Zoltán Sóstai
Department of Logic and Philosophy of Science at Eötvös University, Faculty of Humanities, Budapest, Hungary
ORCID: 0009-0007-8628-4276

Modelling Qualia with Physical Computers

Abstract: According to Frank Jackson's knowledge argument, Mary, who lives in a black-and-white world, has all the physical knowledge about the world, yet she has new information when she sees a red apple. If we accept this, then physicalism – according to which the description of the world can be realised entirely with the help of physical theories – is false. However, even if Jackson's argument about new information is true, we do not have to discard physicalism. Even exclusively physical computer systems are not capable of predicting their own sensory information if they are built using a specific structure and complexity. It can be shown that a specific type of computational system can lead to the creation of decision-related unpredictability of a specific type of sensory information. This investigation takes a charitable position towards the concept of qualia to provide it with a physicalist and computationalist explanation.

Keywords: Turing-Machine; Halting Problem, Knowledge Argument; Physicalism; Qualia

Introduction and problem description

In his article *Epiphenomenal Qualia*[1], Frank Jackson describes a thought experiment aimed at showing that qualia – quotients in English – are fundamental qualitative sensations that even a person with perfect physical knowledge, i.e. with precise knowledge of the initial conditions and the relevant physical laws, cannot know. If we accept the conclusion of this intuitive description of the problem as true then we must conclude that physicalism, according to which the description of the world can be fully realised by means of physical theories, is false. This intuitive knowledge argument can, however, be explained by a physicalist theory of computer systems, according to which, even in the case of computer systems bounded by the physical

[1] Jackson, F., 1982. Epiphenomenal qualia. *Philosophical Quarterly*, 32(April), pp.127-136.

Church-Turing thesis[2] – i.e. exclusively physical – it can be shown that these systems, given a certain structure and a certain complexity, are not capable of computing their own perceptual information correctly in advance - even if they possess information about all the facts of the world. This means that a fundamental unpredictability affects the knowledge of all physical computational systems. I base my original arguments in part on Popper's[3] and Lloyd's[4] arguments about the unpredictability of indeterminism and free will, taking into account their critical revision[5]. In particular, Lloyd's argument based on the computational unpredictability of physical systems will be applied and extended with a concrete physical, computational model to explain qualia. For example, unpredictability may occur in the following actual case: Input > RGB Camera > *Filter X* > Image > Agent

In the example, one element of the computer system (the image-altering *Filter X*) behaves as a black box[6] for the agent (Agent *M),* which can also be considered as another element of the system. The program of *Filter X* cannot be directly known, except by examining its output and input states. However, the computation of *Filter X* in the system occurs anyway. If in such a case the program of *Filter X is* sufficiently complicated, Agent *M* will be unable to correctly predict the operation of *Filter X* or the associated decision problem.

A question about colour information can always be trivially matched with a question about a decision problem. This can be done by asking multiple yes/no questions, e.g. *"What colour will image U(q) be?"* can be mapped to the decision problem *"Will image U(q) be yellow (or any other colour)?"*. The decision problem thus poses the question of whether the computational process that the agent perceives to be taking place in any case, where the program of an element of the process is a black box for the agent, can be predicted in advance by the agent.

[2] Deutsch, D., 1985. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society*, 400(1818), pp. 97-117.
[3] Popper, K., 1950. Indeterminism in classical and quantum physics. *British Journal for the Philosophy of Science*, 1, pp.117-133, 173-195.
[4] Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, pp.3597-3610.
[5] Author, 2023. Analysis of the physical Church-Turing thesis and some philosophical implications of the halting problem. In: M. Nemes, ed., *Impact Points IX. Proceedings of the Conference of the Philosophy Department of the National Association of Doctoral Students*. Budapest: National Association of Doctoral Students, pp.145-158.
[6] Ashby, W.R., 1957. *An Introduction to Cybernetics*. 2nd impression. London: Chapman & Hall, pp. 86-93.

A negative response to this question can have several causes, revealing deeper and deeper layers of the problem. Firstly, it is due to the problem of induction[7], whereby the program of *Filter X* cannot be unambiguously determined by examining a finite number of input-output pairs. Even though it cannot be unambiguously determined, it is still possible for *an agent* - if it has such a capability - to predict it correctly using appropriate heuristics. If, however, *Filter X* contains hidden complexity, hidden parameters[8]- especially if the hidden parameters are rapidly and continuously changing - or if *Filter X's* program is written in a Turing-complete language[9] and the program exploits its complexity, then Agent will be unable to predict *Filter X's* operation correctly.

