P)$ and $k_r(P) \subseteq k(P)$ at every internal node P. Our assumption that any admissible question must result in a yes or no answer yields $k_l(P) \cup k_r(P) = k(P)$ for every internal node P. Now let's partition each collection $k(P)$ into $k^R(P)$ and $k^T(P)$ where $k^R(P)$ contains the assignments in

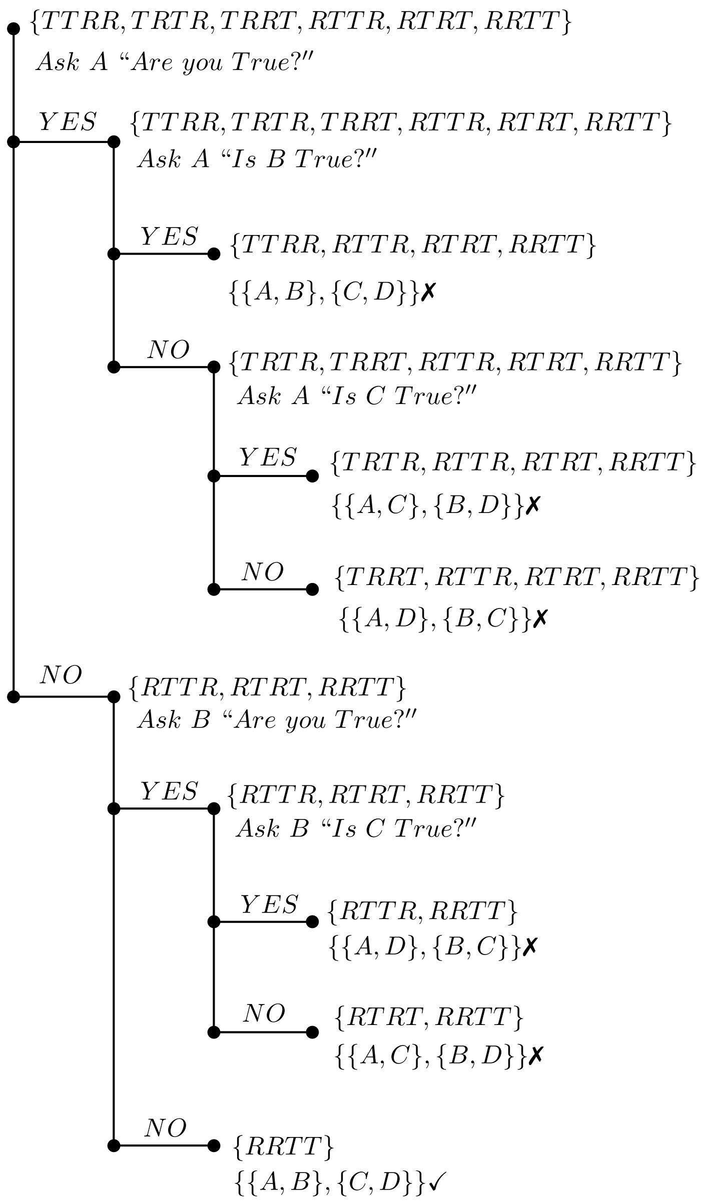


Figure 2: Naive strategy's tree of knowledge

$k(P)$ which give the name Random to the addressee of the question at P, and $k^T(P) = k(P) \setminus k^R(P)$ contains the assignments in $k(P)$ which give the name True to the addressee at P. Since each Random may answer yes or no to any question in any context, $k^R(P) \subseteq k_l(P)$ and $k^R(P) \subseteq k_r(P)$ at each internal node P.

7 Improving Lower bounds

The observations and notational conventions of the previous section prepare us to prove

The Progress Speed Limit Lemma.
For each $P \in \mathcal{N}$, max$\{|k_l(P)|, |k_r(P)|\} \geqslant |k(P)| - 1$.

In other words, there is never any question whose answer is guaranteed to enable us to eliminate more than one possible naming scheme from consideration.

Proof. The collection U itself only has three assignments that give the name True to any one god. So regardless of the addressee of the question at P, $|k^T(P)|$ must always be less than or equal to 3. The inclusion $k^T(P) \subseteq k(P) = k_l(P) \cup k_r(P)$ implies that $k^T(P) \setminus k_l(P)$ and $k^T(P) \setminus k_r(P)$ are disjoint. So $|(k^T(P) \setminus k_l(P)) \cup (k^T(P) \setminus k_r(P))| \leqslant |k^T(P)| \leqslant 3$, and at least one of the two sets $k^T(P) \setminus k_l(P)$ and $k^T(P) \setminus k_r(P)$ has size less than or equal to 1. Suppose $|k^T(P) \setminus k_l(P)| \leqslant 1$. Then since $k^R(P) \subseteq k_l(P)$ we know $k(P) \setminus k_l(P) = k^T(P) \setminus k_l(P)$. Thus $|k(P) \setminus k_l(P)| \leqslant 1$. So $|k_l(P)| \geq |k(P)| - 1$. Similarly, if $|k^T(P) \setminus k_r(P)| \leqslant 1$, then $|k_r(P)| \geqslant |k(P)| - 1$. □

As noted earlier, $|k(\text{root})| = |U| = 6$. Since there are only two assignments of names to gods compatible with each separation of the gods into pairs, for every leaf in any successful strategy tree $|k(\text{leaf})| \leqslant 2$. It thus immediately follows from the Progress Speed Limit Lemma that an optimal RLP strategy tree has a height of at least four. In other words, there is no successful strategy that never requires the use of more than three questions. Though this lower bound improves upon the simple information theoretic lower bound of two, it still fails to be tight. The following additional observation will show that a successful RLP strategy tree must in fact have a height of at least five.

