The ANYNT Project Intelligence Test Λ_{one}

Javier Insa-Cabrera[1] and **José Hernández-Orallo**[2] and **David L. Dowe**[3] and **Sergio España**[4] and **M.Victoria Hernández-Lloreda**[5]

Abstract. All tests in psychometrics, comparative psychology and cognition which have been put into practice lack a mathematical (computational) foundation or lack the capability to be applied to any kind of system (humans, non-human animals, machines, hybrids, collectives, etc.). In fact, most of them lack both things. In the past fifteen years, some efforts have been done to derive intelligence tests from formal intelligence definitions or vice versa, grounded on computational concepts. However, some of these approaches have not been able to create *universal* tests (i.e., tests which can evaluate any kind of subjects) and others have even failed to make a feasible test. The ANYNT project was conceived to explore the possibility of defining formal, universal and anytime intelligence tests, having a feasible implementation in mind. This paper presents the basics of the theory behind the ANYNT project and describes one of the test propotypes that were developed in the project: test Λ_{one}.

Keywords: (machine) intelligence evaluation, universal tests, artificial intelligence, Solomonoff-Kolmogorov complexity.

1 INTRODUCTION

There are many examples of intelligence tests which work in practice. For instance, in psychometrics and comparative psychology, tests are used to evaluate intelligence for a variety of subjects: children and adult Homo Sapiens, other apes, cetaceans, etc. In artificial intelligence, we are well aware of some incarnations and different variations of the Turing Test, such as the Loebner Prize or CAPTCHAs [32], which are also feasible and informative. However, they do not answer the pristine questions: what intelligence is and how it can be built.

In the past fifteen years, some efforts have been done to derive intelligence tests from formal intelligence definitions or vice versa, grounded on computational concepts. However, some of these approaches have not been able to create *universal* tests (i.e., tests which can evaluate any kind of subjects) and others have even failed to make a feasible test. The ANYNT project[6] was conceived to explore the possibility of defining formal, universal and anytime intelligence tests, having a feasible implementation in mind.

[1] DSIC, Universitat Politècnica de València, Spain. email: jinsa@dsic.upv.es
[2] DSIC, Universitat Politècnica de València, Spain. email: jorallo@dsic.upv.es
[3] Clayton School of Information Technology, Monash University, Australia. email: david.dowe@monash.edu
[4] PROS, Universitat Politècnica de València, Spain. email: sergio.espana@pros.upv.es
[5] Universidad Complutense de Madrid, Spain. email: vhlloreda@psi.ucm.es
[6] http://users.dsic.upv.es/proy/anynt/

In the ANYNT project we have been working on the design and implementation of a general intelligence test, which can be feasibly applied to a wide range of subjects. More precisely, the goal of the project is to develop intelligence tests that are: (1) *formal*, by using notions from Algorithmic Information Theory (a.k.a. Kolmogorov Complexity) [24]; (2) *universal*, so that they are able to evaluate the general intelligence of any kind of system (human, non-human animal, machine or hybrid). Each will have an appropriate interface that fits its needs; (3) *anytime*, so the more time is available for the evaluation, the more reliable the measurement will be.

2 BACKGROUND

In this section, we present a short introduction to the area of Algorithmic Information Theory and the notions of Kolmogorov complexity, universal distributions, Levin's Kt complexity, and its relation to the notions of compression, the Minimum Message Length (MML) principle, prediction, and inductive inference. Then, we will survey the approaches that have appeared using these formal notions in order to give mathematical definitions of intelligence or to develop intelligence tests from them, starting from the compression-enhanced Turing tests, the C-test, and Legg and Hutter's definition of Universal Intelligence.

2.1 Kolmogorov complexity and universal distributions

Algorithmic Information Theory is a field in computer science that properly relates the notions of computation and information. The key idea is the notion of the Kolmogorov Complexity of an object, which is defined as the length of the shortest program p that outputs a given string x over a machine U. Formally,

Definition 1 Kolmogorov Complexity

$$K_U(x) := \min_{p \text{ such that } U(p)=x} l(p)$$

where $l(p)$ denotes the length in bits of p and $U(p)$ denotes the result of executing p on U.

For instance, if $x = 1010101010101010$ and U is the programming language Lisp, then $K_{Lisp}(x)$ is the length in bits of the shortest program in Lisp that outputs the string x. The relevance of the choice of U depends mostly on the size of x. Since any universal machine can emulate another, it holds that for every two universal Turing machines U and V, there is a constant $c(U, V)$, which only depends on U and V and does not depend on x, such that for all x, $|K_U(x) - K_V(x)| \leq c(U, V)$. The value of $c(U, V)$ is relatively small for sufficiently long x.

From Definition 1, we can define the universal probability for machine U as follows:

Definition 2 Universal Distribution

Given a prefix-free machine[7] *U, the universal probability of string x is defined as:*

$$p_U(x) := 2^{-K_U(x)}$$

which gives higher probability to objects whose shortest description is small and gives lower probability to objects whose shortest description is large. Considering programs as hypotheses in the hypothesis language defined by the machine, paves the way for the mathematical theory of inductive inference and prediction. This theory was developed by Solomonoff [28], formalising Occam's razor in a proper way for prediction, by stating that the prediction maximising the universal probability will eventually discover any regularity in the data. This is related to the notion of Minimum Message Length for inductive inference [34][35][1][33] and is also related to the notion of data compression.

