On the Computational Measurement of Intelligence Factors

José Hernández-Orallo
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain.
E-mail: jorallo@dsic.upv.es

ABSTRACT

In this paper we develop a computational framework for the measurement of different factors or abilities which are usually found in intelligent behaviours. For this, we first develop a scale for measuring the complexity of an instance of a problem, depending on the descriptional complexity (Levin *LT* variant) of the 'explanation' of the answer to the problem. We centre on the establishment of either deductive and inductive abilities, and we show that their evaluation settings are special cases of the general framework. Some classical dependencies between them are shown and a way to separate these dependencies is developed. Finally, some variants of the previous factors and other possible ones to be taken into account are investigated. In the end, the application of these measurements for the evaluation of AI progress is discussed.

KEYWORDS: *Measurement of Intelligence, Problem Complexity, Descriptional Complexity, Intelligence Factors.*

1 INTRODUCTION

Are AI systems of today more intelligent than those of 40 years ago? Probably the answer is a clear yes, at least for some of the current systems. However, another different question is 'How much more intelligent?', and, even more, in which aspects are they more intelligent?

In this paper we investigate a framework for the evaluation of such a progress in different factors, extending in a natural way the work endeavoured in [13] and [11], which was specific developed for only some inductive factors. The main aim of the extension should be to develop the less number of factors as possible, by proposing general factors instead of specific ones. Moreover, the framework would make it possible to study their theoretical correlations, and reducing, when possible, a factor to another. This leads finally to a group of tests that can be adapted and implemented for measuring different abilities of intelligent systems.

First of all, we must ascertain three issues for any evaluation of the ability of solving a problem: to give a general scale of a complexity of the problem, to settle the unquestionability of the solution to the problem and to establish a way to know whether the subject has arrived to the solution.

Computational complexity scales problems according to the time different kinds of machines require to solve them in the general case by using the optimal algorithm possible. However, most problems of interest in AI are NP-complete. But, remarkably, some instances of NP-complete problems are easier than instances of polynomial problems. This assertion seems to be contradictory, since any instance has an algorithm to solve that instance in linear or even constant time (the program "if the input is x print the solution y"), so there is apparently no reason for stating that an instance can be easier than another. This has been shown to be false up to an extent, because for some problems it is better (shorter) to give a more general solution than the specific solution for an instance of the problem. This has been formalised under the notion of "instance complexity" (see e.g. [17]), which gives the shortest solution to an instance of a problem provided it does not give a contradictory solution for other instances of the same problem.

However, instance complexity is only of interest for large instances of a considerable descriptional complexity (or for sets of instances). Moreover, the difficulty of the problem is not usually related to the descriptional complexity of the solution. For instance, the descriptional complexity of the answers given by a theorem prover (an accepter) are very short, namely one bit to say 'yes' or 'no'. In the same way, the hardness of a prediction problem cannot be measured by the descriptional complexity of the element predicted, but rather by the complexity of the reason why the element has been predicted. The idea is then to measure the descriptional complexity of the 'justification' or 'explanation' of the solution. Consequently, any cognitive skill can be measured within this framework provided that problem and solution can be computationally formalised.

The paper is organised as follows. After Section 2, where some notation is introduced, Section 3 gives a general formula of the hardness of the instance of a problem, by clarifying how to generalise the concept of 'explanation' of a solution to a problem. Section 4 addresses the issue of specialising it for deductive abilities and discusses their measurement. Section 5 does the same thing for inductive abilities, but recognising that it is necessary to solve the unquestionability problem. Section 6 deals with their dependencies and the possibility of taking other factors into account. Section 7 discusses the applications of these measurements, especially for the evaluation of automated reasoning and machine learning systems. Section 8 closes the paper with the results and open problems.

2 PRELIMINARIES

Let us choose any finite alphabet Σ composed of symbols (if not specified, $\Sigma = \{0,1\}$). A string or object is any element from Σ^*, with $\circ$ being the composition operator, usually omitted. By $\langle a,b\rangle$ we denote a standard recursive bijective encoding of a and b, such that there is a one-to-one correspondence between $\langle a,b\rangle$ and each pair (a,b). Note that this usually takes more bits than $a\circ b$. The empty string is denoted by ε. The term $l(x)$ denotes the length or size of x in bits and $\log n$ will always denote the binary logarithm of n.

