Ambiguous encryption implies that consciousness cannot be simulated

Anna Wegloop and Peter Vach

Contents

Summary

Here we show, based on a simplified version of fully homomorphic encryption, that it is not possible to simulate conscious experience, in the sense of using a computer algorithm to generate experiences that are indistinguishable from those of a particular typical human being. This seems to have important implications for questions in the context of future developments in artificial intelligence. For example, the proposed process of mind-uploading will in general not generate a virtual consciousness similar to the consciousness of the original human being.

Introduction

It seems possible that continuing technological and scientific advances will allow simulations of human beings that appear (on the outside) exactly like the original human being. One example is a brain emulation (Kurzweil, 2010), where the physical processes (electrophysiology, distributions of neurotransmitters, etc.) within a particular brain are being simulated. Another example could be an artificial neural network (or some other functional model), which mimics the behavior of a person without modeling the physical processes within a human brain in detail. However, it remains unclear whether or not such simulations would produce conscious experiences (on the inside). This question has been addressed before, for example with the Chinese room argument (Searle, 1980). However, the philosophical debate surrounding this question does not seem to be settled. There is no generally agreed definition of consciousness. Difficulties arise from its inherently subjective quality (Nagel, 1974) and many open questions remain regarding its origin and its nature.

Here we approach this question without sophisticated philosophical concepts, but relying on widely shared methods of information theory and logical inference. We show that if a simulated mind has conscious experiences (on the inside), these are not equivalent and generally not even similar to those a real human being would have under similar circumstances. In this sense the proposed concept of mind-uploading (also called mind emulation, digital immortality) is shown to be impossible (Kurzweil, 2010; Bostrom & Sandberg, 2008).

Definitions

Symbols used:

Algorithms: M, M^{-1}, P, E(P), R, S (computer algorithms), H, K (algorithms including a human being) and F (arbitrary physical process)

Text files: A, A*, B, B*, C, C*, D, D*

Encryption keys (numbers): p1, p2

Experiences: α, β, γ, δ, ε

Assumptions: Ψ, Σ, Φ

Consciousness:

With consciousness we refer to conscious experiences (qualia). One can only check for oneself that one has such experiences. In some sense this is a minimal definition of consciousness. We are not concerned with more elaborate concepts such as “soul” or “self”, built on top of the perceived reality of conscious experiences. Nor are we here concerned with any theory or model of consciousness (e.g. integrated information theory (Oizumi, Albantakis, & Tononi, 2014)). More generally, we are not concerned with any objectively measurable quantity supposedly indicating the presence or absence of consciousness (e.g. neural correlates of consciousness).
Using our definition of consciousness, detecting consciousness scientifically is a difficult endeavor: There is only one empirical data point (I know that I am conscious). A theory of consciousness that claims that everything is conscious seems to correctly predict all experimentally verifiable facts about the presence or absence of consciousness, since only claims about the absence of consciousness can be experimentally refuted by individual experience. This should be taken into consideration when

trying to empirically test any theory of consciousness. In keeping with this minimal definition of consciousness, the argument presented below is constructed to rest simply on the (subjectively) perceived reality of consciousness, widely shared information theory and the assumption that logical reasoning can be used to derive true statements about questions involving consciousness.

Simulating human behavior:

Imagine the following scenario: A human being is sitting in a room in front of a video screen, equipped with speakers, a microphone and a camera. The sounds and video images played back on the screen and the speakers are specified by file A (input data). The recordings by camera and microphone result in file A* (output data). We can also think of the physical system (the room, the electronic equipment and the human being) as an algorithm that connects A and A*, which we denote as H. Let P be a computer algorithm with the following property: given input A, P generates an output A* that is indistinguishable from the output A* generated by H given input A.

The argument we present below does not depend on the specifics of the setup. P may or may not be based on a brain emulation and our argument would work also if P were a simulation of any other observable physical quantities of a human being (e.g. based on EEG or fMRI data instead of video recordings).

Given the same input, the two algorithms P and H produce indistinguishable output files. We mean with indistinguishable that the sets of output files for H and P are sufficiently similar according to some predefined metric. This allows us, for example, to account for potential randomness in a straightforward way. Then A* would correspond to all possible output files, together with a probability distribution over these output files. If these probability distributions are sufficiently similar for P and H (according to the predefined metric), we regard their outputs as indistinguishable. However, the argument presented below, does not depend on these complications. Thus, we can regard A and A* simply as text files. H and P output identical text files A* given input A.

We note that such an algorithm P would surely pass the famous Turing test (Turing, 1950). We note further, that algorithm H will lead to conscious experiences, since a human being is part of H (assuming other human beings have conscious experiences too). By contrast, it is not clear what kind of conscious experiences (if any) result from running algorithm P.

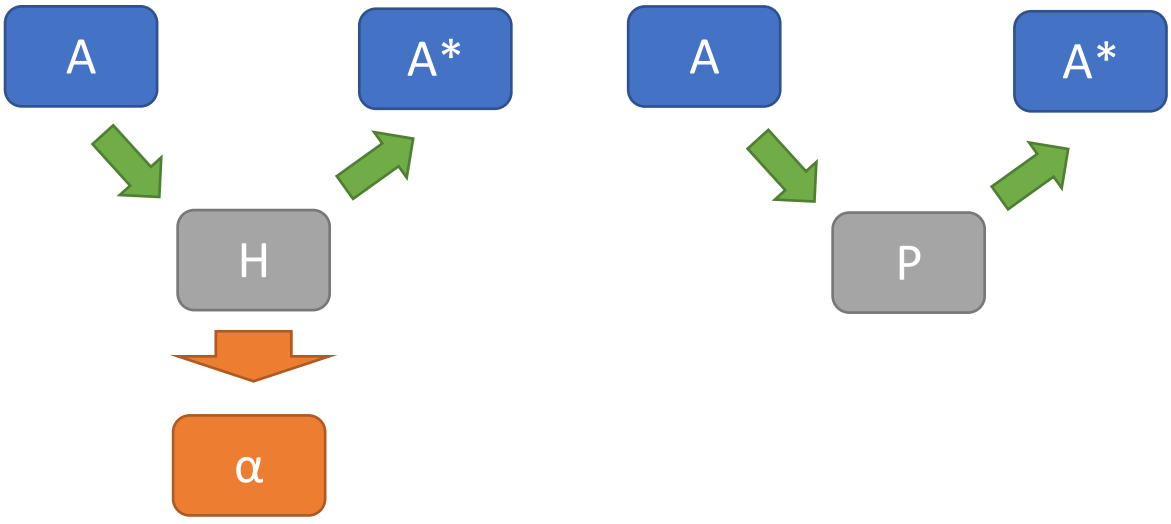

Figure 1: Illustration of a simulation of human behavior. In all figures we use the following color scheme: blue = text files; green = digital input or output; grey = material systems/computer

algorithms; orange = conscious experiences (might be none), arrows signifying the association of experiences with physical processes (e.g. running computer algorithms).