One of the main problems of Algorithmic Information Theory is that Kolmogorov Complexity is uncomputable. One popular solution to the problem of computability of K for finite strings is to use a time-bounded or weighted version of Kolmogorov complexity (and, hence, the universal distribution which is derived from it). One popular choice is Levin's Kt complexity [23][24]:

Definition 3 Levin's Kt Complexity

$$Kt_U(x) := \min_{p \text{ such that } U(p)=x} \{l(p) + \log time(U, p, x)\}$$

where $l(p)$ denotes the length in bits of p, $U(p)$ denotes the result of executing p on U, and $time(U, p, x)$ denotes the time[8] *that U takes executing p to produce x.*

Finally, despite the uncomputability of K and the computational complexity of its approximations, there have been some efforts to use Algorithmic Information Theory to devise optimal search or learning strategies. Levin (or universal) search [23] is an iterative search algorithm for solving inversion problems based on Kt, which has inspired other general agent policies such as Hutter's AIXI, an agent that is able to adapt optimally[9] in all environments where any other general purpose agent can be optimal [17], for which there is a working approximation [31][30].

2.2 Developing mathematical definitions and tests of intelligence

Following ideas from A.M. Turing, R.J. Solomonoff, E.M. Gold, C.S. Wallace, M. Blum, G. Chaitin and others, between 1997 and 1998 some works on enhancing or substituting the Turing Test [29] by inductive inference tests were developed, using Solomonoff prediction theory [28] and related notions, such as the Minimum Message Length (MML) principle. On the one hand, Dowe and Hajek [2][3][4] suggested the introduction of inductive inference problems in a somehow *induction-enhanced* or *compression-enhanced* Turing Test (they arguably called it non-behavioural) in order to, among other things, completely dismiss Searle's Chinese room [27] objection, and also because an inductive inference ability is a necessary (though possibly "not sufficient") requirement for intelligence.

Quite simultaneously and similarly, and also independently, in [13][6], intelligence was defined as the ability to comprehend, giving a formal definition of the notion of comprehension as the identification of a 'predominant' pattern from a given evidence, derived from Solomonoff prediction theory concepts, Kolmogorov complexity and Levin's Kt. The notion of comprehension was formalised by using the notion of "projectible" pattern, a pattern that has no exceptions (no noise), so being able to explain *every* symbol in the given sequence (and not only most of it).

From these definitions, the basic idea was to construct a *feasible* test as a set of series whose shortest pattern had no alternative projectible patterns of similar complexity. That means that the "explanation" of the series had to be much more plausible than other plausible hypotheses. The main objective was to reduce the subjectivity of the test — first, because we need to choose one reference universal machine from an infinite set of possibilities; secondly, because, even choosing one reference machine, two very different patterns could be consistent with the evidence and if both have similar complexities, their probabilities would be close, and choosing between them would make the series solution quite uncertain. With the constraints posed on patterns and series, both problems were not completely solved but minimised.

$k = 9$	:	a, d, g, j, ...	Answer: m
$k = 12$	:	a, a, z, c, y, e, x, ...	Answer: g
$k = 14$	:	c, a, b, d, b, c, c, e, c, d, ...	Answer: d

Figure 1. Examples of series of Kt complexity 9, 12, and 14 used in the C-test [6].

The definition was given as the result of a *test*, called C-test [13], formed by computationally-obtained series of increasing complexity. The sequences were formatted and presented in a quite similar way to psychometric tests (see Figure 1) and, as a result, the test was administered to humans, showing a high correlation with the results of a classical psychometric (IQ) test on the same individuals. Nonetheless, the main goal was that the test could eventually be administered to other kinds of intelligent beings and systems. This was planned to be done, but the work from [26] showed that machine learning programs could be specialised in such a way that they could score reasonably well on some of the typical IQ tests. A more extensive treatment of this phenomenon and the inadequacy of *current* IQ tests for evaluating machines can be found in [5]. This unexpected result confirmed that C-tests had important limitations and could not be considered universal in two ways, i.e., embracing the whole notion of intelligence, but perhaps only a part of it, and being applicable to any kind of subject (not only adult humans). The idea of extending these static tests to other factors or to make them interactive and extensible to other kinds of subjects by the use of rewards (as in the area of reinforcement learning) was suggested in [7][8], but not fully

[7] For a convenient definition of the universal probability, we need the requirement of U being a prefix-free machine (see, e.g., [24] for details). Note also that even for prefix-free machines there are infinitely many other inputs to U that will output x, so $p_U(x)$ is a strict lower bound on the probability that U will output x (given a random input)

[8] Here *time* does not refer to physical time but to computational time, i.e., computation steps taken by machine U. This is important, since the complexity of an object cannot depend on the speed of the machine where it is run.

[9] Optimality has to be understood in an asymptotic way. First, because AIXI is uncomputable (although resource-bounded variants have been introduced and shown to be optimal in terms of time and space costs). Second, because it is based on *a* universal probability over *a* machine, and this choice determines a constant term which may very important for small environments.

developed into actual tests. An illustration of the classical view of an environment in reinforcement learning is seen in Figure 2, where an agent can interact through actions, rewards and observations.

