Just How Random Are Your Answers?

Jeffrey W. Holcomb

jholcomb@holcombtechnologies.com

Affiliations:
Holcomb Technologies

Abstract:

This paper contains the last in a series of proofs to demonstrate that P is a proper subset of NP. This proof relies upon the existence of stochastic answers in the set difference between NP and P. Essentially, since stochastic answers require both an internal measure of randomness and a likelihood of generation that is at least as low as their likelihood of occurrence, they cannot be generated by deterministic automata. Since stochastic answers can be generated by nondeterministic automata, they must belong to NP but not P. Therefore, P is indeed a proper subset of NP.

Key words/phrases: theory of computation, computational complexity, randomness in computing

Draft

Section 1 Introduction:

In March of this year we proposed redirecting the question of whether or not P is equivalent to NP to the question of whether or not stochastic answers can be produced deterministically [1]. Essentially, if we cannot produce stochastic answers deterministically then they must be produced utilizing nondeterministic means; therefore, P must be a proper subset of NP. We began to publish our proof regarding this mater in the three publications [2], [3], and [4]. The first part discusses the properties of deterministic automata with respect to our model [2]. The second part discusses the properties of nondeterministic automata implemented over a single machine with respect to our model [3]. The second part discusses nondeterministic algorithms concurrently implemented over multiple deterministic automata with respect to our model [4]. This paper presents the last in the series of arguments designed to demonstrate that P must be a proper subset of NP.

This paper is divided into six sections: *Introduction*, *Definitions*, *Background*, *Research*, *Conclusion*, and *Future Research*. Our *Definitions* section covers the definitions specifically relevant to this paper, the *Background* section covers work done by previous researchers that is relevant to this paper, the *Research* section presents the research done by our group, the *Conclusion* section is a brief synopsis of the conclusions drawn by our group, and the *Future Research* section outlines the work that still needs to be completed.

Section 2 Definitions:

Definition 1. A **perfectly random number/symbol** is a symbol or numerical value *val* that belongs to a set *vals* of unique values or symbols; where *val* cannot be reproduced from *vals* (or generated in an identical state or valuation) within a series of attempts in any manner that produces an associated probability of generation greater than 1/|*vals*|; where |*vals*| is the cardinality of *vals*, or the total number of possible evaluations for *val* [1], [5].

Draft

Definition 2. A **perfectly random set** is a set *set* of values *val* that belongs to a set *vals* of unique values or symbols that cannot be reproduced (or generated in an identical state/valuation) within a series of attempts in any manner that produces an associated probability of generation greater than $\prod(1/|vals|)$; where $\prod(1/|vals|)$ is the product of the individual probabilities associated with the likelihood of obtaining each individual value *val* in the set *set* [1], [5].

Definition 1 and Definition 2 are in contrast to pseudorandom numbers [5] and pseudorandom sets; which we will refer to in this paper as apparently random numbers and apparently random sets in order to maintain consistency of verbiage within our argument.

Definition 3. A **stochastic answer** to a problem P is an answer that has both the internal quality of randomness and the external quality of randomness in occurrence such that a problem instance $\iota_a \in P$ cannot be mapped directly to a single, individual given answer $\mathcal{A}$, but must be mapped to a set $\boldsymbol{\mathcal{A}}$ of potential answers whose individual associated probability of generation is proportional to their individual probability of occurrence [1].

Definition 3 may seem overly customized for this proof, but is actually meant to quantify a more universal property of the natural world. For example, if we shuffle a deck of cards and then randomly draw one card form the middle of the deck then the likelihood of our drawn card being any specific value will be one in fifty two. No matter how long we draw cards, so long as we return our initial card back to the deck, shuffle the deck thoroughly, and the deck is not a trick deck of cards the likelihood of drawing any given card will always be one in fifty two.

The randomness of occurrence associated with our card value is a fairly universal property that has a dramatic influence on quantum mechanics, molecular mechanics, physical chemistry, optics, biology, and many other disciplines involved in the study of the natural world.

Definition 4. A **non-stochastic answer** to a problem P is an answer that can be determined directly from the supplied problem instance ι such that $\iota \vDash \mathcal{A}$ [1].