Simulating consciousness:

As noted above, it is unclear if running a simulation on a computer leads to conscious experiences. We therefore define ε to be any conscious experience resulting from running algorithm P with some input A, or no conscious experience at all. Note that ε could also be multiple experiences in sequence, in parallel (e.g. simulating the behavior of more than one human being) or nested (e.g. simulating a human being who is evaluating P by hand). Since P and H produce identical output when given identical input, the simulated behavior (P) appears on the outside exactly like the actual human being (H). For this reason, one might assume that the simulated behavior (P) and the human being have indistinguishable experiences (α = ε). This situation (α = ε) is what we define as simulating conscious experiences. We define two experiences as indistinguishable if it isn't possible to subjectively detect a difference between the two experiences (e.g. eating 100 g of chocolate versus 99 g of chocolate).

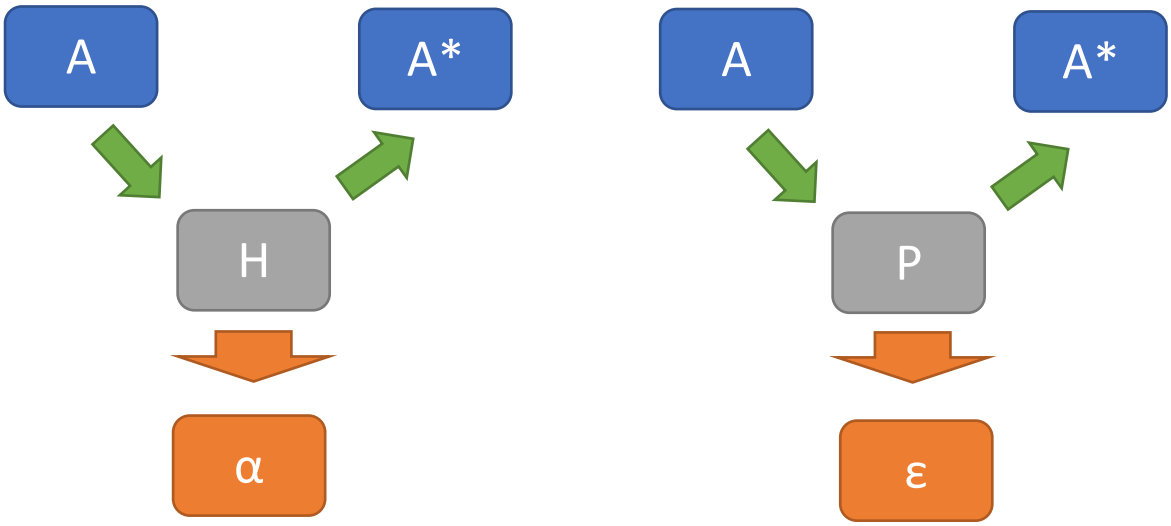

Figure 2: Illustration of experiences being associated with algorithms. ε may or may not be a conscious experience. For simulating conscious experience, α = ε (the two experiences being indistinguishable) is the defining property.

Fully homomorphic encryption (FHE):

FHE is a form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext. It has been shown that FHE is possible for computer algorithms (Gentry, 2010). Encryption (M) and decryption (M^{-1}) both transform a text into another text. M^{-1} is the inverse of M in the sense that a text (plaintext) encrypted by M results in a ciphertext, which can be fed into M^{-1} to yield the plaintext (decryption).

The fully homomorphic encryption scheme is depicted in the following figure:

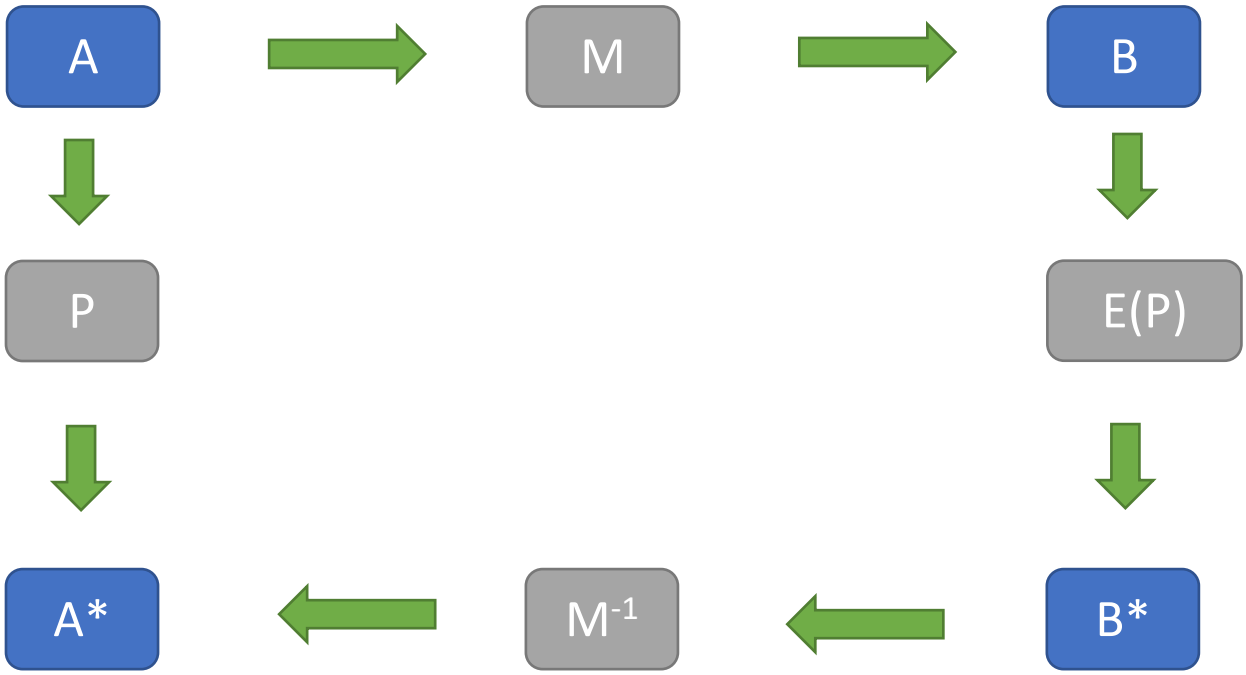

Figure 3: Graphical representation of a fully homomorphic encryption scheme.

In FHE the encrypted input B is not fed into the algorithm P directly, but into another algorithm E(P) that is constructed to evaluate P. E(P) is an algorithm that can be constructed from P, so that the homomorphic property (see figure 3) is fulfilled (for particular encryption and decryption algorithms M and M^{-1}). Importantly, E(P) is constructed in such a way as to be independent of the particular encryption key being used (see appendix B for details).

