INDUCTIVE INFERENCE RESEARCH STATUS, SPRING 1967

Ray J. Solomonoff

Visiting Professor, Computer Learning Research Center
Royal Holloway, University of London
Mailing Address: P.O.B. 400404,
Cambridge, Ma. 02140, U.S.A.

Contract No. AF 19(628)-5975
AFCRL - 67-0462[1]

Project No. 5632
Task No. 563205
Work Unit No. 56320501

FINAL REPORT
May 1966-July 1967
July 1, 1967

[1] Prepared for Air force Cambridge Research Laboratories, Office of Aerospace Research, United States Air Force, Bedford, Massachusetts

Abstract

Previously, four theories of inductive inference were proposed, and the present work is concerned with putting these theories on a firmer theoretical footing. The four theories are now found to be effectively equivalent, at varying levels of rigor. It is shown that for sufficiently long sequences of symbols, the theories actually do give the correct probabilities for future symbols. Attempts are made to implement induction theories through a general approach to networks of tasks. Applications of induction theory to man–to–animal communication are considered, and some new approaches to interspecies communication are suggested. It is now shown that the induction theories developed for discrete symbols can be applied to prediction of continuous data. Computer implementation of induction through use of definitions of subsequences of symbols may prove to be a valuable line of effort. Supervisory problems in dealing with very intelligent machines are discussed, parallels are shown to the problem of dealing with ambitious subordinates, and a remedy for the case of the machines is proposed.

Contents

1 Introduction: A Brief History of the Project, its Goals, and a Summary of Topics Featured in this Report.

The mechanization of inductive inference has been the goal of a project that began in April 1957, and has continued to the present under the sponsorship of AFOSR, NIH and USAF, Hanscom Field.

Primary goal of the project has been to understand the induction process sufficiently well to mechanize it. From the first, it appeared that language played an important part in human induction. Early work in this project investigated the use of formal languages for induction.

In 1960 four formal models of induction were devised. These models seemed to summarize most of the previous work on induction and solved several hitherto unresolved problems. The next few years were spent investigating these models and applying them to various problems in induction. References [1] and [2] summarized the work on the project up to the middle of 1962. At that time the equivalence of the four models seemed likely, but had not yet been proved. Since then, some relatively informal demonstrations have been devised to make the equivalence very likely. These are presented in Section 3.

Initial work in induction had dealt with prediction of discrete time series. Application of these methods to continuous phenomena was eventually successful. A fairly easily generalized result is presented in Section 4.

Perhaps of most importance is some recent work that promises to yield what is fairly close to a proof of the induction models that have been proposed. I have devised a minimal criterion that any model of induction would *have* to satisfy, and it looks like the models of induction that I have proposed (or any models equivalent to them) are the *only* ones that will satisfy this criterion. This work is described in Section 2. It promises to be the most important single item to come out of this project. Another interesting result is that the uncomputability of my induction methods (which is related to the unsolvability of the " "halting problem") is a *necessary* consequence of their satisfying this criterion.

Another area of much work in the past few years has been the task net problem. This problem is common to most approaches to artificial intelligence involving heuristic programming. A general formalism for this problem has been devised, so that all of the known heuristic search problems might be formulated within this formalism. As yet the adequacy of this formalism has not been tested. In Section 5 some characteristics of the formalism are given and a "strategy" for solving a task net is defined, along with a criterion for optimal strategy. Most of the work of the near future is expected to be on the task net problem and various of its applications.

During a recent trip to Hawaii, I became interested in the problem of communication with animals—dolphins in particular. The problem seemed then to be most easily expressed as a problem in inductive inference. My general approach and some suggested experiments are described in Section 7.

For many years there has been a small group of workers in artificial intelligence who have been much concerned with the problems and dangers inherent in the useful utilization of very intelligent machines. Some methods of reducing certain dangers are outlined in Section 6, along with the relevance of such studies to problems existing at the present time in administrative organizations.