The complexity of an object can be measured in many ways, one of them being its degree of randomness [15], which turns out to be equal to its shortest description. Descriptional Complexity, Algorithmic Information or Kolmogorov Complexity was independently introduced by Solomonoff, Kolmogorov and Chaitin to formalise this idea, and it has been gradually recognised as a key issue in statistics, computer science, AI and cognitive science [17][6].

The Kolmogorov Complexity of an object, defined as the shortest description for it, usually denoted by C (plain complexity) or K (prefix-free complexity) turns out to be not computable in general, due to the halting problem. One solution for this is to incorporate time in the definition of Kolmogorov Complexity. The most appropriate way to weight space and time execution of a program, the formula $LT_\beta(p_x) = l(p_x) + \log \tau_\beta(p_x)$, where τ is the number of steps the machine β has taken until x is printed by p_y, was introduced by Levin in the seventies (see e.g. [16]). Intuitively, every algorithm must invest some effort either in time or demanding/essaying new information, in a relation which approximates the function LT. The corresponding complexity, denoted by Kt (see e.g. [17]) is a very practical alternative to K.

3 PROBLEM COMPLEXITY AS EXPLANATORY COMPLEXITY

Consider a problem instance π as a tuple $\langle S,C,I,A,\phi\rangle$ where S is the context or working system where the problem can be established, C is a Boolean function which represents a (syntactical) validity criterion, I is the presentation of the instance, A is the answer and ϕ is a (semantical) verifier[1]. The general problem is denoted by $\pi(\cdot)$ as the tuple $\langle S,C,\phi\rangle$.

We say that E is an explanation for the problem instance π iff E is valid, i.e. $C(\langle S,I,E\rangle) = true$, and E is a means to obtain the solution, i.e., $\phi(\langle S,I,E\rangle) = A$.

From here, it is easy to adapt the definition of Kt to measure the hardness of a problem. Namely, the hardness of a problem instance $\pi\langle S,C,I,A,\phi\rangle$ is then defined as:

$$H(\pi) = \min\{LT(E|\langle S,C,I\rangle) : E \text{ is an explanation for } \pi\} \quad (1)$$

[1] Both C and ϕ could be joined in one function. We have preferred to separate them, because later it will be useful to distinguish between both parts of a correct solution, in order to establish purer factors.

For instance, the hardness of a search problem is usually estimated by the size of the search space. If the search problem is complex, it is necessary to say which branches have been selected in order to arrive to the solution, or either a long time is necessary to explore (and make backtracking) to the misleading ones. It is the function LT which finds a compromise between the information which is needed to guide the search and the logarithm of the time that is also needed to essay all the branches. On the other hand, if the search problem is linear (one possible branch), it is very easier to describe the problem (just follow the rules in the only possible way). However, for very long derivations, the inclusion of time can make hardness high too.

For the evaluation of a subject's ability of solving a kind of problem $\pi(\cdot)$ it is necessary to generate a set of instances of that problem of different hardness. In order to scale the instances more properly, we introduce the concept of k-solvability. An instance of a problem $\pi = \langle S,C,I,A,\phi\rangle$ is *k-solvable* iff k is the least positive integer number such that:

$$H(\pi) \leq k \cdot \log l(I) \quad (2)$$

The use of $\log l(I)$ is justified by the fact that, once the general problem is known, each instance must be 'read' an this takes at least $l(I)$ steps.

Once given a general scale of a complexity of any problem, it is then easy to make a *test* from the previous definition, provided that the unquestionability of the solution to the problem is clear. Unquestionability can only be addressed depending on the kind of problem. We will see this for deductive abilities and especially for inductive abilities in the following sections. Finally, there is no way to know whether the subject has arrived to the solution if the explanation is not given (and usually the explanation is difficult to check or the subject may not be able to express the explanation in a comprehensible form). For instance, the subject may have given the right solution but maybe due to wrong derivations. Fortunately, in the case of multiple solutions, this situation will be discardable in the global reckoning of the test. In the case of few solutions, such as 'yes'/'no', it is then necessary to penalise the errors by using some formula that takes into account the possibility of guessing the right answer 'by mistake'.