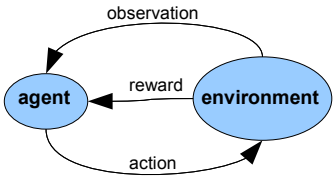

Figure 2. Interaction with an Environment.

A few years later, Legg and Hutter (e.g. [21],[22]) followed the previous steps and, strongly influenced by Hutter's theory of AIXI optimal agents [16], gave a new definition of machine intelligence, dubbed "Universal[10] Intelligence", also grounded in Kolmogorov complexity and Solomonoff's ("inductive inference" or) prediction theory. The key idea is that the intelligence of an agent is evaluated as some kind of sum (or weighted average) of performances in all the possible environments (as in Figure 2).

The definition based on the C-test can now be considered a static precursor of Legg and Hutter's work, where the environment outputs no rewards, and the agent is not allowed to make an action until several observations are seen (the inductive inference or prediction sequence). The point in favour of active environments (in contrast to passive environments) is that the former not only require inductive and predictive abilities to model the environment but also some planning abilities to effectively use this knowledge through actions. Additionally, perceptions, selective attention, and memory abilities must be fully developed. Not all this is needed to score well in a C-test, for instance.

While the C-test selects the problems by (intrinsic) difficulty (which can be chosen to fit the level of intelligence of the evaluee), Legg and Hutter's approach select problems by using a universal distribution, which gives more probability to simple environments. Legg and Hutter's definition, given an agent π, is given as:

Definition 4 Universal Intelligence [22]

$$\Upsilon(\pi, U) = \sum_{\mu=i}^{\infty} p_U(\mu) \cdot E\left(\sum_{i=1}^{\infty} r_i^{\mu,\pi}\right)$$

where μ is any environment coded on a universal machine U, with π being the agent to be evaluated, and $r_i^{\mu,\pi}$ the reward obtained by π in μ at interaction i. E is the expected reward on each environment, where environments are assigned with probability $p_U(\mu)$ using a universal distribution [28].

Definition 4, although very simple, captures one of the broadest definitions of intelligence: "the ability to adapt to a wide range of environments". However, this definition was not meant to be eventually converted into a test. In fact, there are three obvious problems in this definition regarding making it practical. First, we have two infinite sums in the definition: one is the sum over all environments, and the second is the sum over all possible actions (agent's life in each environment is infinite). And, finally, K is not computable. Additionally, we also have the dependence on the reference machine U. This dependence takes place even though we consider an infinite number of environments. The universal distribution for a machine U could give the higher probabilities $(0.5, 0.25, ...)$ to quite different environments than those given by another machine V.

Despite all these problems, it could seem that just making a random finite sample on environments, limiting the number of interactions or cycles of the agent with respect to the environment and using some computable variant of K, is sufficient to make it a practical test. However, on the one hand, this is not so easy, and, on the other hand, the definition has many other problems (some related and others unrelated).

The realisation of these problems and the search for solutions in the quest of a practical intelligence *test* is the goal of the ANYNT project.

3 ANYTIME UNIVERSAL TESTS

This section presents a summary of the theory in [11]. The reader is referred to this paper for further details.

3.1 On the difficulty of environments

The first issue concerns how to sample environments. Just using the universal distribution for this , as suggested by Legg and Hutter, will mean that very simple environments will be output again and again. Note that an environment μ with $K(\mu) = 1$ will appear half of the time. Of course, repeated environments must be ruled out, but a sample would almost become an enumeration from low to high K. This will still omit or underweight very complex environments because their probability is so low. Furthermore, measuring rewards on very small environments will get very unstable results and be very dependent on the reference machine. And even ignoring this, it is not clear that an agent that solves all the problems of complexity lower than 20 bits and none of those whose complexity is larger than 20 bits is more intelligent than another agent who does reasonably well on every environment.

This constrasts with the view of the C-test, which focus on the issue of difficulty and does not make the probability of a problem appearing inversely related to this difficulty. In any case, before going on, we need to clarify the notions of simple/easy and complex/difficult that are used here. For instance, just choosing an environment with high K does not ensure that the environment is indeed complex. As Figure 3 illustrates, the relation is unidirectional; given a low K, we can affirm that the environment will look simple. On the other hand, given an intuitively complex environment, K must be necessarily high.

Environment with high K $\Longleftarrow$ Intuitively complex (difficult) environment
Environment with low K $\Longrightarrow$ Intuitively simple (easy) environment

Figure 3. Relation between K and intuitive complexity.

Given this relation, only among environments with high K will we find complex environments, and, among the latter, not all of them will be difficult. From the agent's perspective, however, this is more extreme, since many environments with high K will contain difficult patterns that will never be accessed by the agent's interactions.

[10] The term 'universal' here does not refer to the definition (or a derived test) being applicable to any kind of agent, but to the use of Solomonoff's universal distribution and the view of the definition as an extremely general view of intelligence.

As a result, the environment will be *probabilistically* simple. Thus, giving most of the probability to environments with low K means that most of the intelligence measure will come from patterns that are extremely simple.