Taking this to the next level, due to the undecidability of the Turing equivalence problem[10], it is generally undecidable for Agent *M* whether the program of the filter it predicts is identical to the program of *Filter X.* Since the Turing equivalence problem is reduced to the halting problem, the consequence of the latter's undecidability is that *Filter X*'s program is generally not correctly predicted[11]. In such a case, the general decidability problem for agent *M* can be formulated as *"Is it decidable for me whether the sensory information encountered in U(q) is the same as the sensory information produced by some program n of my own?*"

If the whole system and one of its programs can be of any complexity, we are faced with the general case of the halting problem. This however is not yet enough to consider any such described system as physical. Realistically, if we consider only finite, time-constrained computers, where *t* involves an empirical measurement constraint on the halting behaviour of any program *n*, it can be said that these *t* time-constrained systems will not be able to correctly predict their own decisions about their sensory information in *t* time[12]. This shows that for finite physical systems, the problem is undecidable.

---

[7] Henderson, L., 2022. The Problem of Induction. In: E.N. Zalta and U. Nodelman, eds., *The Stanford Encyclopedia of Philosophy (Winter 2022 Edition)*. [online] Available at: https://plato.stanford.edu/archives/win2022/entries/induction-problem/ [Accessed 22 May 2024].
[8] Ashby, W.R., 1957. *An Introduction to Cybernetics*. 2nd impression. London: Chapman & Hall, p. 141.
[9] Jones, N.D., 1997. *Computability and Complexity: From a Programming Perspective*. Cambridge, MA: MIT Press, p. 227.
[10] Sipser, M., 2006. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: PWS Publishing, p. 220.
[11] Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, p. 3602.
[12] Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, p. 3603.

Thus, a deterministic, physical explanation can be given for the creation of qualia, that makes an event (e.g., a decision about colour information, sensory information) fundamentally unpredictable from the agent's own subjective point of view, even though it is objectively determined. However, since this event does definitely occur, and the agent is able to know this when it occurs, this means that the agent is able to know information that is in principle impossible to predict correctly in advance, given only an accurate knowledge of physical laws and initial conditions. Despite the unpredictability, the predicted states of sensation do occur, and in this case their occurrences - at the moment when they occur – have a surprising power. We can then say that the immediate experience of the state, when it occurs for the agent, will certainly be surprising, exceeding the agent's previous knowledge – exactly as it is described in Jackson's knowledge argument.

In which case can a physical computer agent (hereafter agent) have unpredictable sensory information?

1. If the system that can compute the sensory information is physical.
2. If an agent capable of processing sensory information is part of a physical system *S* that is capable of generating predictions about sensory information using hypotheses and is able to answer decision problems about them, such as *"Will the image produced by the filter acting on camera image U(q) be yellow?"* or *"Will image U(q) be modified by filter n?"*
3. If a given piece of sensory information (e.g. colour information) cannot be correctly predicted by the agent, even if the prediction is done using finite, empirical measurement constraints.

The system performing the physical computations

Since it is difficult to define precisely why and when a system, especially a computational system, is physical, we can define such a system by assuming that the physical Church-Turing thesis is satisfied. The formulation of the physical Church-Turing thesis by David Deutsch is as follows[13]: "*Any finitely realisable physical system can be perfectly simulated by a finitely realisable universal computer."*

[13] Deutsch, D., 1985. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society*, 400(1818), p. 101.

In its traditional form, the Church-Turing thesis is an intuitive conjecture[14]. According to this conjecture, a function is algorithmically computable if and only if it is computable by a Turing machine. The thesis establishes a link between the formal definition of the Turing machine and the informal definition of algorithmic computability, and is itself an informal thesis, impossible to prove formally. The physical version of the thesis claims more than that. It can even be interpreted as a potentially falsifiable scientific theory: in physical reality, there are no computers – hypercomputers[15] – whose capabilities extend beyond the computational capabilities of Turing machines.

The assumption of the physical Church-Turing thesis ensures that the physical decision process can be simulated on a Turing machine. And if it can be simulated on a Turing machine, then the physical systems under analysis can only perform computations that can be recursively computed by Turing machines. Thus all the limitations of Turing machines will apply to physical systems. In this way, if a certain program, namely the halting program, cannot be computed by Turing machines[16], then there cannot be a physical computer system that can compute it. The usefulness of the thesis for the line of thought presented in this paper is that, even if the physical Church-Turing thesis is accepted as a strict metaphysical premise, there may exist sentient information for a computer system that is not predictable.

It must be emphasised that in order to imagine the computational system in question as physical and not abstract, it must apply empirical measurement constraints on halting behaviour, so it must be able to set constraints such as a time boundary to measure if a computation halts or not. Then, it has to be analysed if the same type of unpredictability also holds for such a finitely constrained system.