Another question is the time limit for making the test. This may highly depend on the factor to be measured, and whether there is a special interest in evaluating the ability to solve a given problem or the ability to solve it quickly. The selection of the time limit and the evaluation of the score according to it could be very interesting for evaluating resource-bounded rational systems.

Finally, we have not considered the possibility of multiple correct explanations for the same solution, which would suggest a modification of (1). Consider the situation of the best explanation with $LT = n$, but several other explanations of $LT = n+1$. Intuitively, the existence of these other explanations also affects the easiness of the solution. However, this is very difficult to evaluate in practice be-

cause there are always infinite slight variations of the best explanation (void steps, redundancies, etc.), so the previous situation is extremely frequent (if avoidable). It is then assumed that for every k:

$$card\{ \quad E : LT(E) = k \text{ and } C(\langle S, I, E\rangle) = true \text{ and } \phi(\langle S, I, E\rangle) = A_i \quad \} << card\{ \quad E : LT(E) = k \text{ and } C(\langle S, I, E\rangle) = true \quad \} \qquad (3)$$

In other words, we assume that the proportion of valid and correct explanations wrt. valid explanations is very small.

Once a general framework is established, let us study which deductive and inductive abilities are feasible and interesting to be measured within it.

4 DEDUCTIVE ABILITIES

Apparently, deductive abilities are much easier to measure, because there is no possible subjectivity about the correct answer; given the premises and the way to operate with them, only one answer is possible.

An instance of a deductive problem $\pi = \langle S, C, I, A, \phi\rangle$ can be defined in terms of the previous framework in the following way: S corresponds to the set of axioms or axiomatic system, C is a Boolean function which says what is a valid application of the axioms[2], I is the instance of the deductive problem, A the answer and ϕ is a verifier, i.e., $\phi(\langle S, I, E\rangle) = A$, in this case, a verifier that checks whether A is a result of applying a solution E to I in S.

In this case the explanation E is represented by a *proof* in S stating that A_i is a the result of I or, in other words, a derivation from I to A_i.

Example: Consider for instance an accepter that tells whether a proposition is a theorem or not. Let S be the axioms of arithmetic. Let C a function that tells that a derivation is valid according to the rules of application of the axioms, and let I be the instance "Is Fermat's famous conjecture true?" (recently a theorem). Which is the hardness of the solution $A =$ 'yes'? The descriptional complexity of A (which is just yes) would say that the instance is very easy, however its hardness given by H turns out to be the LT of the proof with less LT.

Example: Consider instead the instance "solve 2+3" which, also with a low complexity of $A = 5$, turns out to be simple, because the derivation is easily describable from $\langle S, C, I\rangle$. In general, any calculation is shortly describable, so its hardness will depend solely on its temporal cost.

According to this example, we can distinguish some classical deductive problems that can be measured. In particular, the following factors are distinguished:

- Calculus Ability: in this special case, C only allows a specific and deterministic application of the rules or axioms of S. In this case the search space is linear. As it has been said before, its complexity is exclusively given by the logarithm of the time which is needed from the input I to the output A. This ability is not of much interest to be measured nowadays, since it is better done by computers than humans, and it would finally measure the computational power of the subject / machine.
- Derivational Ability: in this case, C only allows a varied application of the rules or axioms of S. Consequently, the search space is open. The complexity is then given by a compromise between the logarithm of the time which is needed to know that a branch leads to no solution, and some information that may say which branches to take (and which ones not to take).
- Accepter Ability (proving ability): It is a special case of the previous ability, with the special feature that A can only be 'yes' or 'no'. Theoretically, there is no reason for expecting that a subject may have a different result in this problem that in the previous one.