3.2 Selecting discriminative environments

Furthermore, many environments (either simple or complex) will be completely useless for evaluating intelligence, e.g., environments that stop interacting, environments with constant rewards, etc. If we are able to make a more accurate sample, we will be able to make a more efficient test procedure. The question here is to determine a non-arbitrary criterion to exclude some environments. For instance, Legg and Hutter's definition forces environments to interact infinitely, and since the description must be finite, there must be a pattern. This obviously includes environments such as "always output the same observation and reward". In fact, they are not only possible but highly probable on many reference machines. Another pathological case is an environment that "outputs observations and rewards at random". However, this has a high complexity if we assume deterministic environments. In both cases, the behaviour of any agent on these environments would almost be the same. In other words, they do not have *discriminative power*. Therefore, these environments would be useless for discriminating between agents.

In an interactive environment, a clear requirement for an environment to be discriminative is that what the agent does must have consequences on rewards. Thus, we will restrict environments to be sensitive to agents' actions. That means that a wrong action might lead the agent to part of the environment from which it can never return (non-ergodic), but at least the actions taken by the agent can modify the rewards in that subenvironment. More precisely, *we want an agent to be able to influence rewards at any point in any subenvironment*. This does not imply ergodicity but reward sensitivity at any moment. That means that we cannot reach a point from which rewards are given independently of what we do (a dead-end).

3.3 Symmetric rewards and balanced environments

An important issue is how to estimate rewards. If we only use positive rewards, we find some problems. For example, an increase in the score may originate from a really good behaviour on the environment or just because more rewards are accumulated since they are always positive. Instead, an average reward seems a better payoff function. Our proposal is to use symmetric rewards, which can range between -1 and 1:

Definition 5 Symmetric Rewards

We say an environment has symmetric rewards when:

$$\forall i : -1 \leq r_i \leq 1$$

If we set symmetric rewards, we also expect environments to be symmetric, or more precisely, to be balanced on how they give rewards. This can be seen in the following way. In a reliable test, we would like that many (if not all) environments give an expected 0 reward to random agents.

This excludes both hostile and benevolent environments, i.e., environments where doing randomly will get more negative (respectively positive) rewards than positive (respectively negative) rewards. In many cases it is not difficult to prove that a particular environment is balanced. Another approach is to set a reference machine that only generates balanced environments.

Using this approach on rewards, we can use an average to estimate the results on each environment, namely:

Definition 6 Average Reward

Given an environment μ, with n_i being the number of completed interactions, then the average reward for agent π is defined as follows:

$$v_\mu^\pi(n_i) = \frac{\sum_{i=1}^{n_i} r_i^{\mu,\pi}}{n_i}$$

Now we can calculate the expected value (although the limit may not exist) of the previous average, denoted by $E(v_\mu^\pi)$, for an arbitrarily large value of n_i.

To view the test framework in more detail, in [11] some of these issues (and many other problems) of the measure are solved. It uses a random finite sample of environments. It limits the number of interactions of the agent with respect to the environment. It selects a discriminative set of environments, etc.

4 ENVIRONMENT CLASS

The previous theory, however, does not make the choice for *an* environment class, but just sets some constraints on the kind of environments that can be used. Consequently, one major open problem is to make this choice, i.e., to find a proper (unbiased) environment class which follows the constraints and, more difficult, which can be feasibly implemented. Once this environment class is identified, we can use it to generate environments to run any of the tests variants. Additionally, it is not only necessary to determine the environment class, but also to determine the universal machine we will use to determine the Kolmogorov complexity of each environment, since the tests only use a (small) sample of environments, and the sample probability is defined in terms of the complexity.

In the previous section we defined a set of properties which are required for making environments discriminative, namely that observations and rewards must be sensitive to agent's actions and that environments are balanced. Given these constraints if we decide to generate environments without any constraint and then try to make a post-processing sieve to select which of them comply with all the constraints, we will have a computationally very expensive or even incomputable problem. So, the approach taken is to generate an environment class that ensures that these properties hold. In any case, we have to be very careful, because we would not like to restrict the reference machine to comply with these properties at the cost of losing their universality (i.e. their ability to emulate or include any computable function).

And finally, we would like the environment class to be user-friendly to the kind of systems we want to be evaluated (humans, non-human animals and machines), but without any bias in favour or against some of them.

According to all this, we define a universal environment class from which we can effectively generate valid environments, calculate their complexity and consequently derive their probability.

4.1 On actions, observations and space

Back to Figure 2 again, actions are limited by a finite set of symbols A, (e.g. $\{left, right, up, down\}$), rewards are taken from any subset R of rational numbers between -1 and 1, and observations are also

limited by a finite set O of possibilities (e.g., the contents of a grid of binary cells of $n \times m$, or a set of light-emitting diodes, LEDs). We will use a_i, r_i and o_i to (respectively) denote action, reward and observation at interaction i.

Apart from the behaviour of an environment, which may vary from very simple to very complex, we must first clarify the *interface*. How many actions are we going to allow? How many different observations? The very definition of environment makes actions a finite set of symbols and observations are also a finite set of symbols. It is clear that the minimum number of actions has to be two, but no upper limit seems to be decided a priori. The same happens with observations. Even choosing two for both, a sequence of interactions can be as rich as the expressiveness of a Turing machine.

Before getting into details with the interface, we have to think about environments that can contain agents. This is not only the case in real life (where agents are known as inanimate or animate objects, animals among the latter), but also a requirement for evolution and, hence, intelligence as we know it. The existence of several agents which can interact requires a *space*. The space is not necessarily a virtual or physical space, but also a set of common rules (or laws) that govern what the agents can perceive and what the agents can do. From this set of common rules, specific rules can be added to each agent. In the real world, this set of common rules is physics. All this has been extensively analysed in multi-agent systems (see e.g. [20] for a discussion).