The model of system *S* computing decisions about sensory information

This chapter presents a formal model, based on physical computation, i.e. a physicalist and functionalist model called *system S* from the point of view of philosophy of mind. The general

[14] Copeland, B.J., 2020. The Church-Turing Thesis. In: E.N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Summer 2020 Edition)*. [online] Available at: https://plato.stanford.edu/archives/sum2020/entries/church-turing/ [Accessed 22 May 2024].
[15] Copeland, B.J., 2020. The Church-Turing Thesis. In: E.N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Summer 2020 Edition)*. Section 5.3.3. [online] Available at: https://plato.stanford.edu/archives/sum2020/entries/church-turing/ [Accessed 22 May 2024].
[16] Sipser, M., 2006. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: PWS Publishing. pp. 173-182.

model builds on the basic concepts of computer science[17] to define a decision problem and agent *M* that can in principle solve the decision problem. In computational theory, a problem is usually defined as the question of deciding whether a given string is a member of some language[18]. The present formal model, in contrast, aims to give the problem a semantics, i.e. to interpret a concrete problem of evaluating colour information as a decision problem computable by a Turing machine. A formal description of the general model is given below, with a textual explanation of the concepts and definitions in the description of the specific problem interpreted in the model:

Variables:

n: the variable representing the n-th Turing machine in an enumeration of Turing machines

x: a variable representing any string, symbol or sequence of symbols in the L_n string

y: a variable representing any element of the formal language of the domain *U*

(io1): variable representing a single input-output pair for L_n

(io2): variable representing an interpreted single input-output pair for L_n

(io3): a variable representing a single initial condition-outcome pair for *UP*

U:

Problem domain.

UP:

Decision problem with specific conditions, a question to be decided. The question is related to problem domain *U*.

L_n*:* A Turing machine, where L_n is the *n-th* Turing machine in a certain enumeration.

L_n *_io1:*

L_n A Turing machine with input-output pairs, where L_n is the *n-th* Turing machine in a certain enumeration.

[17] Hopcroft, J.E., Motwani, R. and Ullman, J.D., 2006. *Introduction to Automata Theory, Languages, and Computation*. Upper Saddle River, NJ: Prentice Hall, pp. 1-36.

[18] Hopcroft, J.E., Motwani, R. and Ullman, J.D., 2006. *Introduction to Automata Theory, Languages, and Computation*. Upper Saddle River, NJ: Prentice Hall, p. 31.

L_n_io2:

This is the nth L_n Turing machine with input-output pairs, which M interprets as the hypothetical solution of *UP*. In other words, L_n is a given hypothesis for a hypothetical solution to the *UP* problem, generated by interpreting the input and output variables of the Turing machine L_n.

INTP(L_n):

INTP(x): For every expression x in the physical tape-alphabet of computer *L*, *INTP(x)* yields an element y in the domain *U*. Specifically, for the uninterpreted input symbols of the alphabet of *L*, *INTP()* assigns a set of initial conditions in *U*. For the uninterpreted output symbols of the alphabet, *INTP()* assigns a set of results in *U*. *INTP(L_io1 → L_io2)*, in other words it follows from the definition of *INTP* that a computation *L* is now interpreted as computing a certain problem *UP*.

M:

M is an intelligent agent capable of theory generation and theory selection. The problem domain of *M is U*. The problem *UP* is restricted *to U* and *M*. *M* can generate a number of hypotheses L_n and then eliminate any false L_n based on a truth function.

TRUE(L_n , UP):

L_n *_io2 IFF UP_io3* each*(io2)* and *(io3)* pair, for a given L_n

Sensory information is the output value of an image input calculated by a computer program. Information processing must always be involved in the digitization of physical information[19]. The operation of a CMOS sensor in a web camera cannot be achieved without signal processing[20]. However, these signal processing programs are typically simple, meaning that their operation is completely known by knowing the program code, and the program code

[19] Chakravorty, P., 2018. What is a Signal? [Lecture Notes]. *IEEE Signal Processing Magazine*, 35(5), pp.175-177.
[20] Fossum, E.R. and Hondongwa, D.B., 2014. A Review of the Pinned Photodiode for CCD and CMOS Image Sensors. *IEEE Journal of the Electron Devices Society*, 2(3), pp.33-43.

cannot be changed by the computer system[21]. There is, however, a class of these programs that can be programmed in a Turing-complete language. Photoshop filters, for example, are typically such programs[22]. If the programming language of these filters is Turing-complete, and indeed any algorithm can be written for the filters, then the filters correspond to a Turing machine.

Every computational task related to a transformation (e.g., applying a filter to an image) implicitly corresponds to a computational task representing a decision problem, even if this decision problem is not explicitly expressed. The transformation computational task in this case may be a colour-value transformation by the filter (e.g. yellow filter). The computer calculates what a given image will look like after the filter has been applied. The output is a colour-related value (*"image q will be yellow"*). However, in addition to the transformation task, there is also an implicit decision task: predicting whether the result of applying the filter satisfies a certain condition or criterion (e.g. *"Will image q be yellow?"*). This secondary task is a decision problem where the output is no longer a colour, but a "yes" (*1*) or "no" (*0*). In the example, if the filter is consistently yellow, the implicit secondary computation is always "yes" (the image will be yellow), which is a *trivial* decision problem. In more complex cases, where the outcome of the primary task is not simply determined, the implicit secondary prediction task becomes meaningful and complex.