The way to implement a concrete test for the previous ability is not complicated. For calculus ability, it is just necessary to generate some derivations. Their length will determine the time which is needed to follow them. On the contrary, for the other two abilities, it is necessary to generate a possible derivation, and look that there are no shorter equivalent derivations. This, in general, will be extremely costly, growing exponentially according to the value of k-solvability. Fortunately, there is no need for efficiency here. A hard test can be generated during days, even weeks, and then passed to several subjects.

5 INDUCTIVE ABILITIES

A sequential inductive problem $\pi = \langle S, C, I, A, \phi\rangle$ can also be defined in terms of the previous framework in the following way: S corresponds to the background knowledge, I is a sequential evidence (with $l(I) = n$), C is a Boolean function which represents the hypothesis selection criterion (e.g. is it the shortest one (or equivalent to it)? Is it the one with less LT?), A is the prediction of the $(n+1)$th element of the sequence and ϕ is a verifier, i.e., $\phi(\langle S, I, E\rangle) = A$, in this case, a verifier that checks whether A is the $(n+1)$th element given by the hypothesis with the background knowledge S and also checks whether both cover I.

In this case the explanation E is represented by a 'hypothesis' wrt. S that affirms that A is 'what follows' I or, in other words, a prediction from I.

Example: Consider for instance a prediction problem. Let S be a background knowledge, containing, among other things, the order of the Latin alphabet. Let C be a function that tells that a hypothesis is the best (or equivalent) according to a selection criterion, and let I the instance "`aaabbbcccdddeeefffgggh`". Which is the hardness of the solution $A =$ '`h`'? The descriptional complexity (in LT terms) of the hypothesis is again what is taken into account.

The main question of evaluation of induction is that of inquestionability. Even if the selection criterion is given,

[2] S and C are usually seen jointly, but its separation will allow a thinner factorisation later on.

two plausible explanations may differ slightly, and the selection criterion would give that one is slightly better than the other, but this would depend highly on the descriptional mechanism used. In [13] and [11] this difficult problem is addressed, according to a comprehensive criterion, a variant of the simplicity criterion based on Kolmogorov Complexity in the style of Solomonoff [20], but ensuring that the data is covered comprehensively, i.e. without exceptions. Accordingly, the *simplest explanatory description*, denoted by $SED(x|y)$, is defined in [11] as the simplest (in LT terms) description which is comprehensive wrt. the data x given the background knowledge y. To ensure unquestionability, the examples are selected such that there are no alternative descriptions of similar complexity that give a different description. Finally, there is a small possibility that a good prediction is given by a 'wrong' explanation. This probability may be neglected in the tests or corrected by a penalising factor in the score of wrong results.

From here, partially independent factors can be measured by using extensions of the previous framework. For instance, inductive abilities, such as sequential prediction ability, knowledge applicability, contextualisation and knowledge construction ability can be measured in the following way:

- Sequential Prediction Ability: several unquestionable sequences of different k-solvability are generated. A test for this ability has been generated in [13] and passed to humans, jointly with a typical psychometrical test of intelligence. The correlation between the results of both tests showed that this is one of the fundamental factors of intelligence, which deserves more experimentation to be done.
- Inductive Knowledge Applicability (or 'crystallized intelligence'): a background knowledge B and a set of unquestionable (with or without B, denoted by $H(x_i|B)$ and $H(x_i)$ respectively) sequences x_i are provided such that $H(x_i|B) = H(x_i) - u$ but still $SED(x_i|B) = SED(x_i)$. The difference of performance between cases with B and without B is recorded. This test would actually measure the application of the background knowledge depending on two parameters: the complexity of B and the usefulness of B, measured by u.
- Inductive Contextualisation: it is measured similarly as knowledge applicability but supplying different contexts $B_1, B_2, ..., B_T$ with different sequences $x_{i,t}$ such that $H(x_{i,t}|B_t) = H(x_{i,t}) - u$. This multiplicity of background knowledge (a new parameter T) distinguishes this factor from the previous one.
- Inductive Knowledge Construction (or learning from precedents): a set of sequences x_i is provided such that there exists a common knowledge or context B and a constant u such that for $H(x_i|B) \leq H(x_i) - u$. A significant increase of performance must take place between the first sequence and the later sequences. The parameters are the same as the first case, the complexity of B and the constant u.