The good thing about thinking of spaces is that a space entails the possible perceptions and actions. If we define a common space, we have many choices about observations and actions already taken.

A first (and common) idea for a space is a 2D grid. From a 2D grid, the observation is a picture of the grid with all the objects and agents inside. In a simple grid where we have agents and objects inside the cells, the typical actions are the movements left, right, up and down. Alternatively, of course, we could use a 3D space, since our world is 3D. In fact, there are some results using intelligence testing (for animals or humans) with a 3D interface [25][36].

The problem of a 2D or 3D grid is that it is clearly biased in favour of humans and many other animals which have hardwired abilities for orientation in this kind of spaces. Other kinds of animals or handicapped people (e.g. blind people) might have some difficulties in this type of spaces. Additionally, artificial intelligence agents would highly benefit by hardwired functionalities about Euclidean distance and 2D movement, without any real improvement in their general intelligence.

Instead we propose a more general kind of space. A 2D grid is a graph with a very special topology, where there are concepts which hold such as direction, adjacency, etc. A generalisation is a graph where the cells are freely connected to some other cells with no particular predefined pattern. This suggests a (generally) dimensionless space. Connections between cells would determine part or all the possible actions, and observations and rewards can be easily shown graphically.

4.2 Definition of the environment class

After the previous discussion, we are ready to give the definition of the environment class. First we must define the space and objects, and from here observations, actions and rewards. Before that, we have to define some constants that affect each environment. Namely, with $n_a = |A| \geq 2$ we denote the number of actions, with $n_c \geq 2$ the number of cells, and with n_ω the number of objects/agents (not including the agent which is to be evaluated and two special objects known as Good and Evil).

4.2.1 Space

The space is defined as a directed labelled graph of n_c nodes (or vertices), where each node represents a cell. Nodes are numbered, starting from 1, so cells are refered to as $C_1, C_2, \ldots, C_{n_c}$. From each cell we have n_a outgoing arrows (or arcs), each of them denoted as $C_i \rightarrow_\alpha C_j$, meaning that action $\alpha \in A$ goes from C_i to C_j. All the outgoing arrows from C_i are denoted by $\vec{C_i}$. At least two outgoing arrows cannot go to the same cell. Formally, $\forall C_i : \exists r_1, r_2 \in \vec{C_i}$ such that $r_1 = C_i \rightarrow_{\alpha_m} C_j$ and $r_2 = C_i \rightarrow_{\alpha_n} C_k$ with $C_j \neq C_k$ and $\alpha_m \neq \alpha_n$. At least one of the outgoing arrows from a cell must lead to itself (typically denoted by α_1 and is the first action). Formally, $\forall C_i : \exists r \in \vec{C_i}$ such that $r = C_i \rightarrow_{\alpha_1} C_i$.

A path from C_i to C_m is a sequence of arrows $C_i \rightarrow C_j, C_j \rightarrow C_k, \ldots, C_l \rightarrow C_m$. The graph must be strongly connected, i.e., all cells must be connected (i.e. there must be a walk over the graph that goes through all its nodes), or, in other words, for every two cells C_i, C_j there exists a path from C_i to C_j.

4.2.2 Objects

Cells can contain objects from a set of predefined objects Ω, with $n_\omega = |\Omega|$. Objects, denoted by ω_i can be animate or inanimate, but this can only be perceived by the rules each object has. An object is inanimate (for a period or indefinitely) when it performs action α_1 repeatedly. Objects can perform actions following the space rules, but apart from these rules, they can have any behaviour, either deterministic or not. Objects can be reactive and can be defined to act with different actions according to their observations. Objects perform one and only one action at each interaction of the environment (except from the special objects Good and Evil, which can perform several actions in a row).

Apart from the evaluated agent π, as we have mentioned, there are two special objects called Good and Evil. Good and Evil must have the same behaviour. By the *same* behavior we do not mean that they perform the same movements, but they have the same *logic* or *program* behind them.

Objects can share a same cell, except Good and Evil, which cannot be at the same cell. If their behaviour leads them to the same cell, then one (chosen randomly with equal probability) moves to the intended cell and the other remains at its original cell. Because of this, the environment becomes stochastic (non-deterministic).

Objects are placed randomly at the cells with the initialisation of the environment. This is another source of stochastic behaviour.

4.2.3 Observations and Actions

The observation is a sequence of cell contents. The cells are ordered by their number. Each element in the sequence shows the presence or absence of each object, included the evaluated agent. Additionally, each cell which is reachable by an action includes the information of that action leading to the cell.

4.2.4 Rewards

Raw rewards are defined as a function of the position of the evaluated agent π and the positions of Good and Evil.

For the rewards, we will work with the notion of trace and the notion of "cell reward", that we denote by $r(C_i)$. Initially, $r(C_i) = 0$

for all i. Cell rewards are updated by the movements of Good and Evil. At each interaction, we set r_i^{Good} to the cell reward where Good is and $-r_i^{Evil}$ to the cell reward where Evil is. Each interaction, all the other cell rewards are divided by a constant n (for example $n = 2$). So, an intuitive way of seeing this is that Good leaves a positive trace and Evil leaves a negative trace. The agent π *eats* the rewards it finds in the cells it occupies. We mean it *eats*, since just after getting the reward, the cell reward is set to 0. Note that if $n = \infty$, then Good and Evil do not leave any trace of rewards.