Section 3 Background:

In March of 2011 [1] we proposed a solution to the question regarding the set equivalency of P and NP by suggesting that the existence of stochastic answers, which are incapable of being generated by a deterministic automaton, implies that P is in fact a proper subset of NP. The question of the set equivalence of P and NP has been an open question since it was first pondered by Jack Edmonds [6] in 1965.

We have been working to formalize our proposal [2], [3], [4] ever since our March publication. Here, we discuss the nature of stochastic answers and present a proof for our theorem that P $\subset$ NP. We began to formalize our proof in [2] where we presented the following arguments for deterministic automata:

Lemma 1: A deterministic automaton M can be defined via a Markov game G_M such that each pure strategy $r_{\alpha,a}$ in G_M represents an instance of M over a given input sequence ι_a: where $r_{\alpha,a}$ and ι_a exist in a one-to-one correspondence over all possible acceptable words ι_a for $\mathcal{M}$, and where each given input sequence ι_a is by definition an instance of a problem $\mathcal{P}$.

Lemma 2: The Bayesian game G_B utilized to represent all possible instantiations of a deterministic algorithm upon an automaton $\mathcal{M}$ can be reduced to a pure coordination game over the set $\Pi(\mathcal{P})$ of partitions, or algorithms, $\boldsymbol{\pi}_\alpha$ for G_B whenever the identity of the problem $\mathcal{P}$ to be solved by $\mathcal{M}$ is known.

We then began to formalize our proof with respect to nondeterministic automata in [3], where we presented the following corollary:

Corollary 1: A nondeterministic algorithm $\boldsymbol{\pi}_\alpha$ instantiated on an automaton $\mathcal{M}$ can be defined via a Markov game G_M such that a) the pure strategies $r_{\alpha,a}$ in G_M represent the instances of $\mathcal{M}$ over a given input sequence ι_a, and where at least one of the stage games G^τ is a nondeterministic stage game; and b) any given possible instance of $\boldsymbol{\pi}_\alpha$ on $\mathcal{M}$ can be defined by a pure strategy $r_{\alpha,a} \in \boldsymbol{\pi}_\alpha$.

We continued our formulation of nondeterministic automata in [4], where we presented the following lemma:

Lemma 3: A nondeterministic algorithm instantiated on a nondeterministic automaton $\mathcal{M}$ can be defined equivalently by a Bayesian game G_B composed of Markov sub-games $G_M \in \boldsymbol{G}$ where each component Markov sub-game G_M is defined over a deterministic algorithm $\boldsymbol{\pi}_\alpha$, or as a single Markov sub-game G_M which contains nondeterministic stage games $q^\tau_{non-stabilizing}$.

Note that Lemma 3 is based upon the arguments presented in both [3] and [4].

All of the main structural elements utilized by our proof are taken from temporal logic [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] and extensive form games. The arguments presented by our proof are with respect to Bayes-Nash equilibria [19] over our temporal logic structures. For our research, we are mainly interested in the game theoretic formulations for the various forms of temporal logic [9], [15], [16], [20], [21], [22], [23], [24].

Draft

Section 4 Research:

Section 4.1 Model Structure

The purpose of this section is to provide an overview of the structure utilized by the arguments in this paper to model automata. For our arguments, we are not concerned about specific implementations or applications of automata but are concerned with the natural structure of automata in general. As such, we will consider an automaton $\mathcal{M}$ as a generic automaton that is capable of solving any given solvable problem $\mathcal{P}$ for an answer $\mathcal{A}$.

Since $\mathcal{P}$ must be expressed in a manner that is understandable to $\mathcal{M}$, we will consider an instance of $\mathcal{P}$ as an acceptable word ι: where $\mathcal{P}$ itself is a set of related words. In this manner, the set **ï** of all problems $\mathcal{P}$ is also the complete set of acceptable words ι for our generic automaton $\mathcal{M}$: finite and infinite.

We define $\mathcal{M}$ formally according to the quintuple (Q, Σ, δ, q^0, F): where Q is a finite set of states for $\mathcal{M}$, Σ is $\mathcal{M}$'s input alphabet; δ is a state transition function for $\mathcal{M}$; q^0 is the initial state, or set of initial states, for $\mathcal{M}$; and F is the set of final, or accepting, states for $\mathcal{M}$ such that $F \subseteq Q$. For deterministic automata, our transition function will be defined as $\delta{:}Q\times\Sigma \rightarrow Q$ [18] [25] [26]. For nondeterministic automata, our transition function will be defined as $\delta{:}Q\times\Sigma \rightarrow 2^{Q}$ [17].