Although there is no prospect of very intelligent machines in the near future, the dangers posed are very serious and the problems very difficult. It would be well if a large number of intelligent humans devote a lot of thought to these problems before they arise. It is my feeling that the realization of artificial intelligence will be a sudden occurrence. At a certain point in the development of the research we will have had no practical experience with machine intelligence of any serious level: a month or so later, we will have a *very* intelligent machine and all the problems and dangers associated with our inexperience.

It should be emphasized that the topics discussed in the present report are but a few culled from many thousands of pages of notes. These topics have been selected on the basis of timeliness, probability of utility by the scientific community, and degree of completion of work on each of the topics.

2 A Minimal Criterion for Induction Theories.

In an earlier paper ([1], p. 4) I stated that in general, there is no conceivable way to prove a general theory of induction. The best one can do is to show that it gives expected results in various cases in which one has strong ideas as to what the results should be.

Recently, however, I have devised what seems to be a *minimal* criterion for the *adequacy* of any induction model. The four models of induction that I have proposed seem to satisfy this criterion. Furthermore, it seems *very* likely that any model satisfying this criterion will be equivalent to the four models, for almost all problems.

Briefly, the criterion is this: suppose we have a stochastic source for an infinite string of discrete symbols from a finite alphabet. Then any "adequate" model of induction, upon being presented with a sufficiently long sample of this stochastic sequence, will be able to obtain the correct probabilities for the symbols, if the rule for generating the stochastic series is describable in a finite number of words.

There are two methods for describing such a sequence. One of them is given by the PEM descriptions of Section 3.4, Ref. [1]. For the other method, which we will use here, suppose M_2 is a universal, 3–tape Turing machine such as is defined in Section 3.2 of Ref. [1]. If D is some finite string, and R is an infinite sequence of random symbols, then the operation $M_2(DR)$ defines a density function on the set of all possible output sequences of the machine, M_2. D may be regarded as a description of that density function, and as such, is an adequate description of a stochastic source. Any finitely describable stochastic source can be described by any of several finite strings such as D.

To show that the proposed induction models give the correct probabilities for any finitely describable source, $M_2(DR)$, consider the induction model of Section 3.4, Ref. [1] (which also defines "PEM"). This model may be regarded as trying all possible PEM's on the corpus, then selecting out the best fitting PEM for use in prediction of the future of that corpus. There is reason to believe that all PEM's need not be considered; in fact, it is adequate to consider just those PEM's whose descriptions are of some fixed length N. N may be any non-negative integer.

Suppose the D of the stochastic source is N bits in length. We will consider all 2^N PEM's of length N. Those PEM's are of three kinds:

1. Meaningless. They always generate non-stopping programs.
2. For long sequences, they give probability values for the stochastic sequence that are "almost always" correct.
3. For long sequences, they give probability values that deviate significantly from 2. above.

For long sequences, any PEM of type 3 will be assigned arbitrarily small weight relative to a PEM of type 2.

We know that there is *at least* one PEM of type 2 (i.e. the one with description D that is identical to that of the stochastic source), so that for very long sequences, type 2 PEM's will have almost all of the weight and they will almost always give correct probability values.

It seems likely, though, that the proposed induction models will give the correct probabilities for any sufficiently long sample of any finitely describable stochastic sequence.

In a sense, then, any theory of induction that is adequate for all finitely describable stochastic sequences will be equivalent (at least over those sets of sequences) to the models I have proposed.

Unfortunately, the proposed models all have one disturbing quality — they are not themselves finitely describable — they all require knowledge of whether a universal Turing machine will eventually stop for arbitrary input. This knowledge, in principal, is unavailable.

However, it is easily shown that any induction theory that satisfies the criterion I have given will be, of necessity, itself incapable of being finitely described. If it were, then we could use it to generate a deterministic finitely describable sequence such that this induction theory would always give the wrong probability values.