When π moves to a cell, it gets the cell reward which is at that cell, i.e. the accumulated reward $\rho = \rho + r(C_i)$. To calculate the average of the rewards, we divide the accumulated reward by the final number of interactions (denoted by n_i). The justification for this option is further investigated in [10].

4.2.5 *Properties*

All the properties mentioned in the previous section (observation-sensitiveness, reward-sensitiveness and balancedness), are met by the environment class described here. For a proof of these properties for this environment class see [9].

5 DESCRIPTION OF THE TEST Λ_{one}

In this section we will explain how an actual test is constructed. In particular, we will see one of our prototypes: Λ_{one}. We will explain how exercises are arranged, we will see an interface for humans and we will comment on some experimental results with this test.

5.1 Episodes

Tests are sequence of exercises (or environments). In particular, Λ_{one}uses 7 environments, each with a number of cells (n_c) from 3 to 9. The size of the patterns for Good and Evil is made proportional to the number of cells, using n_c actions (on average). In each environment, we allow $10 \times (n_c - 1)$ steps, so the agents have the chance to detect any pattern in the environment (exploration) and also have some further steps to exploit the findings (in case a pattern is actually there). The limitation of the number of environments and steps is justified because the tests are meant to be applied to biological agents in a reasonable period of time (e.g., 20 minutes) and we estimate an average of 4 seconds per action. Table 1 shows the choices we made for the test:

Env. #	No. cells (n_c)	No. steps (m)	Pattern length (on average)
1	3	20	3
2	4	30	4
3	5	40	5
4	6	50	6
5	7	60	7
6	8	70	8
7	9	80	9
TOTAL	-	350	-

Table 1. Setting for the 7 environments which compose Λ_{one}.

Before each exercise starts, a random environment is created (by generating the space topology and the behaviour of Good and Evil) using the environment distribution, and the three agents (Good, Evil and the evaluated agent) are placed into the generated space. The interaction starts when the evaluated agent decides which cell to move to. Then, the three agents are moved simultaneously. Once Good and Evil move to a cell, they leave their rewards in their respective cells, and the rest of the cell rewards are deleted. Finally, the evaluated agent collects the reward of the cell where it is, and a new interaction is started. When the test ends, the score of the evaluated agent is calculated as the average of the collected rewards over the whole exercise.

5.2 Interfaces

Applying a test to a kind of individual requires an *interface* for that kind. Clearly, the same test may require different interfaces for adult humans, blind people, a dolphin or a machine. Given the formal definition of the environments, it is relatively easy to figure out an interface for machines. In fact, in our case, we just connect the environments to reinforcement learning agents, in the traditional way.

For biological agents, constructing a user interface is a delicate issue, in the sense that we must ensure that no designing decision should contaminate the evaluation. The interface for humans has been designed with the following principles in mind: 1) The subject must not perceive anything that could distract it from the test. 2) The signs used to represent observations should not have an implicit meaning for the subject (e.g. no skull-and-bones for the Evil agent). 3) Actions and rewards should be easily interpreted by the subject to avoid a cognitive overhead.

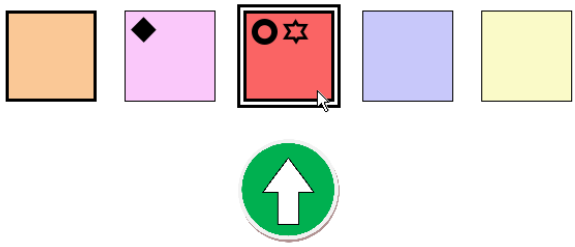

Figure 4. A snapshot of an exercise using a human interface.

In Fig. 4 we can see a snapshot of one exercise using the human interface. In the figure we see an environment with five cells. In the third cell we can see the evaluated agent (◎) sharing the cell with Good (✡). Evil (◇) can be seen in the second cell. The cells directly accessible by the evaluated agent are denoted by thick edges. The third cell has a double border because it is accessible and the user has the cursor over it. In this interaction the evaluated agent has received a positive reward (represented by a green circle with an up arrow) because it has coincided with Good in the cell.

5.3 Experiments

During the project we have made some experiments to analyse how the test works. We just include a brief excerpt from some of them. In [19] we evaluated the performance of a reinforcement learning algorithm. For this experiment, we analysed the results of a well known algorithm in reinforcement learning known as QLearning [37]. For the evaluation, we let QLearning interact with several environment

complexities, and we analysed whether the obtained results correlated with the measure of difficulty. The results were clear, showing that the evaluation obtains the expected results in terms of the relation between expected reward and theoretical problem difficulty. Also, it showed reasonable differences with other baseline algorithms (e.g. a random algorithm). All this supported the idea that the test and the environment class used are on the right direction for evaluating a specific kind of system. However, the main question was whether the approach was in the right direction in terms of constructing universal tests. In other words, it was still necessary to demonstrate if the test serves to evaluate several kinds of systems and put their results on the same scale.