Since we have not yet applied a specific problem $\mathcal{P}$ to $\mathcal{M}$ we are not concerned about our acceptance condition for $\mathcal{M}$; however, we are concerned about the set $\ddot{\imath}$ of acceptable words $\boldsymbol{\iota}$. From $\ddot{\imath}$ we construct our Σ-tree $\boldsymbol{t}$ for $\mathcal{M}$; where $\boldsymbol{t}$ is defined according to the pair $(\nu, \ddot{\imath})$ such that $\nu : \ddot{\imath} \rightarrow \Sigma$. We then say that a run r on $\boldsymbol{t}$ is performed by $\mathcal{M}$ if r maps $\ddot{\imath}$ to Q [8] [11]; where r is a single instance of $\mathcal{M}$ and the set $\boldsymbol{r}$ of runs r forms a Q-tree $\boldsymbol{q}$ in like manner as our Σ-tree $\boldsymbol{t}$[1].

With our Σ-tree and our Q-tree defined and characterized, we can now define our high level game structure. We define our high level game G_B as a Bayesian game over $\boldsymbol{q}$, according to the quadruple $(N, \boldsymbol{G}, \vartheta, \Pi)$; where N is the set of n players for G_B, $\boldsymbol{G}$ is a set of component games for G_B, ϑ is a common prior over all component games that participate in G_B, and Π is a tuple of partitions $\boldsymbol{\pi}_\alpha$ of G_B for each agent $i \in N$. We define a partitioning

$$\Pi : \mathcal{P} \times \boldsymbol{r} \rightarrow 2^{2^{r}} \tag{1}$$

of G_B as a Bayes-Nash equilibrium $\boldsymbol{\mathcal{A}}$[2] for G_B with respect to $\mathcal{P}$, and an ordering of the pure strategies of G_B into partitions $\boldsymbol{\pi}_\alpha$ that define the component games for our current instance of G_B; where each component game is defined as a Markov game G_M over the sub-tree defined by the set of pure strategies $r_{\alpha,a} \in \boldsymbol{\pi}_\alpha$; and where our set of Markov games, for an instance of G_B, exist in a one-to-one correspondence with our set of partitions $\boldsymbol{\pi}_\alpha$ and are viewed as concurrent game structures with respect to a given problem $\mathcal{P}$. Here, we will view each partition $\boldsymbol{\pi}_\alpha$ as an independent algorithm implementable on $\mathcal{M}$. For our purposes, N will always be defined as the set {*time*, *memory*}, regardless of our instances of $\mathcal{M}$. Initially, and before we apply $\mathcal{P}$ to $\mathcal{M}$, ϑ will be defined according to the total likelihood that our players i will play a pure strategy r_a in response to any given problem instance $\boldsymbol{\iota}_a$; however, after we apply $\mathcal{P}$ to $\mathcal{M}$, thereby imparting a partitioning $\Pi(\mathcal{P}, \boldsymbol{r})$, but before we apply a specific problem instance $\mathcal{P}_\alpha$ to $\mathcal{M}$, we will induce a re-evaluation of ϑ such that ϑ will now only provide support to partitions that may be used in response to $\mathcal{P}$. As such, the individual probabilities $c_{i,\alpha,a}$ associated with each individual pure strategy $r_{\alpha,a}$ will now be proportional to the likelihood that a given pure strategy $r_{\alpha,a}$ will be played in response to a given problem instance $\boldsymbol{\iota}_a$ with respect to $\mathcal{P}$.

Section 4.2 The High Level Concept

We can answer the question of whether or not P is a proper subset of NP one of two ways: either $\mathrm{P} \subset \mathrm{NP}$ or $\mathrm{P} \approx \mathrm{NP}$. If we are to show that $\mathrm{P} \approx \mathrm{NP}$ then we must show that all problems $\mathcal{P} \in \mathrm{NP}$ have an associated Bayes-Nash equilibrium $\boldsymbol{\pi}_\alpha$ that can solve $\mathcal{P}$ deterministically while experiencing polynomial growth of resources (usually with respect to time and memory). If we are to show that $\mathrm{P} \subset \mathrm{NP}$ then we must demonstrate that there exists a problem $\mathcal{P}$ such that $\mathcal{P}$ cannot be solved by a single deterministic partition $\boldsymbol{\pi}_\alpha$.