As a concretization of the general model, the specific decision problem UP1 can then be formulated: *"is the filter applied to image q the same as the filter* X *already applied to camera image U?*"

Thus *U(q)* is the universe of camera images *q* that already have a certain filter *X* applied. L_n contains the program of a given filter, which is the n-th filter in an arbitrary enumeration. *M* is an agent capable of generating and selecting theories about filters. The decision problem thus poses the question whether the computational process *X* that would otherwise take place in any case, where the program of some element *X* of the process is unknown, is predictable for agent *M*. Formally, the decision problem *UP1* can be expressed in the simplest way, with a single input, as follows: *UP1(q)***:** *"Will the image U(q) be yellow (or any other colour)?"*

[21] Barkalov, A., Titarenko, L. and Mazurkiewicz, M., 2019. *Foundations of Embedded Systems*. Cham: Springer International Publishing, p. 195.
[22] Knoll, T., et al., 2003. *Adobe Photoshop Application Programming Interface Guide*. Version CS ed. Adobe Systems, pp.19-20. Retrieved November 28, 2019, via UserManual.wiki.

The instances (unique inputs) of the problem and the inputs of the program L_n to solve the problem are unique q images. A specific L_n is thus a hypothetical solution to the problem, which computes in advance whether applying a filter to q images will result in an image of the specified colour. For each image, L_n evaluates *UP1(q)* to correctly predict the colour information decision. This will be a hypothetical answer to the question posed by *UP1*. Both the problem and the output of the Turing machine that computes the output of *UP1* are logical binary values: *0* or *1*, which can be false or true, but can also be undefined.

Key concepts

Sensory information, colour information: output data of the camera image *U(q)* that appears as input to agent *M* after further computation. In this example, it is equivalent to the information content of an RGB image. If the program of L_n is generic and the complexity of the filter program is increased, the sensing information can be not only colour information but also any other sensing information. In this case, the *'sense'* may correspond to the input of any peripheral of *M*, or even to any input of *M* that is computed as the output of programs for *M*. Given the appropriate complexity of L_n, they may co-occur and the sensory information may interact.

Detection process: the process of calculating the output value of *U(q)* from a set of q images. In the case of the system, this can be done directly, or it can be done by prediction, or precomputation, generated by the agent.

Decision value related to sensory information, colour information: an output value of *UP1(q)*. The output of *UP1(q)* is a logical value based on the colour result of applying the filter to image *q*. Thus defined, the output of *UP1(q)* is no longer just colour information, but is now a yes/no answer to a question about colour information.

Prediction, forecast, decision: when *M* generates the hypothesis L_n and then computes the interpreted output of L_n with at least one input *q*, then L_n forms a forecast. Since L_n represents a decision or answer related to a decision problem, L_n also computes a decision related to *UP1* at the same time. The output of the prediction or decision is binary, representing a yes or no answer to the decision problem.

Prediction of a decision on a piece of sensory information: a decision problem. Can the value of *UP1(q)* be computed in advance by a correct program?

Evaluation: Evaluation of $TRUE(L_n , UP)$ for any input.

Definition of system S: Based on the above, *S* is a system capable of computing sensory information in two ways. *S* is primarily a system that computes sensory information from a source U(q) using a program *X*, where the program *X* is unknown to agent *M* (black box), but the result of the computation of *X* is presented to agent *M* as input sensory information. This sensory information and the decision problems associated with the sensory information in *S* are computed by *X* in any case. *S* is also, in a secondary way, a system for predicting *UP* decision problems related to sensory information by means of L_n programs, in which Agent *M* can generate and evaluate L_n programs that hypothetically compute these *UP* problems.

When is a decision on sensory information not predictable?

Type S systems exist. In practice, an image processing system interprets the incoming information, the display of any image presupposes a certain amount of information processing[23]. Therefore, the display of information requires the execution of some computational process.

Self-learning, artificially intelligent agents are capable of modifying, creating and running programs, generating and evaluating hypotheses[24]. One such system is ChatGPT[25], in which the problem of unpredictability can be demonstrated experimentally. If, for example, the AI agent is able to detect that its camera image, which previously behaved in a constant, predictable way, does not display the world in the same way as before under new conditions, but has no direct information about why this happens, the agent can infer that there is some filter between the world and the camera image in the computational process. This filter then acts as a black box for the agent, i.e. it can only deduce what exactly the filter's program is by examining the input-output information pairs.

It follows from the description of the system *S* that the agent *M* creates predictions in order to predict the program of its own existing *Filter X*. These predictions are based on finite pairs of input-output information, so they are in fact generated in the same way as any empirical theory[26].