In [18] we compared the results of two different systems (humans and AI algorithms), by using the prototype described in this paper and the interface for humans. We set both systems to interact with exactly the same environments. The results, not surprisingly, did not show the expected difference in intelligence between reinforcement learning algorithms and humans. This is explained by several reasons. One of them is that the environments were still relatively simple and reinforcement learning algorithms could still capture and represent all the state matrix for these problems with some partial success. Another reason is that exercises were independent, so humans could not reuse what they were learning on some exercises for others, an issue where humans are supposed to be better than these simple reinforcement algorithms. Also, another possibility is the fact that the environments had very few agents and the few agents that existed were not reactive. This makes the state space bounded, which is beneficial for Q-learning. Similarly, the environments had no noise. All these decisions were made on purpose to keep things simple and also to be able to formally derive the complexity of the environments. In general, other explanations can be found as well, since the lack of other interactive agents can be seen as a lack of social behaviours, as we explored in subsequent works [12].

Of course, test $\Lambda_{one}$was just a first prototype which does not incorporate many of the features of an anytime intelligence test and the measuring framework. Namely, the prototype is not anytime, so the test does not adapt its complexity to the subject that is evaluating. Also, we made some simplifications to the environment class, causing objects to lose reactivity. Furthermore, it is very difficult to construct any kind of social behaviour by creating agents from scratch. These and other issues are being addressed in new prototypes, some of them under development.

6 CONCLUDING REMARKS

The ANYNT project aimed at exploring the possibility of formal, universal and feasible tests. As already said, test $\Lambda_{one}$is just one prototype that does not implement all the features of the theory of *anytime universal tests*. However, it is already very informative. For instance, the experimental results show that the test Λ_{one}goes in the right direction, but it still fails to capture some components of intelligence that should put different kinds of individuals on the right scale.

In defence of test Λ_{one}, we have to say that it is quite rare in the literature to find the same test applied to different kinds of individuals[11]. In fact, as argued in [5], relatively simple programs can get good scores on conventional IQ tests, while small children (with high potential intelligence) will surely fail. Similarly, illiterate people and most children would score very badly at the Turing Test, for instance. And humans are starting to struggle with many CAPTCHAs.

All this means that many feasible and practical tests work because they are specialised for specific populations. As long as the diversity of subjects is enlarged, measuring intelligence becomes more difficult and less accurate. As a result, the mere possibility of constructing universal tests is still a hot question. While many may think that this is irresoluble, we think that unless an answer to this question is found, it will be very difficult (if not impossible) to assess the diversity of intelligent agents that are envisaged for the forthcoming decades. Being one way or another, there is clearly an ocean of scientific questions beyond the Turing Test.

[11] The only remarkable exceptions are the works in comparative psychology, such as [14][15], which are conscious of the difficulties of using the same test, with different interfaces, for different subjects.

ACKNOWLEDGEMENTS

This work was supported by the MEC projects EXPLORA-INGENIO TIN 2009-06078-E, CONSOLIDER-INGENIO 26706 and TIN 2010-21062-C02-02, and GVA project PROMETEO/2008/051. Javier Insa-Cabrera was sponsored by Spanish MEC-FPU grant AP2010-4389.

REFERENCES

[1] D. L. Dowe, 'Foreword re C.S. Wallace', *The Computer Journal*, **51**(5), 523–560, Christopher Stewart WALLACE (1933–2004) memorial special issue, (2008).

[2] D. L. Dowe and A. R. Hajek, 'A computational extension to the Turing Test', *in Proceedings of the 4th Conference of the Australasian Cognitive Science Society, University of Newcastle, NSW, Australia*, (1997).

[3] D. L. Dowe and A. R. Hajek, 'A computational extension to the Turing Test', *Technical Report #97/322, Dept Computer Science, Monash University, Melbourne, Australia, 9pp, http://www.csse.monash.edu.au/publications/1997/tr-cs97-322-abs.html*, (1997).

[4] D. L. Dowe and A. R. Hajek, 'A non-behavioural, computational extension to the Turing Test', in *International conference on computational intelligence & multimedia applications (ICCIMA'98), Gippsland, Australia*, pp. 101–106, (1998).

[5] D. L. Dowe and J. Hernandez-Orallo, 'IQ tests are not for machines, yet', *Intelligence*, **40**(2), 77–81, (2012).

[6] J. Hernández-Orallo, 'Beyond the Turing Test', *Journal of Logic, Language and Information*, **9**(4), 447–466, (2000).

[7] J. Hernández-Orallo, 'Constructive reinforcement learning', *International Journal of Intelligent Systems*, **15**(3), 241–264, (2000).

[8] J. Hernández-Orallo, 'On the computational measurement of intelligence factors', in *Performance metrics for intelligent systems workshop*, ed., A. Meystel, pp. 1–8. National Institute of Standards and Technology, Gaithersburg, MD, U.S.A., (2000).

[9] J. Hernández-Orallo, 'A (hopefully) non-biased universal environment class for measuring intelligence of biological and artificial systems', in *Artificial General Intelligence, 3rd International Conference AGI, Proceedings*, eds., Marcus Hutter, Eric Baum, and Emanuel Kitzelmann, "Advances in Intelligent Systems Research" series, pp. 182–183. Atlantis Press, (2010).

[10] J. Hernández-Orallo, 'On evaluating agent performance in a fixed period of time', in *Artificial General Intelligence, 3rd Intl Conf*, ed., M. Hutter et al., pp. 25–30. Atlantis Press, (2010).