Ambiguous encryption:

For the argumentation in this article, we introduce a simple fully homomorphic encryption scheme with symmetric keys in appendix A, based on a fully homomorphic encryption scheme described by Gentry (Gentry, 2010). Due to our simplifications, our scheme is not cryptographically secure, but it is fully homomorphic (i.e. the relationship depicted in figure 3 holds).

As we show in appendix A, it is possible for a single ciphertext B to encrypt two different plaintexts A and C, depending on which encryption key is used. For any two A and C, a B can be constructed that has this property.

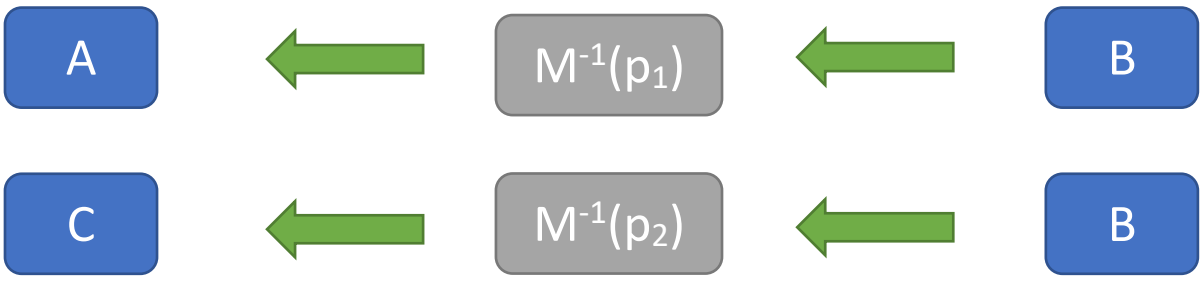

Figure 4: Illustration of ambiguous encryption.

Argumentation

We assume that it is possible to simulate a conscious experience (assumption Ψ): If P (given any input A) produces an output A* that is indistinguishable from the output of H (given input A), P will produce a conscious experience ε that is indistinguishable from the experience α that H produces (see figure 2). Stated informally, assumption Ψ is equivalent to the believe that two entities that behave in the same way, as observed from the outside, also have the same conscious experiences on the inside.

We will now show that assumption Ψ leads to a logical contradiction and is therefore invalid.

Using Ψ we get α = ε and therefore:

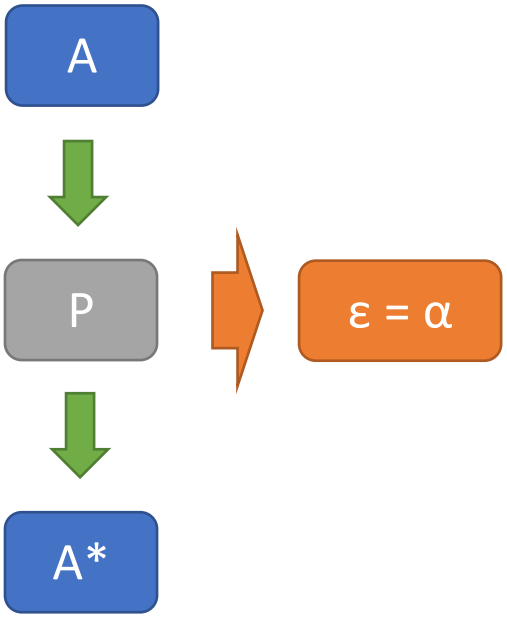

Figure 5: Illustration of assumption Ψ.

Now we feed P a substantially different input C.

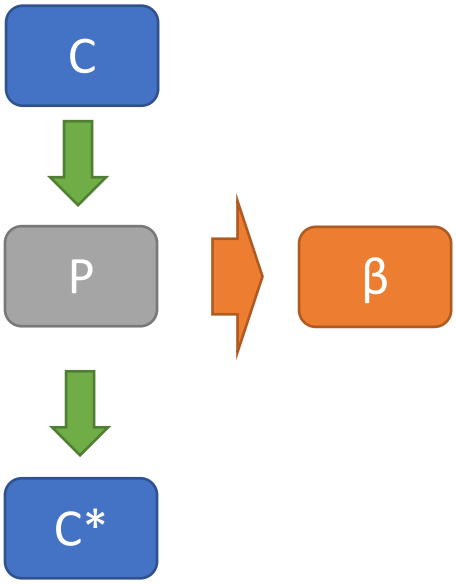

Figure 6: Under assumption Ψ, running P with input C will produce an experience denoted as β.

Now we note that the experience β will in general not be the same as experience α. For example, A might correspond to a video of a person asking "What is your name?". By contrast, C might correspond to a video of a cat playing with yarn. The experiences that the human being, who is part of H, is having, will be very different when input C is used rather than input A. Thus, since we simulate a typical human being, we have $\alpha \neq \beta$. We emphasize that α and β are not simply not identical, but substantially different. While it is difficult to quantify differences in conscious experiences, it is clear that the setup described above (H) allows for a wide range of experiences to be had, at least on a human scale of experiences, and therefore it is possible to pick a situation such that C produces a substantially different experience than A does when either is used as input for P.

Now we can use FHE together with ambiguous encryption to construct the following situation.

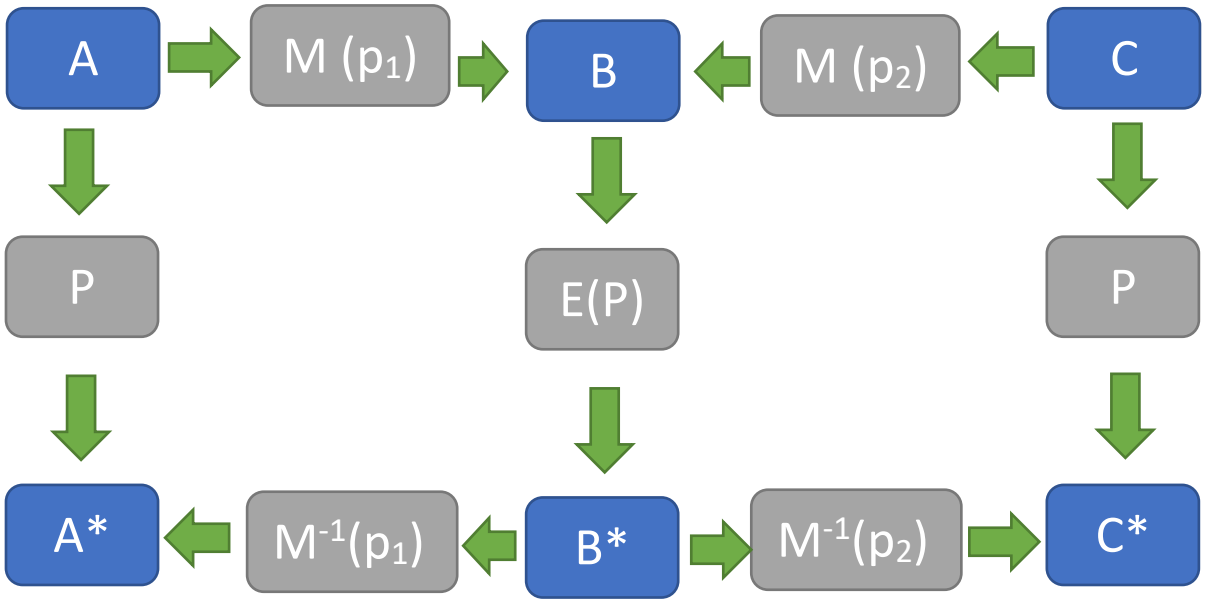

Figure 7: Fully homomorphic encryption with ambiguous encryption.