We can see from Lemma 1 and Corollary 1 that the main difference between deterministic and nondeterministic automata is that nondeterministic automata require the existence of at least one

[1] We define a Q-tree $\boldsymbol{q}$ as a pair $(\Xi, \boldsymbol{r})$ where $\boldsymbol{r}$ is our set of runs r, $\Xi{:}\boldsymbol{r} \rightarrow Q$, and $r{:}\ \ddot{\imath} \rightarrow Q$. We define a Q-tree in this manner rather than as was done by Pnueli and Rosner in [11] because we find it more intuitive to our adaptation for a Bayesian game. That is, a run r can now be interpreted directly as a sequence of state-input pairs that are easily reinterpreted as pure strategies for our Bayesian game. Ultimately, defining our Q-tree this way enables us to define the probability distribution for our Bayesian game on top of our Q-tree instead of as an integral part of our Q-tree.

[2] The set of all Bayes-Nash equilibria for G_B will be denoted as $\ddot{\boldsymbol{\mathcal{A}}}$.

Draft

nondeterministic stage game. This translates into G_B requiring that support is applied to at least two pure strategies for some problem instance ι_a. It is the existence of this requirement of multiplicity in response to ι_a that enables $\mathcal{M}$ to behave nondeterministically [3] [4].

We know from the nature of a Bayes-Nash equilibrium that this multiplicity in pure strategy responses to ι_α can only exist if each pure strategy response $r_{\alpha,a}$ to ι_a is equivalently optimal to all other pure strategy responses to ι_α. This suggests that we ought to be able to just consolidate our support under one of said pure strategies and prune the newly unsupported pure strategies from π_α. In fact, this is essentially the suggestion made in the second half of McNaughton's 1966 paper [27]. The problem with this suggestion is that we do not know whether or not we can remove the multiplicity in our pure strategy responses to ι_a without eliminating $\mathcal{M}$'s ability to answer $\mathcal{P}$. Specifically, does there exist a problem $\mathcal{P}$ that has an associated answer $\mathcal{A}$ that cannot be produced in a deterministic manner? We will attempt to answer this question with the example of stochastic answers.

Section 4.3 Stochastic Answers

To address the question of whether or not P is a proper subset of NP, we will begin by investigating the problem of finding a stochastic answer; such as a perfectly random set of perfectly random numbers. To clarify our definition of a perfectly random set of perfectly random numbers, a perfectly random set of perfectly random numbers is distinct from an apparently random set of apparently random numbers in that a perfectly random set of perfectly random numbers has an internal randomness and a randomness of occurrence whereas an apparently random set of apparently random numbers will have only a high quality of apparent or measurable internal randomness associated with it, or a high Kolmogorov complexity measure [28], [29].

While this is not in opposition to the requirements of a stochastic answer, we can see from Definition 3 that a high Kolmogorov complexity measure does not on its own qualify a value as stochastic. Specifically, we know from the properties of iterative functions and fractal geometry [30] that a given function f can exactly reproduce an answer $\mathcal{A}$ with a high associated Kolmogorov complexity measure for all instantiations of f. That is, despite having a high Kolmogorov complexity, $\mathcal{A}$ may be identical for all given instantiations $r_{\alpha,a}$ of $\mathcal{M}$. This is in contradiction to Definition 3; which requires that $\mathcal{M}$ is incapable of reproducing $\mathcal{A}$ with a greater likelihood than the likelihood of occurrence for $\mathcal{A}$.