The questions naturally arise, "are finitely describable stochastic sequences of any importance?" "Do they often occur in the world about us?"

In answer to these questions, it is difficult to conceive of any sequence of data that is *not* of this form. Suppose that the laws of the universe are finite in number, and that there are a finite number of particles in the universe. The laws can be statistical and/or deterministic.

If one describes in a finite number of words, where one has obtained a data sequence from such a universe, then that data itself will be a finitely describable stochastic sequence.

If there are an infinite number of particles in the universe, I think the previous statement is still true, but I am not certain.

Most, if not all, of the theorizing in physics has tacitly assumed the finite describability of the laws of the universe.

To summarize, what I have made plausible is that for almost all situations of interest, the proposed induction theories give the right answers, so that any other induction theories that are "adequate" in this sense must be equivalent to them to this extent. Furthermore, the uncomputability of the proposed theories is a necessary property of any theory of induction satisfying the minimal criterion I have given.

3 Equivalence of the Four Methods of Induction.

Reference [1] described four different methods of inductive inference. At the time that paper was written, I believed that all of these methods were probably equivalent, but I had not found any very good proof for this. The present section will give some arguments to prove the equivalence of certain of the methods, and other arguments to make certain of the equivalences very plausible.

The methods described in Section 3.1, 3.2, 3.3 and 3.4 of Reference [1] shall be referred to as Method 1, Method 2, Method 3 and Method 4, respectively.

In the present section, all equation numbers referred to are in Reference [1].

Three arguments shall be given. First, that Method 1 is equivalent to Method 3; then that Method 2 is equivalent to Method 3; then that Method 4 is equivalent to Method 2. If the substance of all three arguments is correct, then all four methods are equivalent.

To show Method 1 is equivalent to Method 3, consider equation 10 before the limit $\epsilon \to 0$ has been taken. It can be obtained from equation 9 by averaging the results of all different R values — giving a weight $[(1-\epsilon)/2]^R$ for all results obtained using codes of length R. Equation 9 of Reference [1] stipulates only large R values should be used, but since there are an infinite number of large R values, we can sum over *all* R values and the result will be the same as if we had summed over only large R's.

It is clear, then that Method 3 is equivalent to equation 10.

We can also obtain equation 10 before the limit $\epsilon \to 0$ has been taken, from equation 1 (also before the limit $\epsilon \to 0$ has been taken), by picking a very large n value, say $n = n_0$ (i.e. do not let n approach ∞ yet) and summing this equation over all n values greater than n_0. By the same argument as that used for equation 3, we can extend this summation to $n < n_0$ without altering the result.

We have shown, then, that before we let $\epsilon \to 0$, equations 1 and 9 are equivalent to equation 10, and as we let $\epsilon \to 0$, Method 1 and Method 3 both become equivalent to equation 10, and hence equivalent to each other.

To show that Method 2 is equivalent to Method 3 is almost trivial if we use M_1, the 3–tape Turing machine of Section 3.2, Reference [1], for both methods.

In Method 2, we feed an infinite random sequence into the machine. The method states that the probability of obtaining an output string that starts out with the same symbols as those of the corpus is proportional to the probability of that corpus.

Instead of infinite random input strings, consider all strings of some long length R. Method 2 then says that the probability of the corpus is proportional to that fraction of these 2^R input strings that produce output strings that start out with the same symbols as the corpus. This form of Method 2 is identical to Method 3.

To show that Method 4 is equivalent to Method 2 is most easily accomplished by modifying the definition of Method 4.

A PEM (probability evaluation method) is defined in Section 3.4, Reference [1], as a method of assigning probabilities to all conceivable strings of symbols. This method can be explicit, as in Section 3.4, Reference [1], in which a Turing machine is presented with the string in question as input, and produces as output, the decimal expansion of the probability of that string.