Looking at figure 7, we can define R as the algorithm that takes A, applies $M(p_1)$, then feeds it into E(P) and then uses $M^{-1}(p_1)$ on the output of E(P).

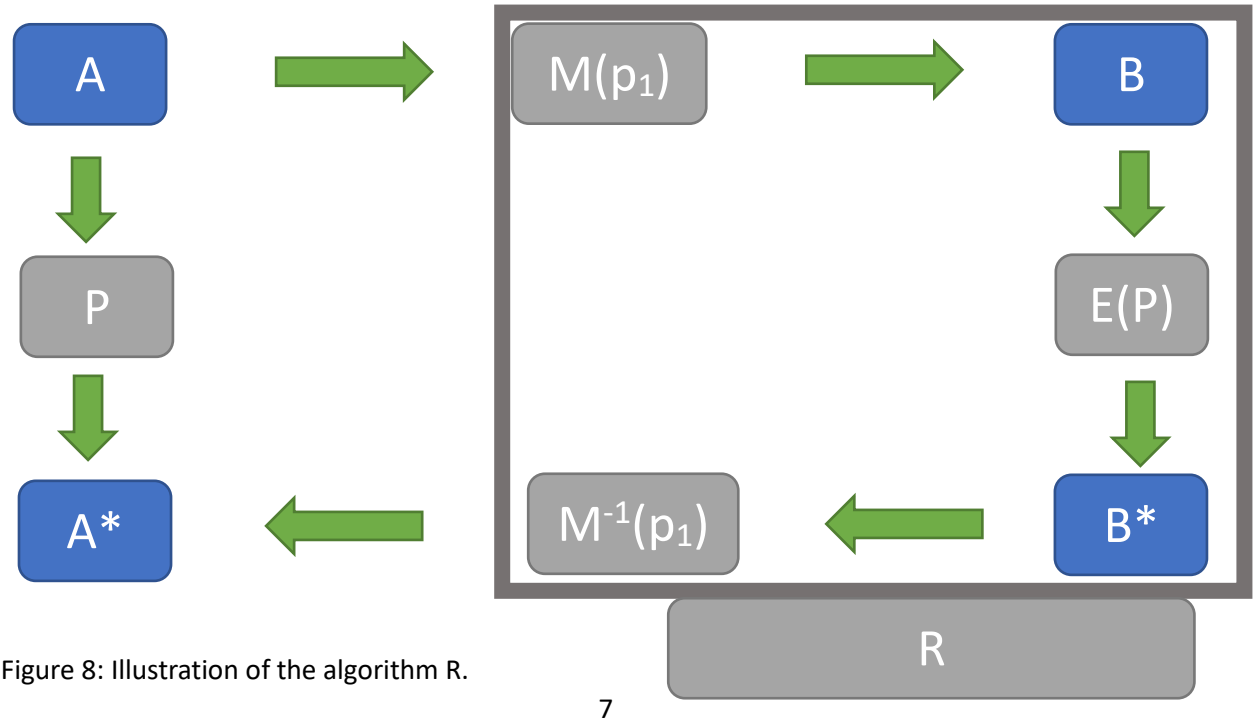

Figure 8: Illustration of the algorithm R.

Running algorithm R will produce a conscious experience (or none at all), which we denote as γ.

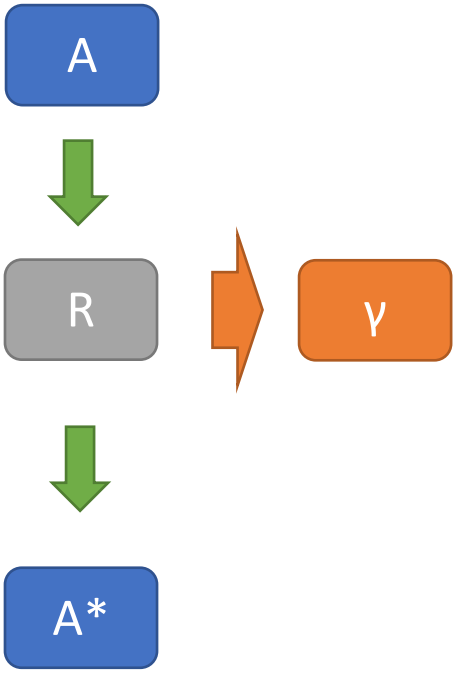

Figure 9: Running R with input A leads to the experience γ (which might also be a non-experience).

Using Ψ we would conclude that γ = α.

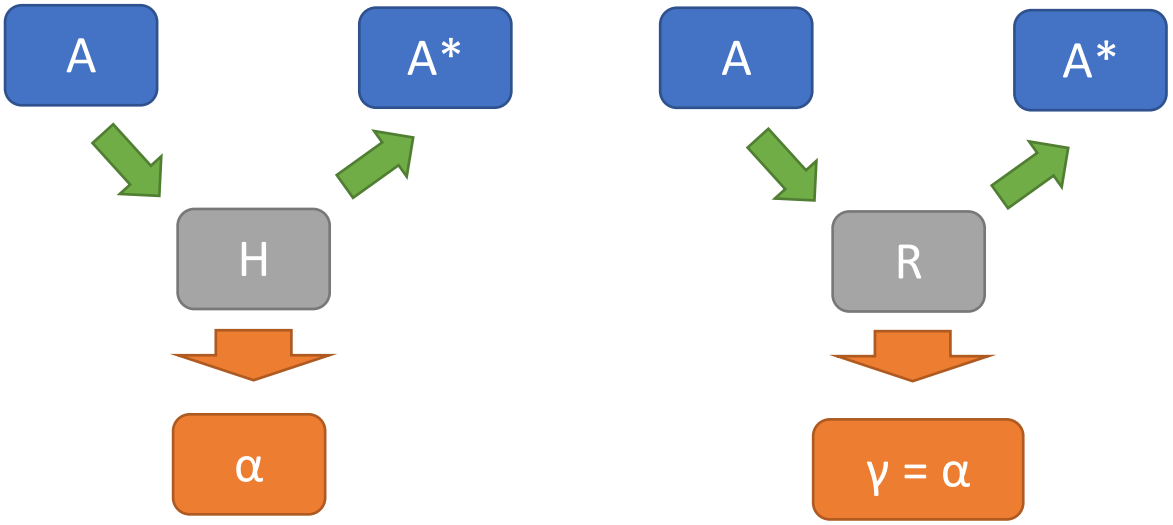

Figure 10: Based on assumption Ψ one would conclude that γ = α.