[23] Chakravorty, P., 2018. What is a Signal? [Lecture Notes]. *IEEE Signal Processing Magazine*, 35(5), p. 176.
[24] Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2012. *Foundations of Machine Learning*. Cambridge, MA: MIT Press.
[25] OpenAI, et al., 2023. GPT-4 Technical Report. *arXiv*. [online] Available at: https://arxiv.org/abs/2303.08774 [Accessed 22 May 2024].
[26] Popper, K., 1959, 2005. *The Logic of Scientific Discovery*. Taylor & Francis e-Library. pp. 37-56.

In such a case, the consequence of the problem of induction is that the program of the filter cannot be uniquely determined by finite observation.

As an interesting example of such unpredictability, we can imagine all possible states of an image from a camera with a resolution of only 320 x 200 pixels and a colour depth of only 1 bit are 2^{64000}. If the number of atoms in the universe is $\sim 10^{82} = \sim 2^{256}$, and we want to physically realise and store these possible states as data, it is understandable why Agent *M* is not able to compute all possible states, i.e. all possible sensory information, in advance and why Agent M has to guess them by some heuristic.

The problem is further complicated if *Filter X* contains one or more continuously varying hidden parameters. A hidden parameter[27] in this case means that the output of the filter depends on some environmental variable that is not observable to the system in a constant way. If, for example, the filter varies as a function of temperature, and the agent is exposed to continuously fluctuating temperature conditions, it will receive sensory information indistinguishable from a random program. Under these conditions, the principle of the filter cannot be learned[28]. As a concrete example, it is also possible to generate a hidden parameter by basing the colour added by the filter on the colour of a particular pixel of the camera image, or even on the average colour of the image.

The general case of the prediction problem of sensory information

For general unpredictability, additional conditions must be met. If both the primary computational processes that are directly involved and the secondary computational processes that predict them are sufficiently complex, a system can be created whose state of ignorance corresponds perfectly to the state Jackson describes for Mary[29], in which something is then missing from her knowledge before she leaves the black and white room.

Filters[30] are usually only a narrow class of all executable programs. However, in order to be undecidable, it is necessary that the class of programs under consideration is equal to the class of all possible programs, i.e., it is a necessary precondition that the program used to process the

[27] Ashby, W.R., 1957. *An Introduction to Cybernetics*. 2nd impression. London: Chapman & Hall, p. 141.
[28] Ashby, W.R., 1957. *An Introduction to Cybernetics*. 2nd impression. London: Chapman & Hall, p. 134.
[29] Jackson, F., 1982. Epiphenomenal qualia. *Philosophical Quarterly*, 32(April), pp.127-136.
[30] Gonzalez, R., 2018. *Digital Image Processing*. New York, NY: Pearson. ISBN 978-0-13-335672-4.

camera signal must be written in a universal, Turing-complete program language. The digital filters and image processing plug-ins used in computers and cameras are exactly such, typically software written in C++.

For example, an undecidable program might be implemented in the following pseudocode program. For the sake of example, suppose that there exists some standard enumeration of filter programs L_n and suppose that there exists a standard enumeration of program inputs i_n.

Filter Program 2 (def):

Get image 'q'.

n = Rnd(n)

Get 'i_n '.

Get 'Ln '.

If Ln (q+i_n) halts, then colour 'q' green.

If Ln (q+i_n) doesn't stop, then colour 'q' red.

In general, the problem of prediction is reduced to the problem of Turing equivalence[31]. Briefly, the Turing equivalence problem is as follows: given two programs *P* and *Q,* do they give the same result for all possible inputs? In theory, we can easily construct a Turing machine *Q* that always stops. Then suppose that we have a program *PEQ that* checks the equivalence of *Q* with another arbitrary program *P* such that *PEQ(P, Q)* is true if and only if *P* stops for all possible inputs. Thus, if we had such a program *PEQ(P, Q),* we could solve the halting problem, which is undecidable. Therefore, *PEQ(P, Q)* is also undecidable.

It must be emphasised that the problem of Turing equivalence is not a physical problem, because in physical reality, no two programs can be *equivalent*. Therefore in real-life application, this problem is better understood as a correspondence problem, especially in the upcoming case, where Agent *M* must utilise empirical measurement constraints on the halting condition of any program. In such a case, two physical programs can be considered *corresponding to each other* if they match each other according to predefined empirical criteria, e.g. they have the same program size or run for the same amount of time.

[31] Sipser, M., 2006. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: PWS Publishing. p. 220.

How is the general case of unpredictability using the halting problem invoked? Interpreting Seth Lloyd's similar conjecture about free will[32], and interpreting the given problem in this way as a decision problem, the undecidability of the halting problem is represented in the system as follows. The n-th decision unit of the system solving the decision problem corresponds to a program L_n computing a decision on sensory information, which receives the information needed to make the decision based on q inputs, and then reaches the decision result *n(q)=(yes)*, *n(q)=(no)*, or no result *(n(q)* undefined). A prediction is thus in fact nothing more than a decision or a decision process, i.e. a computational process that, for any q inputs of a given program Ln , gives a result *n(q)=(yes)*, *n(q)=(no)*, or *n(q)* undefined.