For example, let us say that we create 100 perfectly random three-digit numbers x_i, each with an associated probability of creation of 1/1000 (utilizing the standard base ten digit set with each digit having a probability of 1/10). Let us then say that we randomly draw five numbers x_{53}, x_4, x_{29}, x_{31}, and x_{73}; each draw having an associated probability of 1/100. Such a draw will have an associated probability of occurrence:

$$\text{probability of occurance of } \mathcal{A} = \left(\left(\frac{1}{100}\right)\times\left(\frac{1}{10}\right)^3\right)^5 = \frac{1}{10^{25}}$$

We then completely discard all five numbers from our set of perfectly random numbers, restore our system to the same initial state it was in before we created our initial set of 100 perfectly random numbers, and then we repeat the process of creating a perfectly random set of perfectly random numbers. If the average number of times that we draw the sequence $[x'_{53}, x'_4, x'_{29}, x'_{31}, x'_{73}]$, where $x'_{53} = x_{53}$, $x'_4 = x_4$, $x'_{29} = x_{29}$, $x'_{31} = x_{31}$, and $x'_{73} = x_{73}$, is at all greater than 10^{-25} then our process of generating perfectly random numbers is not a true random number generator. This is true even if the numbers generated have the internal quality of randomness or a high Kolmogorov complexity.

Since a deterministic automaton is in its very nature required to reproduce values in a predictable, non-stochastic fashion, it cannot avoid reproducing the same apparently random set x_{53}, x_4, x_{29}, x_{31}, and x_{73} of

Draft

apparently random numbers each time it is restored to the same initial state. As such, we are required to utilize a stochastic, or nondeterministic, automaton to produce the desired perfectly random set of perfectly random numbers.

Section 4.4 Is P Equivalent to NP?

Looking back at Lemma 2 and Lemma 3 we can see that the question of whether or not P ≈ NP is a question of whether or not, for all problems $\mathcal{P}$, G_B contains a Bayes-Nash equilibrium that provides support to only a single partition $\boldsymbol{\pi}_\alpha$ that is determinizable [8], [17], [18], [27]. This implies that if every problem $\mathcal{P}$ can be solved by a single component deterministic Markov game G_M then P ≈ NP; however, if there exists a problem $\mathcal{P}$ that cannot be solved by a single component deterministic Markov game G_M then P⊂NP.

Since we know that the optimal algorithmic solution to every given problem P is a Bayes-Nash equilibrium $\boldsymbol{\mathcal{A}}$, it is tempting to just say that we can select the deterministic Bayes-Nash equilibrium associated with any given problem $\mathcal{P}$. That is, we know that in order for the pure strategies $r_{\alpha,a1}$, $r_{\alpha,a2}$, …, which are able to be invoked in response to $\boldsymbol{\iota}_a$, to participate in $\boldsymbol{\pi}_\alpha$ said pure strategies $r_{\alpha,a1}$, $r_{\alpha,a2}$, … must be equivalently optimal with respect to $\boldsymbol{\iota}_a$; otherwise, the non-equivalently optimal pure strategies will decrease the optimality of $\boldsymbol{\pi}_\alpha$. This implies that we ought to be able to consolidate our support for $r_{\alpha,a1}$, $r_{\alpha,a2}$, … under a single pure strategy $r_{\alpha,aj}$ and prune the newly unsupported pure strategies ($r_{\alpha,a1}$, $r_{\alpha,a2}$, …)/$r_{\alpha,aj}$ from $\boldsymbol{\pi}_\alpha$: thereby determinizing $\boldsymbol{\pi}_\alpha$. In fact, this is essentially another way of describing McNaughton's method for determinization presented in [27]. The problem with this statement is that it assumes that we can determinize G_B without destroying $\mathcal{M}$'s ability to answer $\mathcal{P}$. More specifically, does there exist a problem $\mathcal{P}$ with an associated answer $\mathcal{A}$ for which we cannot remove the uncertainty built into G_B, through the multiplicity of pure strategy responses $r_{\alpha,a}$ to a given problem instance $\boldsymbol{\iota}_a$, without also removing $\mathcal{M}$'s ability to answer $\mathcal{P}$ in a way that satisfies every element of $\mathcal{P}$? If such a problem exists then said problem $\mathcal{P}$ must require a stochastic answer.

To answer this question we will consider the example of the random set of random numbers, presented in Section 4.3. As discussed in Section 4.3, a perfectly random set of perfectly random numbers cannot be produced by deterministic means. This implies that we must treat G_B as either containing optimal non-deterministic sub-game equilibria G_M or containing Bayes-Nash equilibria $\mathcal{A}$ that applies support to multiple deterministic sub-games G_M.

Lemma 4: No problem $\mathcal{P}$ that is defined over a stochastic answer $\mathcal{A}$ can be solved via non-stochastic, or deterministic, methods.