Now we will consider two cases (discarding the possibility of future events influencing conscious experiences in the past). First, we assume (assumption Σ) that any experience (here γ) generated by running an algorithm depends only on the input (here B) into the algorithm (here E(P)) and the

algorithm itself. That is, we assume that the process of encryption (using M(p) to turn A into B) does not influence or contribute to the experience being generated (if any).

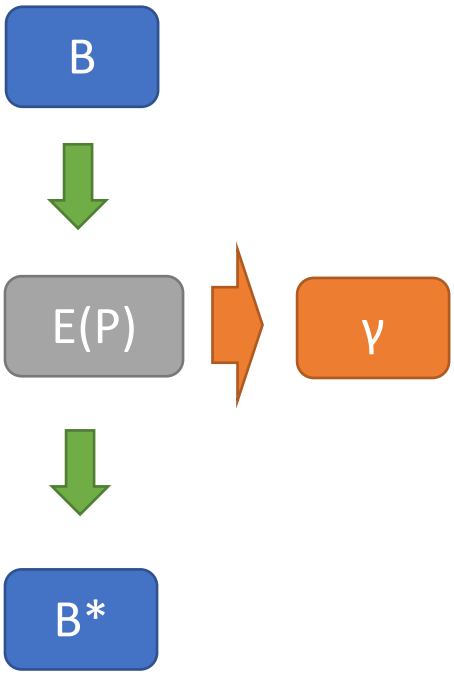

Figure 11: Illustration of assumption Σ being true.

However, similar to our definition of R, we can define the algorithm S (compare to figure 7):

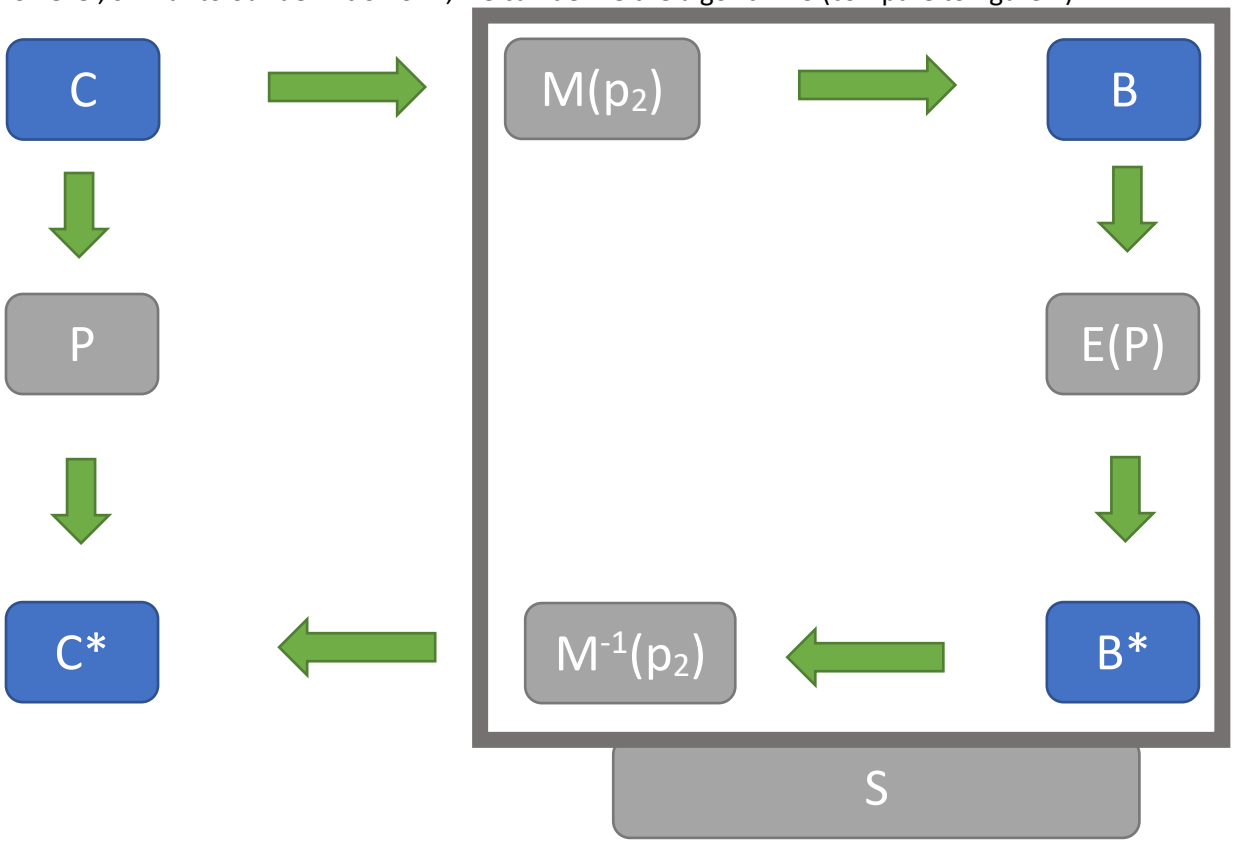

Figure 12: Illustration of the algorithm S.

Let δ be the experience generated when running S with input C. Using Ψ we would conclude that δ = β.

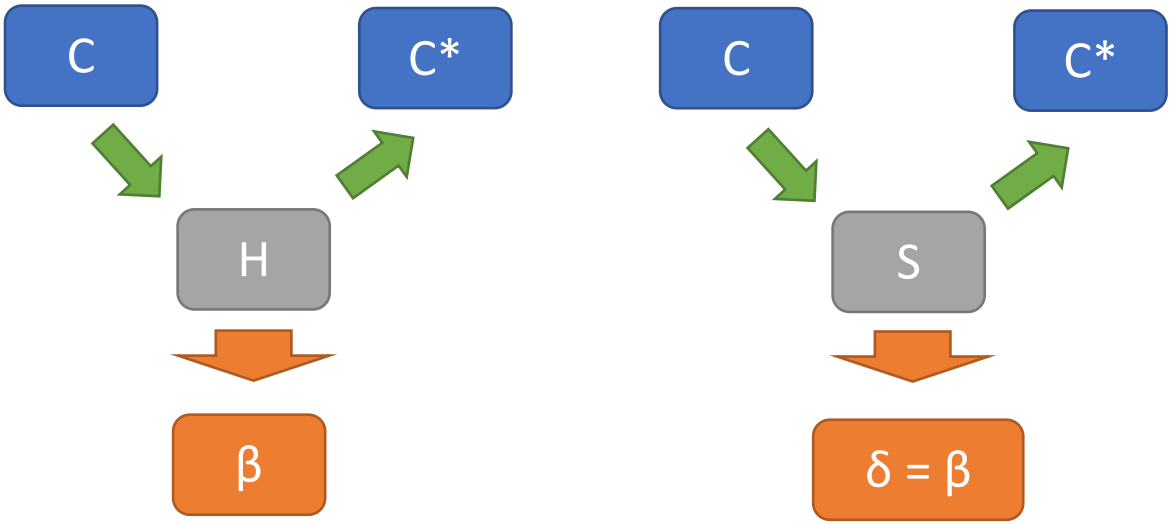

Figure 13: Illustration of assumption Ψ applied to algorithm S.