In the case of prediction using a general program, the variable *n in* fact denotes a class of decisive physical units - all such units, i.e. the entire class of these units - which can be recursively enumerated[33]. When can *n* be any program? When *n* programs can process not only *q pieces of* image information, but also any other *q′=(q+i)* pieces of information. When can input *q′* be any? This is possible in several cases. For example, it is possible when any other input *i* can be added to image *q*, i.e. *q′=(q+i)*. But it is also the case when *q = m(i)*, i.e. input image *q* can itself be an image generated by a subroutine *m* from input *i*. Such a general *M* agent, performing any *n* predictions, cannot predict its decision results. Consider the following function f_m computed by the agent's *m* program: $f_m(n,q) = n(q)$ when L_n stops at input *q*, and $f_m(n,q)=F(fail)$ when L_n does not stop at input *q*. Given the undecidability of the halting problem, the result of this $f_m(n,q)$ cannot be computed in general by any *m* programs of agent *M*[34].

The finite, physical case of the prediction problem

The preceding argument uses the general halting problem, which is about Turing machines that may run for any number of steps. Physical systems, however, terminate within some finite time. The question is therefore whether the unpredictability result survives the transition from the idealized, unbounded setting to a physically realistic, time-constrained one. We construct the bounded version of the decision problem by letting *t* define a bound. Consider it as an upper

[32] Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, pp. 3597-3610.
[33] Sipser, M., 2006. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: PWS Publishing. p. 170.
[34] Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, p. 3602.

bound on the number of computation steps a machine may execute and also a bound on its program size.

Every decision system in our enumeration is now truncated accordingly: $L_{n\,t}(q) = L_n(q)$ if L_n produces an output on input q within at most t steps, and $L_{n\,t}(q) = 0$ otherwise. Also, failure to answer by step t counts as 'no'. We can now ask whether there exists a uniform procedure that, for an arbitrary time-bounded decision system $L_{n\,t}$ and an arbitrary input q, determines what verdict the system reached — and does so within the same time bound t. Define $f(L_{n\,t}, q) = L_{n\,t}(q)$. This is the function that looks up the answer: given the index of a decision system and an input, it returns whatever that system decides within time t. f is computable: a Turing machine can simulate L_n on q for t steps and read off the result. However, a diagonal argument demonstrates that it cannot do that within t[35]:

1. Consider the two-dimensional list A_T whose (L_n, q)-th value is $f(L_{n\,t}, q)$. This list contains all programs $L_{n\,t}(q)$ *that* can be computed within time t.
2. Let $f(L_{n\,t}, q) = L_{n\,t}(q)$.
3. Let $g(q) = 0$ if $f(q,q) = 1$, and vice versa. That is, $g(q) = \sim f(q,q)$.
4. If f can be calculated within time t, *so* can g.
5. But if g can be computed in time t, then $g(g)$ is necessarily equal to $f(g,g)$ (2).
6. And this is a contradiction, since $g(g)$ *is* defined to be equal to $\sim f(g,g)$ (3).

Since the assumption that f is computable within time t leads to a contradiction, we must reject it: neither f nor g can be computed in time t. A predictor can eventually work out what every t-bounded decision system decided, but only by expending computational resources that exceed the original time bound. This result bears directly on the prediction question that motivates the entire section. The system's original question was: "will the decision system reach a verdict within time t, and if so, what verdict?" The diagonalization argument shows that no uniform procedure can answer this question within time t itself. Any general method — regardless of how it is designed — must exceed the time bound t in order to determine what t-bounded decision systems decide. Prediction, when directed at an entire class of decision makers operating under a shared resource constraint, is provably more expensive than the decisions it seeks to anticipate. f is a function *about* the set that is nevertheless *outside* the set: f

[35] Mimics the argument put forward in Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, pp.3602-3603.

cannot be evaluated within the time bound t, and if we expect f to be total decidable, it cannot exist within the set of all functions A_T, which by construction contains only those functions computable within t steps. Self-reference is therefore blocked — not by some artificial restriction or by fiat, but by the time-complexity gap itself. The function $f_t(L_{n\prime}, q)$ cannot answer questions about its own behaviour, nor can it serve as a representative of the broader class of t-bounded computations to which it nominally belongs.

The conclusion is that t-finite physical computational systems cannot predict the totality of output values of t-finite physical computational systems. The unpredictability that appeared in the general, unbounded case is not an artefact of idealization. It persists under exactly the kind of finite, empirically constrained conditions that characterize real physical computation.

Since only finite systems with measurement constraints on halting behaviour can be considered as real physical systems, it is reasonable to ask why it is important to argue for the general case. The reason is that Frank Jackson's original thought experiment asked what Mary could or could not know in case she had perfect knowledge of all physical facts and all physical theories. If only the finite case had been argued for, then a non-physicalist defender of qualia could appeal that the finite case of the prediction problem with measurement constraints does not operate with the perfect knowledge of *all* physical facts.