Proof:

For the proof of Lemma 4 we will assume that $\mathcal{P}$ is solvable and that there exists a generic automaton $\mathcal{M}$ that has the capability of solving $\mathcal{P}$. We know from Lemma 2 and Lemma 3 that we can define $\mathcal{M}$ with respect to a Bayesian game G_B defined over the $\boldsymbol{q}$-tree for $\mathcal{M}$ that can be utilized to solve $\mathcal{P}$; however, we have not shown whether or not G_B must be deterministic or nondeterministic, or if every problem $\mathcal{P}$ must have an associated deterministic Bayes-Nash equilibrium $\boldsymbol{\mathcal{A}}$.

We know from Section 4.3 that for a stochastic answer $\mathcal{A}$ the valuation for $\mathcal{A}$ cannot be identical to previous valuations with a frequency greater than that of the likelihood of occurrence for said valuation of $\mathcal{A}$ (from this point on we will refer to this frequency as the likelihood of generation). The only way that a valuation for $\mathcal{A}$ can vary is if there are multiple possible valuations for $\mathcal{A}$ after the application of $\boldsymbol{\iota}_a$ to G_B; which can only occur if there are multiple possible pure strategy responses for $\boldsymbol{\iota}_a$. Since a unique

Draft

valuation for $\mathcal{A}$ can only come from a unique pure strategy response $r_{\alpha,a}$, we must have at least one pure strategy response $r_{\alpha,aj}$ for each unique valuation for $\mathcal{A}$.

Since the likelihood of generation for any given valuation for $\mathcal{A}$ cannot exceed the likelihood of occurrence we must have at least as many unique values for $\mathcal{A}$ as required to prevent a repetition of any given valuation within a statistical frequency greater than that of the likelihood of occurrence of $\mathcal{A}$. Since each unique valuation must have a corresponding unique pure strategy $r_{\alpha,a}$ that produced said valuation, we must have at least as many pure strategies that can be invoked by M in response to ι_a as required to prevent a repetition of any given valuation at a frequency greater than the likelihood of occurrence of $\mathcal{A}$. This means that if the likelihood of occurrence for any given valuation is $1/x$ then we must have at least x possible valuations for $\mathcal{A}$[3]: and therefore x pure strategies $r_{\alpha,aj}$ that can be invoked in response to ι_a. Since our deterministic automaton is required to have a one-to-one correspondence between problem instances ι_a and pure strategies $r_{\alpha,a}$, as shown in Lemma 1, stochastic answers cannot be generated by deterministic automata.

Let us assume to the contrary that we can solve P via a single deterministic algorithm π_α on a deterministic automaton. This would imply that our single possible pure strategy response $r_{\alpha,a}$ for ι_a is capable of generating at least x possible valuations for a stochastic answer $\mathcal{A}$: where $\mathcal{A}$ has an associated likelihood of occurrence of $1/x$. The ability of $r_{\alpha,a}$ to have more than a single valuation is in contradiction to the definition of our state transition function $\delta: Q \times \Sigma \rightarrow Q$ for deterministic automata. As such, deterministic automata are not capable of generating stochastic answers.

Q.E.D

Note that it is tempting to try and argue that a random input sequence ι_a on a deterministic automaton M is capable of generating a stochastic answer $\mathcal{A}$. In response to this argument, note that all of the arguments presented in the proof of Lemma 4 are independent of the specific input sequence ι_a (including all of the arguments in [2] [3] [4]). The reason for this is that, while external randomness can be utilized by M to produce a pseudorandom answer $\mathcal{A}$, complete knowledge about the external randomness that affected M will always allow us to exactly reproduce $\mathcal{A}$. Essentially, and in consideration of external sources of randomness, we can redefine a stochastic answer as having a likelihood of generation with respect to all possible random input sequences ι_a such that if ι_a was exactly reproduced in series then the likelihood of generation for $\mathcal{A}$ would be less than the likelihood of occurrence of $\mathcal{A}$.

Lemma 4 leads naturally into the following theorem.

Theorem 1: P is a proper subset of NP.

Proof:

Theorem 1 is a natural derivation of Lemma 4. That is, if no problem P defined over a stochastic answer $\mathcal{A}$ can be solved via a deterministic methodology, then there must be a subset of such problems P that exist in NP but not P. However, let us assume to the contrary that P $\approx$ NP $\forall P$. If this is true then either a) any given problem P defined over a stochastic answer $\mathcal{A}$ can be solved via a deterministic partition of G_B or b) there does not exist any such problems P that are defined over a stochastic answer $\mathcal{A}$.