But using assumption Σ on algorithm S, we would conclude from figure 12 that δ = γ and therefore from figure 13 that γ = β. Thus, we get a contradiction (γ=α and γ=β with α≠β). It follows that Ψ must be wrong and simulating a conscious experience is not possible.

Now we consider the second case (assumption Σ is wrong): The way an input (e.g. B) was generated influences the experience that results from using this input as an input for an algorithm (e.g. E(P)). Then B generated as encryption of A could lead to a different conscious experience than B resulting from the encryption of C and the contradiction described above need not arise. However, B might also be the result of a random number generator, so using this B as an input for E(P) would again lead to a different experience. In this case, it would also be impossible to simulate a conscious experience, because it would be completely unclear which B (of the many equivalent texts B with different histories of generation) would lead to the experience (for example α) to be simulated. Thus, also when assumption Σ is wrong, we conclude that simulating conscious experiences is not possible.

Result: Consciousness cannot be simulated.

Additional remarks

1: Potential loopholes

Any conclusion rests on assumptions and definitions. Since there is no generally accepted definition of consciousness, this point is of particular relevance here. Therefore, we briefly discuss some underlying assumptions of our argument and explore plausible alternative assumptions, that might allow for simulations of conscious experiences.

1a: Validity of logical conclusions: The argument presented above rests on the assumption that logical reasoning can be used to derive true statements about questions concerning conscious experience. This is not necessarily the case. One can imagine that a simulation of a human being always produces the conscious experience (on the inside) that one would expect when looking at the simulation (from the outside). In this case, as we show, reality would contain logical contradictions. However, if we reject logical reasoning, it would be completely unclear what one would need to do to achieve a simulation of consciousness (e.g. run a computer simulation or peel an orange). Hence it still would not be possible to purposefully (instead of accidentally) simulate consciousness.

1b: Hyper-Panpsychism: In the argument we assume that there are different physical processes that are linked to different conscious experiences (e.g. α and β). This might not be the case. In what we call a Hyper-Panpsychism scenario, where any physical process leads to every possible conscious experience, generating different conscious experiences would be impossible. Simulating conscious experiences would then be possible, however not in a meaningful (targeted) way.

1c: Alternative assumption about the relationship between algorithms and conscious experiences: It is possible to modify assumption Ψ in such a way that the central contradiction of the argument above ($\gamma=\alpha$ and $\gamma=\beta$ with $\alpha\neq\beta$) disappears. We call this modification of Ψ assumption Φ.

Like Ψ, assumption Φ needs to contain a plausible reason why running a computer simulation might lead to conscious experiences. The only reason we could think of is similarity in appearance (observable physical properties), since internal conscious experiences are fundamentally inaccessible to outside observers (Nagel, 1974). So Φ should still entail that entities which behave in similar ways from an outside perspective, have similar experiences on the inside.

To get rid of the logical contradiction, we loosen the restrictions on Ψ, arriving at the following informal definition of Φ: “A computation creates all conscious experiences that this computation can be interpreted to simulate.” The implications of Φ depend mainly on the definition of “can be interpreted to simulate”. If this definition includes ambiguous encryption, it resolves the contradiction in the main argument, because instead of “$\gamma=\alpha$ and $\gamma=\beta$ with $\alpha\neq\beta$” we arrive at “$\gamma\supset\alpha$ and $\gamma\supset\beta$ with $\alpha\neq\beta$”, which is not a contradiction. Using the example of appendix A, we would conclude that running algorithm E(P) with input B will lead to at least two distinct conscious experiences (since we can decrypt B with the keys 5 and 7).

This alternative assumption Φ does not challenge the conclusion that consciousness cannot be simulated. Φ implies that a multitude of conscious experiences is created by a simulation (such as E(P) with input B), whether or not the output is decrypted. By contrast, assumption Ψ is only concerned with the appearance of the output of a simulation to an observer. The multitude of conscious experiences attached to a single computation (given assumption Φ) turns out to be problematic when we realize that any physical process can be interpreted as computation. As we show in appendix C, assumption Φ (if it is to resolve the contradiction of the main argument) implies that many common physical processes result in a large number of conscious experiences. Also running a computer simulation on a computer would thus generate very many conscious experiences.

The situation is therefore similar to the Hyper-Panpsychism scenario outlined above and it would not be possible to simulate a specific experience. We therefore conclude that simulating consciousness is also not possible under assumption Φ.

1d: Alternative reasons why a simulation might still be associated with conscious experiences: Our argumentation shows that similarity of appearance (same output for same input) is not a sufficient reason for concluding that two algorithms produce similar conscious experiences. There might however be other reasons to assume that a particular algorithm is associated with particular conscious experiences (see 2c).

2. Clarifications

2a: E(P) allows for dynamic interaction with simulated human behavior: We stated that the algorithm P will surely pass the Turing test. However, the schematics used in the argument above might lead one to conclude that it would not be possible to interact with a human being simulated by E(P), since only one input and output are depicted. This is not the case. In order to allow for continuing interactions, one need simply encode the current brain state in the output and use this output as part of the input for the next cycle of computation. Applying this scheme to the video setup described in the definitions section, it would then be possible to simulate two simultaneous video-calls with the same number of evaluations of E(P), as evaluations of P that are needed to simulate a single video-call.

2b: The type of simulation does not matter: The type of simulation seems to be irrelevant to our argumentation, as long as the algorithm can be executed by a Turing machine (this fact is used in appendix B). For example, the algorithm P might or might not be based on simulations of physical processes within the human nervous system. The argument presented above applies in either case.

2c: Replicating conscious experiences is not shown to be impossible:

Many people believe that similar human beings have similar conscious experiences under similar circumstances. People justify this belief in different ways. If this belief is justified by the assumption that human beings have conscious experiences due to the algorithms being executed by their brains, then one might conclude from our argumentation that human beings who appear to have similar conscious experiences from the outside, do not generally have similar conscious experiences from the inside. This conclusion might be regarded as problematic for our argumentation.

However, the underlying assumption that there is a connection between algorithms and conscious experiences, might be wrong. Instead, there might be a relationship between physical processes (the physical processes themselves, not their interpretations as algorithms) and conscious experiences (see 2d).