Demonstrating unpredictability with a thought experiment

Consider the computing physical agent described above, let us call it Mary-demon after Jackson's Mary. Let us consider Mary-demon's knowledge and abilities to be as powerful as those of Laplace's demon[36], with the important remark that Mary-demon is also a physical entity, i.e. it is *itself part of the physical universe*[37]. Suppose that Mary-demon not only knows all physical facts but is also capable of prediction along all *possible* scientific theories, i.e. capable of simulating all possible decision processes along any possible physical facts. Consider all possible brain-state configurations of Mary-demon, including all the facts of the world, as

[36] Laplace, P., 2009. *Essai Philosophique sur les Probabilités*. 5th ed. Cambridge: Cambridge University Press.
[37] Popper, K., 1992. *The Open Universe: An Argument for Indeterminism From the Postscript to the Logic of Scientific Discovery*. London: Routledge. p. 29.

simulatable on a Turing machine. Even this system would then be incapable of predicting all future states, given the undecidability of the halting problem[38].

Mary-demon's knowledge only covers the decisions that can ever be physically realised. It is important to note that, under these conditions, she does not necessarily realise every single Turing machine, but only some narrow class of them. Since the undecidability of the halting problem requires that, in the general case, the class of Turing machines in the problem includes all Turing machines, this alone would not necessarily make Mary-demon unable to predict her own sensory information. Then the condition that the demon itself is part of the physical universe becomes important. If this is so, then - considering that the physical universe has a finite bound - we are not dealing with the general case of the halting problem, but rather with the finite case described above. The argument then describes a finite physical process that can be used to explain the knowledge gap in Jackson's thought experiment above. Despite the indeterminacy caused by the undecidability of the halting problem, the unpredictable computations of the system do in fact occur, in which case they are unexpected and surprising at the moment they occur and the perceiving physical agent learns that they constitute new - but entirely physical - knowledge. If we identify this state with the appearance of a quale, we get a physicalist, mechanistic, non-intuitive explanation of qualia.

The implications of the unpredictability are thus also related to the philosophy of mind's views on quals. According to Dennett[39], the properties of qualia are privacy, infallibility, inexpressibility, and intrinsicity. All of these can be interpreted in the above model. Predictable sensory information cannot be unique, for if it can be predicted, it can be predicted more than once, and moreover, predictable information can be predicted for other systems, provided that predictability presupposes complete knowledge of the program code and its outputs. If the program code is known, it can be copied, in which case the uniqueness of the outputs is lost. In contrast, sensory information that is not predictable, but is nevertheless displayed, can be considered unique. Because of its uniqueness, this information is both completely private for *M* and subjective[40], in that it is computed exactly (but not in advance) only for *M*. For the reasons

[38] Turing, A.M., 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, pp.230-265 [Erratum in Proceedings of the London Mathematical Society (1937) 43, p.544-546].
[39] Dennett, D.C., 1988. Quining Qualia. In: A.J. Marcel and E. Bisiach, eds., *Consciousness in Contemporary Science*. Oxford: Oxford University Press.
[40] Nagel, T., 1974. What is it Like to be a Bat?. *Philosophical Review*, 83(October), pp. 435-450.

given above, sensory information is also inexpressible in the sense that the program required for its occurrence cannot be determined, nor can it be passed on to any other system. If the program of *filter X* is not directly known, then the program of the entire system *S* containing *filter X* is not exactly known. An external system *K* would then be unable to predict the state of *S* (and *M*) for exactly the same reasons that *M* is unable to accurately predict the program of *filter X*. Because of the induction problem and the Turing equivalence problem, system *K* cannot compute what *M* computes at the moment it computes the output of the filter program. *M* is also infallible about the appearance of sensory information[41]. Since *M* is the last member of an information processing chain, it cannot be mistaken in the sense that when information appears to it, it already has the modifications it has already undergone. Intrinsicity is the consequence of the fact that the unpredictability does not depend on external factors, but is simply a consequence of the system's own structure.

In an epistemological sense, it misses from *M*'s knowledge ─ since it cannot predict an event, hence it does not know the explanation of the occurrence of the event ─, but this does not necessarily mean that it misses from *M*'s knowledge of the physical world in a metaphysical sense ─ since this predictability would occur even if *M* really knew everything about the functioning of a determinate physical universe, and the universe really did contain only physical matter. To quote Joseph Levine[42], it cannot be metaphysically excluded that pain (which is a sensory information) is equivalent to the occurrence of some physical process (the firing of C-fibres, or the computation of a filter by a computer that will occur anyway), but if it is, it is a brute, inexplicable fact for agent *M*.

The sensory information presented to *M* is determined by the system *S*, but in principle is unpredictable from the state of the agent *M*. Thus, this unpredictability reinforces the illusion of independence of higher and lower levels (i.e., consciousness experiences unique to *M* and correlated determinate physical brain states), i.e., dualistic intuition.