It is obvious that these four factors should correlate, especially with the first one, which constitutes a necessary condition for having a minimal score in the other factors.

6 DEPENDENCIES AND OTHER FACTORS

Although there is a common (but argueable) view of induction and deduction as inverse processes, they are not inverse in the way they use computational resources. In fact, any inductive process requires deduction to check the hypotheses. Hence, obviously, inductive ability is influenced by deductive ability. This has been usually recognised by IQ tests, where deductive and inductive abilities usually correlate. Due to this fact, inductive factors usually are the main part of intelligence tests, because deductive abilities are implicitly evaluated.

However, if we are looking for 'pure' factors the question is whether there is a way to separate this deductive 'contamination' in inductive factors.

The idea is to provide 'external' deductive abilities when measuring inductive factors, in order to 'discount' the deductive effort that otherwise should be done. For this, given a problem $\pi(\cdot) = \langle S, C, \phi \rangle$ it is only necessary to provide an 'oracle' which computes ϕ in constant time. The subject must only guess models (hypotheses) and check them in the oracle, by providing the hypothesis to it and comparing the results with the evidence I. This would measure the 'creative' part of induction. In the following, let us denote by 'purely' inductive the corresponding factors to those highlighted in the previous section which result from providing the oracle.

This resembles a 'trial and error' problem considering reality acting as the oracle. The issue is how to implement this in a feasible way, especially for evaluating complex agents or even human beings. The best way, in our opinion, is the construction of a 'virtual' world where the subject to be evaluated can interact and essay its hypotheses with no effort.

In a similar way as the oracle for ϕ, some difference could be estimated if the validity verifier C is (also) given. Although this would not be much representative for deduction, for induction it would discount the ability of working with the selection criterion, which is an important trait of induction, at least to see whether a 'wrong' selection criterion is embeded in the system.

Nonetheless, deductive ability is also influenced by inductive ability as long as the problems become harder. Some lemmata or rules can be generated by an intelligent subject in order to help to shorten the proof from the premises to the conclusion. This may explain why artificial problem solvers without inductive abilities have not been able to solve complex problems, and this is especially clear in Automatic Theorem Proving. Consequently, recent systems are beginning to use ML techniques for improving performance. Background knowledge could also be examined in deduction, provided S includes the axioms but also some useful properties. This finally gives similar factors as those given for induction:

- Deductive Knowledge Applicability: how lemmata or properties are used for a deductive problem.
- Deductive Contextualisation: the ability of using different contexts for different problems.
- Deductive Knowledge Construction: this will measure the increase of performance between first instances and last ones.

Finally, we have given a measurement for sequential induction, and it seems interesting to evaluate non-sequential induction as well, where an unordered set of elements is given as evidence from an unknown function that maps whether an element belongs to a set. In this case, the test could give some possible values which might be members of the set, although only one of them is really in it. Solomonoff formalised deterministic (sequential) prediction [20] and recently, has formalised non-sequential prediction [22]. This problem is similar to the inductive problem of learning a Boolean classifier and can be extended to the case of a general classifier. To eliminate the deductive contamination of the measurement of non-sequential induction, the 'oracle' ϕ should be a classifier, telling, given a hypothesis, to which class the element belongs. The essay of an 'oracle' that accepts several elements at a time should be considered as well.

Once the basic deductive and inductive factors have been recognised, the question is whether there are many other factors which are relevant to be measured. For instance, memory or 'memo-isation ability' is a factor that is knowledge-independent and it can be easily measured. However, this factor is not very interesting for AI nowadays.

Other factors, such as analogical and abductive abilities can be shown to be closely connected to inductive and deductive abilities both theoretically and experimentally. A first approach for measuring them has been attempted in [13], and the test applied to human beings has shown the correlation with inductive abilities.