[3] ${}_yC_1 = y \therefore$ if $y < x$ then after no more than y unique valuations we will be required to repeat a value for $\mathcal{A}$; causing our probability of generation for $\mathcal{A}$ to be less than $1/x$.

a) The ability of a given problem $\mathcal{P}$ defined as requiring a stochastic answer $\mathcal{A}$ to be solved by a deterministic partition $\pi_\alpha \in \Pi$ is in contradiction to Lemma 4. As such, either there does not exist any such problem $\mathcal{P}$ or Theorem 1 is correct.
b) The denial of the existence of problems defined over stochastic answers is in contradiction to our example of the perfectly random set of perfectly random numbers. As such, Theorem 1 must be correct.

Q.E.D

Section 5 Conclusion:

In this paper we completed our formalization of the proof proposed in [1]; that is, we completed our proof that P $\subset$ NP. More specifically, we demonstrated that problems requiring stochastic answers belong to NP but not P. Since problems requiring stochastic answers belong to NP/P, P must be a proper subset of NP.

Section 6 Future Research:

We have completed our proof that P $\subset$ NP; however, there still remains the question of whether or not all problems involving non-stochastic answers belong to P or NP. It is our belief that all problems involving non-stochastic answers do in fact belong to P; however, this has yet to be demonstrated. That said, we believe that this can be demonstrated by revisiting McNaughton's 1966 [27] determinization method with the added consideration of the game structure presented here and the Kolmogorov complexity [28], [31] of both the problem instance ι_a and the answer $\mathcal{A}$.

Additionally, there exists a common belief that Savitch's theorem [32] additionally proves that P = NP. With regards to the technical aspects of Savitch's theorem, our group is in complete agreement; however, Savitch's theorem never once considers the existence of random goals. Specifically, Savitch's theorem is indisputably correct in the context of problems with associated non-stochastic answers: but not in the context of problems which require stochastic answers. This oversight of Savitch's theorem manifests itself first in theorem 1 of [32] and finally in theorem 4 of [32]. A formal investigation of the effect of this omission will need to be performed.

For more information on the work presented here please see the author's website at http://www.holcombtechnologies.com/Computational%20Equilibria.aspx.

Works Cited:

[1] J. W. Holcomb, "Stochastic Answers and the Question of Whether or Not PSPACE is a Proper Subset of NPSPACE"; Journal of Applied Global Research, Vol. 4, Issue 9, pp. 1-11, May 2011.
[2] J. W. Holcomb, "Deterministic Automata on Bayesian Games"; unpublished.
[3] J. W. Holcomb, "Requiring Multiplicity for Nondeterministic Automata"; The 2011 International Conference on Foundations of Computer Science, 2011in press.
[4] J. W. Holcomb, "Requiring Multiplicity for Nondeterministic Automata, Part II"; unpublished.
[5] E. W. Weisstein, CRC Concise Encyclopedia of Mathematics, Chapmen & Hall, 1999.

Draft

[6] J. Edmonds, "Paths, Trees, and Flowers"; Canadian Journal of Mathematics, Vol. 17, pp. 449-467, 1965.

[7] A. Pnueli, "The Temporal Logic of Programs"; Proceedings of the 18th Annual Symposium on Foundations of Computer Science, IEEE, 1977, pp. 46-57.

[8] M. O. Rabin, "Decidability of Second-Order Theories and Automata on Infinite Trees"; Bulletin of the American Mathematical Society, Vol. 74, Issue 5, pp. 1025-1029, 1968.

[9] R. Alur, T. A. Henzinger, and O. Kupferman, "Alternating-Time Temporal Logic"; Journal of the ACM, Vol. 49, Issue 5, pp. 672-713, September 2002.

[10] O. Kupferman, M. Y. Vardi, and P. Wolper, "An Automata-Theoretic Approach to Branching-Time Model Checking"; Journal of the ACM, Vol. 47, Issue 2, pp. 312-360, 2000.

[11] A. Pnueli, and R. Rosner, "On the Synthesis of a Reactive Module"; Proceedings f the 16th ACM Symposium on Principles of Programming Languages, ACM, 1989, pp. 179-190.