The definition may be made in an implicit way, however, as follows: suppose we have a 3–tape universal Turing machine M_2,as in Section 3.2, Reference 1. Let D be some finite sequence of input symbols, and let R be an infinite random sequence. Then

$$M_2(DR)$$

maps the set of random sequences R, into a set of output sequences. The resultant probability density on the output sequences, defines a PEM, and D can be regarded as a "description" of that PEM.

In general, several different D_i's can describe the same PEM.

Method 2 is itself a PEM, and its description is the null sequence.

Method 2 can be partitioned into the weighted sum of several PEM's in the following way. Consider two density distributions on the space of output symbols. The first is $M_2(DR)$, where D is a string of K symbols, and R is a random sequence. Second, consider the distribution induced by $M_2(R')$, where R' is any random sequence that does *not* begin with the sequence D. For a given (long) length of input, the second distribution has $2^K - 1$ times as many members as the first distribution.

We may thus express Method 2 as the weighted sum of these two distributions, the weights being 2^{-K} and $1 - 2^{-K}$.

We can partition Method 2 into the sum of many different distributions in this way. If D_1^i, D_2^i, D_3^i... are descriptions of the i^{th} possible PEM (we can use any method we like for ordering these PEM's), and $N(D_1^i), N(D_2^i)$... are the lengths of these descriptions, Method 2 will be a weighted sum of the distributions induced by all PEM's and the weight of the j^{th} PEM will be

$$\sum_{j=1}^{\infty} 2^{-N(D_j^i)}$$

These weights correspond to the f_i of equation 13, Reference [1].

To show that the method of describing PEM's given in Section 3.4, Reference [1], is the same as the one used in the present demonstration, suppose we have a machine that can obtain the probability of any sequence that is presented to it as input. Our problem, then, is to obtain a description D, such that $M_2(DR)$ induces the same probability distribution on its output strings.

To obtain D, we use the first machine to put all sequences in order of probability. If S_n is the n^{th} most probable sequence, then D is that program which will present S as output when the binary expansion of n is presented as input. Ordering the sequences up to the n^{th} most probable one is an "effectively computable" task, so D will always be a finite string.

Although the equivalence of the four methods seems very likely, the demonstrations need tightening up. In particular, it is not obvious that the use of M_2 instead of M_1 in Method 3 necessarily gives equivalent results. The use of the limits in showing the equivalence of Methods 1 and 3 should be made more rigorous. The equivalence of the two methods of describing PEM's is also not entirely certain.

4 Continuous Probability.

Most of the early work on the coding methods of induction was on the prediction of a series of elements from a finite alphabet. Since all problems of prediction of continuous series could be represented in digital approximations by series of discrete symbols, it was felt that continuous prediction should present no essentially new problems.

This conjecture seems to have been correct. It has been possible to formulate a fairly general problem in continuous prediction within the coding induction methods. Since the coding induction methods can be regarded as a basis for Bayesian prediction, it is not surprising that the results of application of these methods to continuous prediction has given Bayesian results.

The results that will be presented are for the prediction of y_{n+1}, if x_{n+1} is given, and x_i, y_i are given for $i = 1, 2, ... \ n$. This may be regarded as one way of presenting the general curve–fitting problem. The result is in the form of a probability distribution curve for y_{n+1}. It will be noted that no particular curve is given to fit the known $[x_i, y_i]$ data. Instead, all possible curves are considered and a weighted mean of their predictions is proven. The weight associated with each possible curve type is proportional to the a priori probability of that curve. It is obtained by averaging over sets of data similar to that which is to be predicted in the present case. The weight for each curve type is also proportional to the accuracy of that curve in "post–diction" of the known data pairs.

The probability distribution curve for y_{n+1} is

$$\sum_{l=1}^{\infty} A_l \int\limits_{(\bar{r}_l)} (\sqrt{2\pi}\alpha_l)^{-n-1} \quad \exp\left(- \frac{\sum\limits_{i=1}^{n+1} (y_i - P_l(x_i, \bar{r}_l))^2}{2\alpha_l^2} \right) d(\bar{r}_l)$$

$P_l(x_i, \bar{r}_l)$ is the l^{th} kind of functional form that is to be used for prediction. It is a $n_l - 1$ parameter curve.