However, not every factor is meaningful. Factors like "playing chess well" are much too specific to be robust to the subject's background knowledge. However, it cannot be discarded that some game-playing factor could measure competitivity and interactivity abilities aside from deductive and inductive abilities.

Finally, we have considered individual tests which measure one factor. For measuring several factors at a time, the exercises should be given one by one and, after each guess, the subject should be given the correct answer (rewards and penalties can be used instead). This has two advantages: there is no need for the subject to understand natural language (or any language) to order to be explained the purpose of the test, and there is no need to tell which factor or purpose is to be measured in each part of the test. There is also one disadvantage, deductive problems should be posed in terms of 'learn to solve', and this may devirtualise them.

7 APPLICATIONS

Modern AI systems are much more functional than systems from the sixties or the seventies. They solve problems in an automated way that before required human intervention. However, these complex problems are solved because a methodical solution is found by the system's designers, not because most current systems are more intelligent than preceding ones. Fortunately, the initial aim of being more general is still represented by some subfields of AI: automated reasoning and machine learning.

Automated reasoning (more properly called Automatic Theorem Proving) is addressing more complex problems by the use of inductive techniques, while maintaing their general deductive techniques. These systems, in fact, have been used as the 'rational core' of many systems: knowledge-based systems, expert systems, deductive databases, ... But, remarkably, the evaluation of the growth of automated reasoning has not been established from the success of these applications but from the increasingly better results on libraries of problems, such as the TPTP library [23]. However, there is no theoretical measurement about the complexity of the problems which compose these libraries. Instead, some approximations, such as the number of clauses, use of some lemmatas, etc., have been used. Following the approach presented in this paper it would be interesting to give a value of $k - solvability$ of each of the instances of these libraries.

In a similar way, machine learning has recently taken a more experimental character and systems are evaluated wrt. sets of problems. Except from general problems (classes), where their complexity (or learnability) has been established, there is no formal framework for giving a scale for concrete instances.

In this new and beneficial interest in measurement, Bien et al. [1] have defined a 'Machine Intelligence Quotient' (MIQ), or, more precisely, two MIQs, from ontological and phenomenological (comparative) views. Any comparison needs a reference, and the only reference of intelligence is, for the moment, the human being. This makes the approach very anthropocentric, like the Turing Test. The ontological approach, however, is not based on computational principles but on a series of characteristics of intelligence that are defined on linguistical terms rather than computational/mathematical ones, such as long-term learning, adaptation, recognition, optimization, etc.

The evaluations generally based on performance in some specific problems highlight the difference between measuring functionality and general intelligence. Although measures of performance can be very appropriate for specific systems where functionality is clear, in general this would not allow for the comparison of intelligence skills of different systems devised for quite different goals. We share the view that "it is time to begin to distinguish between general, intelligent programs and the special performance systems" [19]. How to define general and absolute characteristics of intelligence computationally is more difficult but also more attractive. In any way, in our opinion, the real progress in

the 'intelligence' of AI systems can only be measured in this way.

8 CONCLUSIONS AND FUTURE WORK

From a theoretical point of view, the framework presents a formal, non-anthropomorphic and non-specific functionality oriented measurement of intelligence factors. This clarifies the distinction among evolutionary-acquired knowledge, life-acquired-knowledge and 'liquid intelligence' (or individual adaptability), or in AI terminology, among functionality, performance and adaptability.

Among the problems for making these measurement reliable there is the selection of a reference machine. The evaluation of abilities with instances is dangerous because it depends on constants. Since there is no apparent preference for any descriptional mechanism, we plan to adapt these notions for logic programming, because it is a paradigm that has been used both for automated deduction and machine learning (ILP) as well as other uses (abduction, theory revision, ...), and, in our opinion, is not biased.

Several tests for different subfields of AI could be devised following this paradigm, and the increasing scores for solving more and more complex (k-solvable) problems may be a way to know how much intelligent AI systems are wrt. previous generations systems.

REFERENCES

[1] Bien, Z., Kim Y. T. and Yang, S. H., "How to Measure the Machine Intelligence Quotient (MIQ): Two Methods and Applications", *World Automation Congress (WAC)*, TSI Press, Albuquerque, NM, 1998.