References

Ashby, W.R., 1957. *An Introduction to Cybernetics*. 2nd impression. London: Chapman & Hall, pp. 86-93.

[41] Dennett, D.C., 1988. Quining Qualia. In: A.J. Marcel and E. Bisiach, eds., *Consciousness in Contemporary Science*. Oxford: Oxford University Press.
[42] Levine, J., 1983. Materialism and qualia: the explanatory gap. *Pacific Philosophical Quarterly*, 64, pp.354-361.

Barkalov, A., Titarenko, L. and Mazurkiewicz, M., 2019. *Foundations of Embedded Systems*. Cham: Springer International Publishing, p. 195.

Chakravorty, P., 2018. What is a Signal? [Lecture Notes]. *IEEE Signal Processing Magazine*, 35(5), pp.175-177.

Copeland, B.J., 2020. The Church-Turing Thesis. In: E.N. Zalta, ed., *The Stanford Encyclopedia of Philosophy (Summer 2020 Edition)*. [online] Available at: https://plato.stanford.edu/archives/sum2020/entries/church-turing/ [Accessed 22 May 2024].

Dennett, D.C., 1988. Quining Qualia. In: A.J. Marcel and E. Bisiach, eds., *Consciousness in Contemporary Science*. Oxford: Oxford University Press.

Deutsch, D., 1985. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society*, 400(1818), pp.97-117.

Fossum, E.R. and Hondongwa, D.B., 2014. A Review of the Pinned Photodiode for CCD and CMOS Image Sensors. *IEEE Journal of the Electron Devices Society*, 2(3), pp.33-43.

Gonzalez, R., 2018. *Digital Image Processing*. New York, NY: Pearson. ISBN 978-0-13-335672-4.

Hartmanis, J. and Stearns, R.E., 1965. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117, pp.285-306.

Henderson, L., 2022. The Problem of Induction. In: E.N. Zalta and U. Nodelman, eds., *The Stanford Encyclopedia of Philosophy (Winter 2022 Edition)*. [online] Available at: https://plato.stanford.edu/archives/win2022/entries/induction-problem/ [Accessed 22 May 2024].

Hopcroft, J.E., Motwani, R. and Ullman, J.D., 2006. *Introduction to Automata Theory, Languages, and Computation*. Upper Saddle River, NJ: Prentice Hall, p. 31.

Jackson, F., 1982. Epiphenomenal qualia. *Philosophical Quarterly*, 32(April), pp.127-136.

Jackson, F., 2007. Epiphenomenal qualia. Translated by Á. Polgárdi. *Difference*, IX(1), pp.67-82.

Jackson, F., 2008. What Mary didn't know. In: A. Gergely, T. Demeter, G. Forrai and J. Tőzsér, eds., *Collection of Texts on Philosophy of Mind*. Budapest: L'Harmattan.

Jones, N.D., 1997. *Computability and Complexity: From a Programming Perspective*. Cambridge, MA: MIT Press, p. 227.

Knoll, T., et al., 2003. *Adobe Photoshop Application Programming Interface Guide*. Version CS ed. Adobe Systems, pp.19-20. Retrieved November 28, 2019, via UserManual.wiki.

Laplace, P., 2009. *Essai Philosophique sur les Probabilités*. 5th ed. Cambridge: Cambridge University Press.

Levine, J., 1983. Materialism and qualia: the explanatory gap. *Pacific Philosophical Quarterly*, 64, pp.354-361.

Lloyd, S., 2012. A Turing Test for Free Will. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370, pp.3597-3610.

Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2012. *Foundations of Machine Learning*. Cambridge, MA: MIT Press.

Nagel, T., 1974. What is it Like to be a Bat?. *Philosophical Review*, 83(October), pp.435-450.

OpenAI, et al., 2023. GPT-4 Technical Report. *arXiv*. [online] Available at: https://arxiv.org/abs/2303.08774 [Accessed 22 May 2024].

Popper, K., 1950. Indeterminism in classical and quantum physics. *British Journal for the Philosophy of Science*, 1, pp.117-133, 173-195.

Popper, K., 1992. *The Open Universe: An Argument for Indeterminism From the Postscript to the Logic of Scientific Discovery*. London: Routledge.

Popper, K., 1959, 2005. *The Logic of Scientific Discovery*. Taylor & Francis e-Library.

Sipser, M., 2006. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: PWS Publishing.

Sostai Zoltan, 2023. Analysis of the physical Church-Turing thesis and some philosophical implications of the halting problem. In: M. Nemes, ed., *Impact Points IX. Proceedings of the Conference of the Philosophy Department of the National Association of Doctoral Students*. Budapest: National Association of Doctoral Students, pp. 145-158.

Turing, A.M., 1936. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, pp.230-265 [Erratum in Proceedings of the London Mathematical Society (1937) 43, p.544-546].