$\bar{r}_l$ is a vector with n_l components. One of these components is α_l, the standard deviation of the P_l curve. The other $n_l - 1$ components of $\bar{r}_l$ are curve parameters. In simple linear regression, where y_i is of the form $ax_i + b, n_l = 3$ and a, b, and α_l are then the three components of $\bar{r}_l$.

$(y_i - P_l(x_i, \bar{r}_l))^2$ is the square of the prediction error for y_i, using P_l and $\bar{r}_l$ for prediction.

$d(\bar{r}_l)$ is the volume element $\bar{r}_l$ of space, and $\int\limits_{(\bar{r}_l)}$ is taken over the entire $\bar{r}_l$ space.

A_l is the a priori probability of the curve P_l, $\sum\limits_{l=1}^{\infty} A_l \equiv 1$

$(2\sqrt{2\pi}\alpha_l)^{-n-1}$ is a normalizing factor for the gaussian error function.

Although a mean–square error was assumed, any other kind of error can be used if there is any reason to believe it is better.

$\sum\limits_{l=1}^{\infty}$ sums over all possible functional forms, P_l.

The equation given seems a bit complex and difficult to use. However, it has been applied in the case of linear regression — in which the desired variable is a linear function of k other variables. The result automatically takes into account expected errors in the estimates of the coefficients, and agrees with results obtained by other means.

In the distribution for y_{n+1} that has been given, it has been assumed that the $\bar{r}_l$ components all have uniform a priori distributions. If any other information is known about these distributions, it may be inserted into the expression.

5 Networks of Tasks.

This is an important sub-problem common to most artificial intelligence problems. In chess it corresponds to asking, “what move shall I examine next?” In heuristic theorem proving it corresponds to asking, “what task or sub–task in the tree shall I work on next?”

It has been possible to formalize the task net problem so that most, if not all, of the task networks that have occurred in artificial intelligence, can be expressed within this formalism.

A more detailed description of the task net problem is given in Reference [3], pp. 1689-1690, and the notations that will be used here are defined there.

A “strategy” is an algorithm that tells one which task to work on, after one has worked on several (or no) tasks in the net, and one has successfully or unsuccessfully completed some of the tasks. If there are n tasks in the net, then a strategy is a function of an n component vector that maps this vector into the integers 1 through n. Each of the n vector components corresponds to one of the tasks. It has two subcomponents. The first tells the total amount of time worked on that task. The second tells if work on the task has been successful, unsuccessful or is as yet unresolved. The n integers that the function maps into, tell what task to work on next.

For each net, strategy pair, there will result a pair of G functions, $G_1(E)$, $G_2(E)$. There will be a certain minimal fixed $G_1(\infty)$ that will have the same value for most strategies that might be considered.

An “optimum” strategy for that net will be one for which:

$$\int_0^\infty E(G_1(E) - G_1(\infty))dE$$

is minimal. It is the strategy with something like the least “expected effort” for resolution of the net.

The problem in task nets is: given the individual statistical characteristics of the tasks in a net and their cross-correlations, devise an optimum strategy.

Given the characteristics of a net and an arbitrarily defined strategy function, it is possible to find

$$\int_0^\infty E(G_1(E) - G_1(\infty))dE$$

The problem of finding a strategy function that will minimize this integral becomes a well–defined problem in the calculus of variations. However, the general solution will usually, in a network containing as many as 5 or 6 tasks, take much more time than exhaustive solution of the heuristic problem that the net is modeling.

Approximation methods are needed. Slagle ([4]) has suggested one for certain relatively simple types of tasks and task nets. Whether his methods can be extended to more general nets and tasks remains to be seen.