[11] J. Hernández-Orallo and D. L. Dowe, 'Measuring universal intelligence: Towards an anytime intelligence test', *Artificial Intelligence*, **174**(18), 1508–1539, (2010).

[12] J. Hernández-Orallo, D. L. Dowe, S. España-Cubillo, M. V. Hernández-Lloreda, and J. Insa-Cabrera, 'On more realistic environment distributions for defining, evaluating and developing intelligence', in *Artificial General Intelligence 2011*, eds., J. Schmidhuber, K.R. Thórisson, and M. Looks (eds), volume 6830 of *LNAI*, pp. 82–91. Springer, (2011).

[13] J. Hernández-Orallo and N. Minaya-Collado, 'A formal definition of intelligence based on an intensional variant of kolmogorov complexity', in *In Proceedings of the International Symposium of Engineering of Intelligent Systems (EIS'98)*, pp. 146–163. ICSC Press, (1998).

[14] E. Herrmann, J. Call, M. V. Hernández-Lloreda, B. Hare, and M. Tomasell, 'Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis', *Science*, **Vol 317**(5843), 1360–1366, (2007).

[15] E. Herrmann, M. V. Hernández-Lloreda, J. Call, B. Hare, and M. Tomasello, 'The structure of individual differences in the cognitive abilities of children and chimpanzees', *Psychological Science*, **21**(1), 102, (2010).

[16] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*, Springer, 2005.

[17] M. Hutter, 'Universal algorithmic intelligence: A mathematical top→down approach', in *Artificial General Intelligence*, eds., B. Goertzel and C. Pennachin, Cognitive Technologies, 227–290, Springer, Berlin, (2007).

[18] J. Insa-Cabrera, D. L. Dowe, S. España-Cubillo, M. V. Hernández-Lloreda, and J. Hernández-Orallo, 'Comparing humans and AI agents', in *Artificial General Intelligence 2011*, eds., J. Schmidhuber, K.R. Thórisson, and M. Looks (eds), volume 6830 of *LNAI*, pp. 122–132. Springer, (2011).

[19] J. Insa-Cabrera, D. L. Dowe, and J. Hernández-Orallo, 'Evaluating a reinforcement learning algorithm with a general intelligence test', in *CAEPIA, Advances in Artificial Intelligence*, volume 7023 of *LNCS*, pp. 1–11. Springer, (2011).

[20] D. Keil and D. Goldin, 'Indirect interaction in environments for multi-agent systems', *Environments for Multi-Agent Systems II*, 68–87, (2006).

[21] S. Legg and M. Hutter, 'A universal measure of intelligence for artificial agents', in *International Joint Conference on Artificial Intelligence*, volume 19, p. 1509, (2005).

[22] S. Legg and M. Hutter, 'Universal intelligence: A definition of machine intelligence', *Minds and Machines*, **17**(4), 391–444, (2007). http://www.vetta.org/documents/UniversalIntelligence.pdf.

[23] L. A. Levin, 'Universal sequential search problems', *Problems of Information Transmission*, **9**(3), 265–266, (1973).

[24] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications (3rd ed.)*, Springer-Verlag New York, Inc., 2008.

[25] F. Neumann, A. Reichenberger, and M. Ziegler, 'Variations of the turing test in the age of internet and virtual reality', in *Proceedings of the 32nd annual German conference on Advances in artificial intelligence*, pp. 355–362. Springer-Verlag, (2009).

[26] P. Sanghi and D. L. Dowe, 'A computer program capable of passing IQ tests', in *Proc. 4th ICCS International Conference on Cognitive Science (ICCS'03), Sydney, Australia*, pp. 570–575, (July 2003).

[27] J. Searle, 'Minds, brains, and programs', *Behavioral and Brain Sciences*, **3**(3), 417–457, (1980).

[28] R. J. Solomonoff, 'A formal theory of inductive inference. Part I', *Information and control*, **7**(1), 1–22, (1964).

[29] A. M. Turing, 'Computing machinery and intelligence', *Mind*, **59**, 433–460, (1950).

[30] J. Veness, K. S. Ng, M. Hutter, and D. Silver, 'Reinforcement learning via AIXI approximation', in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pp. 605–611, (2010).

[31] J. Veness, K.S. Ng, M. Hutter, W. Uther, and D. Silver, 'A Monte Carlo AIXI Approximation', *Journal of Artificial Intelligence Research*, **40**(1), 95–142, (2011).

[32] L. Von Ahn, M. Blum, and J. Langford, 'Telling humans and computers apart automatically', *Communications of the ACM*, **47**(2), 56–60, (2004).

[33] C. S. Wallace, *Statistical and Inductive Inference by Minimum Message Length*, Ed. Springer-Verlag, 2005.

[34] C. S. Wallace and D. M. Boulton, 'A information measure for classification', *The Computer Journal*, **11**(2), 185–194, (1968).

[35] C. S. Wallace and D. L. Dowe, 'Minimum message length and Kolmogorov complexity', *Computer Journal*, **42**(4), 270–283, (1999). Special issue on Kolmogorov complexity.

[36] D.A. Washburn and R.S. Astur, 'Exploration of virtual mazes by rhesus monkeys (macaca mulatta)', *Animal Cognition*, **6**(3), 161–168, (2003).

[37] C.J.C.H. Watkins and P. Dayan, 'Q-learning', *Mach. learning*, **8**(3), 279–292, (1992).