Lemma. *If $|k(P)| = 3$ for some node P in a successful RLP strategy tree, then P's children can't both be leaves.*

Proof. Given $|k(P)| = 3$, $|k_l(P)|$ and $|k_r(P)|$ can't both be less than or equal to 1 since $k_l(P) \cup k_r(P) = k(P)$. Similarly, $|k_l(P)| = 2$, $|k_r(P)| = 0$ and vice versa are impossible. Furthermore $k_l(P)$ and $k_r(P)$ cannot both consist of complementary pairs. Since distinct complementary pairs are disjoint, if both $k_l(P)$ and $k_r(P)$ were complementary pairs, then $|k(P)| = |k_l(P) \cup k_r(P)|$ would be 2 or 4. Finally we need to consider (and dismiss) the possibility that one of the sets $k_l(P)$ and $k_r(P)$ is a complementary pair and the other a singleton. Suppose, for instance, $k_l(P) = \{\text{TRRT}, \text{RTTR}\}$ and $k_r(P) = \{\text{WXYZ}\}$, where WXYZ represents some element of U. If the addressee of the question at P is B or C, then TRRT$\in k^R(P) \subseteq k_r(P)$ forces WXYZ=TRRT, which is impossible. Similarly, if the addressee of the question at P is A or D, then RTTR$\in k^R(P) \subseteq k_r(P)$ forces WXYZ=RTTR, which is also impossible. The other permutations are perfectly analogous. □

From the preceding arguments and observations a successful strategy tree must have some node P with $|k(P)| = 3$. This node must be at a depth of at least 3 from the root. This node must also have some descendant with depth greater by at least 2. So the height of a successful RLP strategy tree must be at least 5. This time we've finally honed in on the answer. All that remains is to exhibit a successful strategy with a tree of height exactly 5. (This result may be labelled a theorem by any reader who feels the use of lemmas so demands.)

8 An Optimal Strategy

Though the construction of an optimal strategy leaves us with plenty of latitude, it will be convenient to exploit the notation and structures we've already introduced. Although the decomposition $k(P) = k^R(P) \cup k^T(P)$ depends on the addressee of the question at P, note that $k(P)$ itself is determined by the part of the strategy tree strictly above node P. So node P may be constructed from $k(P)$. (This is even vacuously true at the root.) If there is some X from {A, B, C, D} such that X is assigned the name True by more than one element of $k(P)$ then clearly there is some Y from {A, B, C, D}\{X} such that X and Y are both assigned True by one element, say α, of $k(P)$, while X and not Y are assigned True by another element, say β, of $k(P)$. In this case let node P be an internal node with the question "Is Y named True?" addressed to X. Otherwise, let P be a leaf. Notice that when P is made into an internal node $k_r(P)$ must contain α and not β, while $k_l(P)$ must contain β and not α. It follows both that $max\{|k_l(P)|, |k_r(P)|\} \leqslant |k(P)| - 1$ and that $k_l(P)$ and $k_r(P)$ are both nonempty. Thus any internal node P (for which $|k(P)|$ is clearly greater than or equal to 2) has depth at most four. So our construction yields a tree with depth at most five, as claimed. For any leaf P there is no X in {A, B, C, D} such that X is assigned the name True by more than one element of $k(P)$. This implies that the nonempty set $k(P)$ is either a singleton or a complementary pair. Either way the leaf P can be assigned a partition of {A, B, C, D} into two pairs which is consistent with $k(P)$. This completes the construction of a successful optimal strategy.

9 Exercises

Clearly the RLP may be viewed as a special case of the parametrized family of similar puzzles with n Trues and m Randoms. Our original statement of the RLP presumes the existence of a successful strategy. The interested reader is invited to show that this presumption is justified whenever $n \geqslant m$, but does not hold when $m > n > 0$. The final exercise left to the reader is to demonstrate the claim made earlier that any solution to the HLPE must exploit the freedom to choose the second question's addressee on the basis of the answer to the first question.

References

[1] G. Boolos. "The hardest logic puzzle ever", *The Harvard Review of Philosophy* (6), 1996.

[2] G. Brassard and P. Bratley. *Fundamentals of Algorithmics*, Englewood Cliffs, NJ: Prentice-Hall, 1996.

[3] B. Rabern and L. Rabern. “A simple solution to the hardest logic puzzle ever”, *Analysis* (68), 2008.

[4] T.S. Roberts. “Some thoughts about the hardest logic puzzle ever”, *Journal of Philosophical Logic* (30), 2001.