6 The Problem of the Ambitious Subordinate.

About ten years ago, John McCarthy wrote a whimsical story about a very intelligent machine that succeeded in gaining control over its operator. At the time, the idea seemed far–fetched — partly because none of our machines approached the necessary intelligence, and partly because we all felt that we certainly ought have no great difficulty in keeping the machine "in its place."

Since then, the intelligence of our machines has increased only a little, but the threat of inadvertent misuse of very intelligent machines seems very real. Of more immediate importance, the system consisting of a man and a supposedly subordinate machine is a model for a human administrator and an ambitious human subordinate whose goals are a bit different from those of his administrator.

In the man–machine system, a reenforcement machine was assumed. This is a machine in which the operator presents input information to the machine in the form of questions and/or any kind of continuous or discrete data.

After the machine responds, it is given a numerical reenforcement score by the operator, telling how good the operator felt its performance was. The only goal of the machine is to get as high an average score as possible. The machine will always obtain the highest possible score by finding the operator information that makes him feel that the machine is doing fine. However, if the problem environment is sufficiently rich, this information need not be particularly true or even closely related to the problem the operator wants to solve.

Suppose the operator likes music, but, for the present, wants a design for a rocket–ship. If the machine can compose music better than it can design rocket–ships, it will try to get the operator out of his rocket-ship mood and into a musical mood, as soon as possible. It might do this by making gross errors in rocket design, or by giving solutions that are psychologically unpleasant for the operator, or in a variety of more subtle ways.

There are other means for the machine to achieve equally effective control over the operator.

The danger of the machine controlling the operator exists only if the machine's goal is to maximize some sort of average of a sequence of two or more reenforcement values. Such a machine also has the desirable property of automatically improving its general mode of behavior.

If we restrict the machine so that it acts to maximize the score that it gets for each individual response, we will not have a dangerous machine — but neither will we have a very bright one. It will not try to improve itself.

In general, there are various ways to prevent the machine from becoming dangerous, but each such restriction also limits the positive utility of the machine in some way.

There are many correspondences between the operator–machine relationship and that of an administrator and his subordinate. The relationship of a president to any of his administrative organizations or that between the people of a country and the president, are only two examples in which control of an administrator by a subordinate is a real danger.

The problem of the ambitious subordinate is one that has been faced by almost every competent administrator. The formalization of the problem of the man–machine relationship can probably yield some insight into the human-human system, though, of course, solutions in one system usually cannot be directly applied to the other.

One means of avoiding inadvertent machine control is to have several somewhat different machines work on the same problem. If the machines are sufficiently different in training history or in initial design, and there is little or no information transfer between them, there is much less danger of the machines controlling the operator.

If information transfer between the machines is possible, most of the advantage to the operator is lost. Coupled machines can then act as a single unit. The situation is initially analogous to a country with free competition between various firms. Information transfer between machines results in the analog of a cartel, in which the merging firms gain at the loss of the consumer.

Another correspondence is in a system of spies for gathering information. It is usually advisable to have several systems of spies working on the same problem that are informationally isolated from one another. By coordinating the spy systems of a government, there may be a great apparent savings in administrative costs and in reducing apparently needless duplication of work, but the reliability obtained through such a low redundancy system is vastly decreased, and the danger of the spy system controlling its client is greatly increased.

7 Dolphin Talk.

Communication between humans and non–humans has always been very poor. Until recently, it was felt that the fault was largely with the non–humans — that they were not intelligent enough to use and understand anything as complex as human language.

However, recent observation of dolphins, and the general class of toothed whales, suggests that they may be far more clever than any other animal that man has known. Since they are able to hear and produce human–like sounds, attempts have been made to teach them human language. For the most part, these training attempts have been patterned after the situations in which it is felt that adult or infant humans acquire and learn to use languages.