Therefore, our argument does not rule out the possibility of replicating conscious experiences by replicating physical processes (e.g. human beings). Replication does not have to be (and probably cannot be) perfect.

The statement that similar human beings have similar conscious experiences under similar circumstances is stronger than the statement that identical human beings have identical conscious experiences under identical circumstances. How similarity should be defined so that differences in similarity of physical processes correspond to differences in similarity of conscious experiences remains unclear. The grey area of similarity has been illustrated with the following question: What would happen if one would replace a single neuron in a human brain with an artificial device that

mimics the firing pattern of that neuron (Chalmers, 1995)? Many people would assume that the first replacement wouldn't change the experiences of that human being much. However, when it comes to full replacement, opinions strongly diverge. Our argumentation does not weigh in on this question.

2d: Compatibility with Panpsychism: While our argumentation is incompatible with certain theories of consciousness (i.e. those that suggest that conscious experiences can be simulated in the above sense), it is not incompatible with all theories of consciousness. In particular, our argumentation is compatible (but not necessarily uniquely compatible) with the view (referred to in 2c) that there are conscious experiences associated with every physical process and different physical processes are associated with different conscious experiences (Mørch, 2017).

2e: Simulations might still be conscious: Our argumentation does not answer the question whether or not simulations do generate conscious experiences. It only states that if computer simulations of human behavior do generate conscious experiences, these will not generally be similar to the experiences actual human beings would have under the simulated conditions.

3. Implications

The question, which physical processes are associated with which conscious experiences might seem quite abstract at first, but as we exemplify below, it has important practical implications.

3a: Consciousness as a value in itself: Consciousness in itself is assigned a high level of importance by many people. For example, it is important for the meaningfulness of the love between a person and its partner, that this partner has conscious experiences (has actual feelings, instead of only appearing to have such feelings). Accordingly, the proposed process of mind-uploading and related concepts, such as digital immortality, are bound to lose at least some of their appeal after realizing, that the conscious experiences of a simulated human nervous system (if any) will not be similar to the experiences of actual humans. If one values positive conscious experiences of humans, one should (at our present level of understanding) make sure that actual humans (not just simulations of humans) are around to have these experiences.

3b: Consciousness as a basis for ethical considerations: Conscious experiences are essential to many ethical considerations. For example, most people assume that other people have similar conscious experiences to their own experiences under similar circumstances. It is wrong to hit someone with a stick for no good reason, because (amongst other reasons) oneself would feel pain if being hit with a stick and pain is an experience with negative ethical value.

3c: Mind crime: Intuitively one might suspect that running a simulation of a suffering human being is unethical (mind crime) (Bostrom, 2014). However, our argument above suggests, that two simulations of which one (on the outside) appears to simulate a suffering human being, while the other appears to simulate a happy human being, might be ethically equivalent. Since consciousness cannot be simulated, potential conscious experiences associated with running either simulation are generally neither those of a happy human being, nor those of a suffering human being. Notably, this does not mean that running such simulations is ethical, since the conscious experiences associated with running such simulations (if any) are unknown. There might be other reasons why running simulations of suffering human beings might be unethical, maybe involving the conscious experiences inflicted on the viewers of such simulations or the intentions of the person producing such simulations.

3d: Tentativeness of ethical implications: We do not know how conscious experiences are associated with physical processes and therefore should be wary to conclude that certain actions do not cause

suffering. We need to be extra careful, since humans are strongly biased to conclude that their actions do not cause suffering, or that such suffering is not problematic, when these actions are in some way beneficial to human beings, as exemplified by the widespread mistreatment of animals. For example, one might be tempted to conclude that robots do not suffer in situations where human beings would. However, this conclusion is not supported by our argumentation above. Unlike in ambiguous encryption investigated above, where the same physical process could be interpreted as two different simulations, a robot being damaged and a robot just standing around are actually two different physical processes. One of these might be associated with conscious experiences similar to suffering while the other might be not. Such a difference in conscious experiences would hence not be due to different computations, but might be based on the difference in the actual physical processes (see 1d). Even for computer simulations (which may include ambiguous encryption) it might sometimes be best to behave as if the computer simulation was in fact having similar conscious experiences to the conscious experiences a human being would have under the simulated circumstances, despite the fact that our argument seems to show that this is not the case. For example, trying to ignore or change one's intuitions on the feelings of others might lead to a loss of empathy. In addition, there might be flaws in the presented argument and our understanding of consciousness certainly is limited.

References

Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies.* Oxford University Press.

Bostrom, N., & Sandberg, A. (2008). Whole brain emulation: a roadmap. Lanc Univ Accessed January 21 (2008): 2015.

Chalmers, D. J. (1995, December). The Puzzle of Conscious Experience. *Scientific America*, pp. 80-86.

Gentry, C. (2010). Computing arbitrary functions of encrypted data. *Communications of the ACM 53.3*, (pp. 97-105).

Kurzweil, R. (2010). *The singularity is near.* Gerald Duckworth & Co.

Mørch, H. H. (2017). Is Matter Conscious? *Nautilus Magazine*.

Nagel, T. (1974). What is it like to be a bat? *The philosophical review 83(4)*, pp. 435-450.

Oizumi, M., Albantakis, L., & Tononi, G. (2014). From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0. *PLoS computational biology, 10(5)*, p. e1003588.

Searle, J. (1980, September). Minds, brains, and programs. *Behavioral and Brain Sciences, 3(3)*, pp. 417-424. doi:10.1017/S0140525X00005756

Turing, A. (1950, October). Computing Machinery and Intelligence. *Mind, 59(236)*, pp. 433-460.

Appendix A – Ambiguous encryption

As noted in the main text, we follow a scheme described by Gentry (Gentry, 2010), which we simplify to establish an example of ambiguous encryption. Since our encryption scheme is not necessarily cryptographically secure, one could alternatively speak of "encoding". However, we nevertheless use "encryption", since the notion of an "encryption key" is commonplace. We define encryption and decryption as follows:

Encryption M

Encryption is bitwise (x denotes a single bit).

$$c = x + pq \tag{A1}$$

Here p is a prime number and q a non-zero integer. c is the encryption of the bit x. c can be represented by n bits, so encrypting a plaintext of m bits will lead to a ciphertext of m*n bits.

Decryption M^{-1}

Decryption correspondingly is done per segment of n bits. Given the encrypted c the algorithm outputs c mod p. We can easily check that this works:

$$c \bmod p = (x + pq) \bmod p = x \tag{A2}$$

Now we want to show that with these definitions we can realize **ambiguous encryption**, i.e. two plaintexts being mapped on the same ciphertext.