[12] E. A. Emerson, and C. S. Jutla, "Tree Automata, mu-Calculus and Determinacy"; Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, 1991, pp. 368-377.

[13] M. De Wulf, L. Doyen, N. Maquet, and J. Raskin, "Antichains: Alternative Algorithms for LTL Satisfiability and Model-Checking"; Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, Vol. 4963, 2008, pp. 63-77.

[14] M. De Wulf, L. Doyen, T. A. Henzinger, and J. Raskin, "Antichains: A New Algorithm for Checking Universality of Finite Automata"; Proceedings of the 18th International Conference on Computer Aided Verification, Springer, Vol. 4144, 2006, pp. 17-30.

[15] F. Horn, W. Thomas, and N. Wallmeier, "Optimal Strategy Synthesis in Request-Response Games"; Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis, Springer, Vol. 5311, 2008, pp. 361-373.

[16] K. Chatterjee, T. A. Henzinger, and F. Horn, "Finitary Winning in Omega-Regular Games"; ACM Trans. Comput. Logic, Vol. 11, pp. 257-271, November 2009.

[17] S. Safra, "On the Complexity of Omega-Automata"; Proceedings of the 29th Annual Symposium on Foundations of Computer Science, IEEE, 1988, pp. 319-327.

[18] M. Y. Vardi, "Automatic Verification of Probabilistic Concurrent Finite-State Programs"; Proceedings of the 26th Annual Symposium on Foundations of Computer Science, IEEE, 1985, pp. 327-338.

[19] K. Leyton-Brown, and Y. Shoham, Essentials of Game Theory: a Concise, Multidisciplinary Introduction, Morgan & Claypool Publishers, 2008.

[20] E. Filiot, N. Jin, and J. Raskin, "An Antichain Algorithm for LTL Realizability"; Proceedings of the 21st International Conference on Computer Aided Verification, Springer-Verlag, Vol. 5643, 2009, pp. 263-277.

[21] B. Jobstmann, and R. Bloem, "Optimizations for LTL Synthesis"; Proceedings of the 6th International Conference on Formal Methods in Computer-Aided Design, IEEE Computer Society, 2006, pp. 117-124.

[22] K. Chatterjee, L. Doyen, T. A. Henzinger, and J. Raskin, "Algorithms for Omega-Regular Games with Imperfect Information"; Proceedings of the 20th International Workshop on Computer Science Logic, Springer, Vol. 4207, Issue 3:4, September 2006, pp. 287-302.

[23] T. Agotnes, V. Goranko, and W. Jamroga, "Alternating-Time Temporal Logics with Irrevocable Strategies"; Proceedings of the 11th Confrence on Theoretical Aspects of Rationality and Knowledge (TARK '07), ACM, 2007, pp. 15-24.

[24] K. Chatterjee, T. A. Henzinger, and N. Piterman, "Strategy Logic"; Proceedings of the 18th International Conference on Concurrency Theory, Lisbon, Vol. 4703, 2007, pp. 59-73.

[25] W. Thomas, Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, Vol. B, Elsevier Science Publishers, 1990, pp. 133-191.

[26] T. Kropf, Introduction to Formal Hardware Verification, Springer, 1999.

[27] R. McNaughton, "Testing and Generating Infinite Sequences by a Finite Automaton"; Information and Control, Vol. 9, 1966, pp. 521-530.

Draft

[28] L. Staiger, "The Kolmogorov Complexity of Real Numbers"; Proceedings of the 12th International Symposium on Fundamentals of Computation Theory, Springer, Vol. 1684, 1999, pp. 536-546.

[29] F. Henning, and L. Staiger, "Iterated Function Systems and Control Languages"; Information and Computation, Vol. 168, pp. 125-143, 2001.

[30] K. J. Falconer, Fractal Geometry, Mathematical Foundations and Applications, John Wiley & Sons Ltd., 1990.

[31] M. Zimand, "Possibilities and Impossibilities in Kolmogorov Complexity Extraction"; SIGACT News, Vol. 41, Issue 4, pp. 74 - 94, 2010.

[32] W. J. Savitch, "Relationships Between Nonderterministic and Deterministic Tape Complexities"; Journal of Computer and System Sciences, Vol. 4, pp. 177-192, 1970.

Draft