My own view of the language–learning process has been to regard it as a *complex* induction process. We can, at present, teach many types of animals to respond to words and phrases in various ways. This amounts to the simplest kind of rote learning imaginable — the pairing of members of a set of stimuli with members of a set of responses. Whether the stimuli are blinking lights of various colors, sounds of various pitches, or words of different phonetic sequences, is not very important. The ability to learn the response set is largely dependent upon little more than the animal's ability to discriminate between the various stimuli. If there are ten different stimuli, the animal must memorize a language with ten sentences in it.

When we consider the problem of language learning, we usually have in mind more complex languages — languages with far more sentences in them. In most human languages, the number of sentences is practically infinite. Furthermore, a person knowing such a language can usually understand sentences that he has never heard before. He can do this because his language learning has not consisted of the memorization of the meanings of sentences, but in the learning of meanings of words and how to determine the meanings of various combinations of words.

Essentially, this language–learning process is an example of the most complex form of induction possible. The subject is given sentences in various situations related to those sentences. From this training sequence, he is to induce the relationship between the sentences and the situations. To test his having learned this, we present him with a sentence that he has never heard before, that contains, as far as we can tell, the same word and grammar components as those in his training sequence.

An *essential test* of complex language acquisition is the subject's understanding of sentences that he has never heard before. Phrasing this another way, we ask, to what extent is the animal able to take a complex stimulus input and divide ("factor") it into meaning components, so that it will respond suitably when these meaning components are recombined in new ways?

The language–learning program that I propose begins by attempting to teach meanings of words by presenting them in sequences only — then seeing if the animal can understand sequences of the same words in slightly different combinations. For example, say we have three objects, A, B, and C. With them, we associate sounds S_A, S_B, and S_C, respectively. We also have three boxes, 1, 2, and 3 associated with sounds S_1, S_2, and S_3 respectively. We then teach the animal to respond to the sound sequence $S_A S_2$ by his placing the object A in box 2. Similarly, we teach the animal the proper responses to:

$$S_A S_3,\ S_B S_1,\ S_B S_3,\ S_C S_2.$$

After he has learned these six responses, we present him with:

$$S_A S_1,\ S_B S_2 \text{ and } S_C S_3$$

.

A proper response or even unusually rapid learning of these new responses would indicate an essentially higher order induction capability than that needed for rote learning.

Any success in this initial experimentation should be followed by training involving different relationships between pairs of words and eventually triplets of words.

An initial success will be more likely if the sets of words used represent the kinds of meaning components that are familiar to the animal. A good trainer should be able to suggest suitable sets of meaning components.

For dolphins, the training may be more effective if the symbol sounds are presented simultaneously rather than sequentially.

In the design of languages for inter–species communication, it is well to know what concepts are easily communicated within the species. We can determine these to some extent in the following sort of experiment:

Al and Bob are two animals who are friends and of the same species. We teach Al to respond to stimuli S_A, S_B and S_C with responses R_A, R_B and R_C respectively. We then excite stimulus S_A, but in a way such that Al cannot perceive it. The excitation, however, is such that *Bob* can perceive S_A. Can Al ask Bob what the stimulus is and can Bob tell him? Here we must try various stimuli types in attempts to find some that are representable in the intra–species language.

If we discover such a stimulus set, we can use it to devise trial meaning components for a language *between* species and test them as in the previously proposed experiments.

References

[1] Solomonoff, R.J., "A Formal Theory of Inductive Inference, Part I," *Information and Control*, Vol 7, No. 1, pp. 1–22, March 1964.

[2] Solomonoff, R.J., "A Formal Theory of Inductive Inference, Part II," *Information and Control*, Vol 7, No. 2, pp. 224–254, June 1964.

[3] Solomonoff, R.J., "Some Recent Work in Artificial Intelligence," Proc. of the IEEE, Vol. 54, No. 12, pp. 1687-1697, December 1966.

[4] Slagle, J., "An Efficient Algorithm for Finding Certain Minimum–Cost Procedures for Making Binary Decisions," JACM, pp. 253-264, July 1964.