[2] Blum L. and Blum M., "Towards a Mathematical Theory of Inductive Inference". *Inf. and Control*, 28:125–155, 1975.

[3] Bradford P. G. and Wollowski, M., "A Formalization of the Turing Test (The Turing Test as an Interactive Proof System)". *SIGART Bulletin*, **6**(4), p. 10, 1995.

[4] Chaitin, G. J., "Gödel's Theorem and Information". *Int. J. Theo. Phys.*, **21**, 941-54, 1982.

[5] Eysenck, H. J., *The Structure and Measurement of Intelligence*, Springer-Verlag, 1979.

[6] Gammerman, A. and Vovk, V. (eds.), Special Issue on Kolmogorov Complexity, *The Computer Journal*, **42**(4), 1999.

[7] Gold, E. M., "Language Identification in the Limit", *Inform & Control*, **10**, 447-474, 1967.

[8] Harman, G., "The inference to the best explanation", *Philos. Review*, **74**, 88-95, 1965.

[9] Herken, R., *The universal Turing machine: a half-century survey*, Oxford Univ. Press, 1988, 2nd Ed., 1994.

[10] Hernández-Orallo, J., "Computational Gain and Inference", *Collegium Logicum*, **4**, Springer, in press, 2000.

[11] Hernández-Orallo, J., "Beyond the Turing Test", to appear in *Journal of Logic, Language and Information*, Vol. 9 no. 4, 2000.

[12] Hernández-Orallo, J., "Computational Measures of Information Gain and Reinforcement in Inference Processes", to appear in *AI Communications*, 2000.

[13] Hernández-Orallo, J. and Minaya-Collado, N., "A Formal Definition of Intelligence Based on an Intensional Variant of Kolmogorov Complexity" In *Proc. of the Intl. Symp. of Engin. of Intelligent Systems* (EIS'98), ICSC Press, 146–163. 1998.

[14] Johnson, W. L., "Needed: A New Test of Intelligence", *SIGART Bulletin*, **3**(4), 7–9, 1992.

[15] Kolmogorov, A. N., "Three Approaches to the Quantitative Definition of Information", *Problems Inform. Transmission*, **1**(1):1-7, 1965.

[16] Levin, L. A., "Universal search problems", *Problems Inform. Transm.*, **9**, 265-6, 1973.

[17] Li, M. and Vitányi, P., *An Introduction to Kolmogorov Complexity and its Applications*, 2nd Ed., Springer-Verlag, 1997.

[18] Neisser, U., Boodoo, G., Bouchard, T. J., Boykin, A. W., Brody, N., Ceci, S. J. Halpem, D. F., Lochlin, J. C., Perloff, R., Sternberg, R. J. and Urbina, S., "Intelligence: Knowns and Unknowns", *American Psychologist*, **51**, 77–101, 1996.

[19] N. J. Nilsson, Eye on the Prize. *AI Magazine*, July 1995.

[20] Solomonoff, R. J., "A formal theory of inductive inference", *Inf. Control*, **7**, 1-22, March, 224–254, June, 1964.

[21] Solomonoff, R. J., "Complexity-based induction sytems: comparisons and convergence theorems", *IEEE Trans. Inform. Theory*, IT-**24**, 422–438, 1978.

[22] Solomonoff, R. J., "Two Kinds of Probabilistic Induction", in the 'Special Issue on Kolmogorov Complexity', *The Computer Journal*, **42**(4), 256–259, 1999.

[23] Suttner, C. B. and Sutcliffe, G., "The TPTP Problem Library: CNF Release v1.2.1", *Journal of Automated Reasoning*, **21**(2), 177–203, 1998.

[24] Turing, A. M., "On computable numbers with an application to the Entscheidungsproblem", *Proc. London Math. Soc.*, series 2, **42**, 230–65, 1936. Cor., Ibid, **43**, 544–6, 1937.

[25] Turing, A. M., "Computing Machinery and Intelligence", *Mind*, **59**, 433–460, 1950.