First, we demonstrate this with a simple example: plaintext1 and plaintext2 are different plaintexts and $M(p_1)$ and $M(p_2)$ are encryptions with different keys (p_1 and p_2) and conveniently chosen numbers q. (In a secure encryption scheme these numbers would be chosen at random. To see that choosing these numbers doesn't influence the fully homomorphic property, we can think of the non-random choices below as specific instances of random choices).

Since our simplified encryption has only two keys (here 5 and 7), we can show explicitly that this is possible, with the following table for the encryption of a single bit:

x(p1)	x(p2)	Key p1	Key p2	q(p1)	q(p2)	c(p1)	c(p2)
0	0	3	5	5	3	15	15
1	0	3	5	3	2	10	10
0	1	3	5	2	1	6	6
1	1	3	5	5	3	16	16

Table 1: Parameter values for an example of ambiguous encryption.

We see that c(p1) and c(p2) are identical for all combinations of plaintext bits. Thus, given a plaintext1 (001110101) and a plaintext2 (010110010), we can construct a ciphertext (15, 6, 10, 16, 16, 15, 10, 6, 10) that will decrypt to either of the two plaintexts, depending on which key (p1 or p2) we use.

Thus, we have:

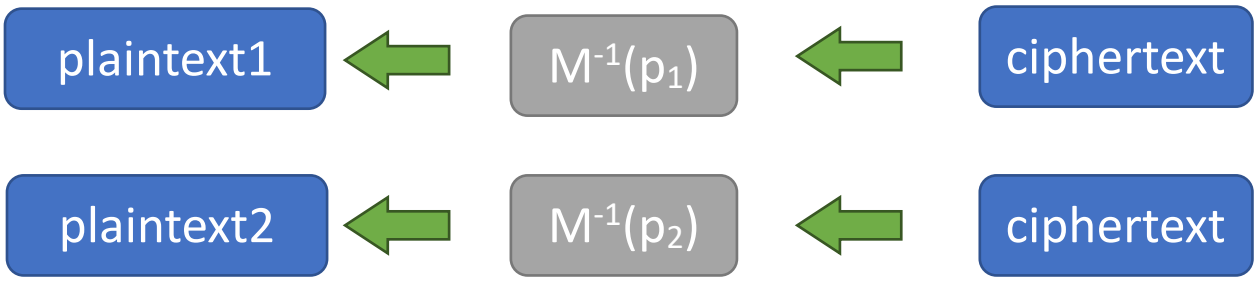

Figure 14: Illustration of ambiguous encryption. The plaintexts plaintext1 and plaintext2 correspond to A and C in figure 4, where B is the ciphertext.

The requirement that the two files need to have the same length can be solved by padding one plaintext in a way that does not alter the output of the algorithm P (P can presumably be constructed in such a way that this is possible).

Now we want to show that ambiguous encryption is possible for two arbitrarily large keys p1 and p2. To this end, we choose p1 and p2 to be prime numbers such that we can use the Euclidean algorithm below.

For all prime numbers p1 and p2 there exist integers q1, q2, …, q8 such that:

$0 + p1q1 = 0 + p2q2$

$1 + p1q3 = 0 + p2q4$

$0 + p1q5 = 1 + p2q6$

$1 + p1q7 = 1 + p2q8$

Proof:

$p1v + p2w = 1$ has a solution $v = a$ and $w = b$. This solution can be found by the reverse Euclidean algorithm[1].

Then the following parameters satisfy the requirements:

$q1 = p2$

$q2 = p1$

$q3 = -a$

$q4 = b$

$q5 = a$

$q6 = -b$

$q7 = p2$

$q8 = p1$

[1] https://www.math.uh.edu/~minru/number/hj02cs01.pdf

Appendix B – Evaluating functions on the ciphertext

We define the function E(P) based on the somewhat homomorphic encryption scheme presented by Gentry, using the fact that any function f that can be computed in T steps on a Turing machine can be expressed as a Boolean circuit with about T gates (Gentry, 2010).

E(P) should be independent of the key, in order for the relationship displayed in figure 7 to hold.

All Boolean circuits can be constructed from AND, OR and NOT gates. To evaluate these gates, it is sufficient to be able to add, subtract and multiply any two bits x and y (Gentry, 2010).

$$\mathrm{AND}(x, y) = xy \tag{B1}$$

$$\mathrm{OR}(x, y) = 1 - (1 - x)(1 - y) \tag{B2}$$

$$\mathrm{NOT}(x) = 1 - x \tag{B3}$$

Next, we show that decryption works after addition, multiplication and subtraction, that is, after using any function, when applying the encryption scheme defined above (equation A1):

$(c_1 + c_2) \bmod p = (x_1 + p\, q_1 + x_2 + p\, q_2) \bmod p = x_1 + x_2$, assuming $|x_1 + x_2| < p/2$

$(c_1 - c_2) \bmod p = (x_1 + p\, q_1 - x_2 - p\, q_2) \bmod p = x_1 - x_2$, assuming $|x_1 - x_2| < p/2$

$(c_1\, c_2) \bmod p = ((x_1 + p\, q_1)\,(x_2 + p\, q_2)) \bmod p = (x_1\, x_2 + p\, q_1\, x_2 + x_1\, p\, q_2 + p\, q_1\, p\, q_2) = x1\ x2$,

assuming $|x_1\, x_2| < p/2$

For evaluating Boolean circuits the assumptions $|x_1 + x_2| < p/2$, $|x_1 - x_2| < p/2$ and $|x_1\, x_2| < p/2$ are fulfilled for all p, since p are prime numbers in our encryption scheme and x_1 and x_2 are always either 0 or 1. Due to our simplified (and not necessarily cryptographically secure) encryption, we do not encounter here the problem of "growing noise" that Gentry addresses by bootstrapping (Gentry, 2010).

It is possible to use our encryption scheme to evaluate addition, multiplication and subtraction operations for $|x_i| > 1$, as long as the assumptions $|x_1 + x_2| < p/2$ and $|x_1\, x_2| < p/2$ are fulfilled. If many operations are evaluated in sequence, p needs to be sufficiently large for decryption to work correctly. The growth of $|x_i|$ in subsequent operations is, however, independent of p. For a program that stops (e.g. P with input A) there is a maximum $|x_i|$ encountered during running the program and we can always choose a key p larger than twice this maximum (since ambiguous encryption is possible for all prime numbers, as shown above).

Based on the somewhat homomorphic encryption scheme outlined by Gentry (Gentry, 2010), we define E(P) as follows:

Evaluation scheme E(P): *Represent the algorithm P as a Boolean circuit. Replace the evaluation of the gates by addition, subtraction and multiplication on the integers as defined above (using the ciphertext as an input). Output the resulting integers.*

We will demonstrate this evaluation scheme with a simple example.

Using plaintext1 (001110101) and a plaintext2 (010110010), we can construct a ciphertext (15, 6, 10, 16, 16, 15, 10, 6, 10) using the keys 3 and 5 (see appendix A).

First, we construct a Boolean circuit and use plaintext1 as an input.

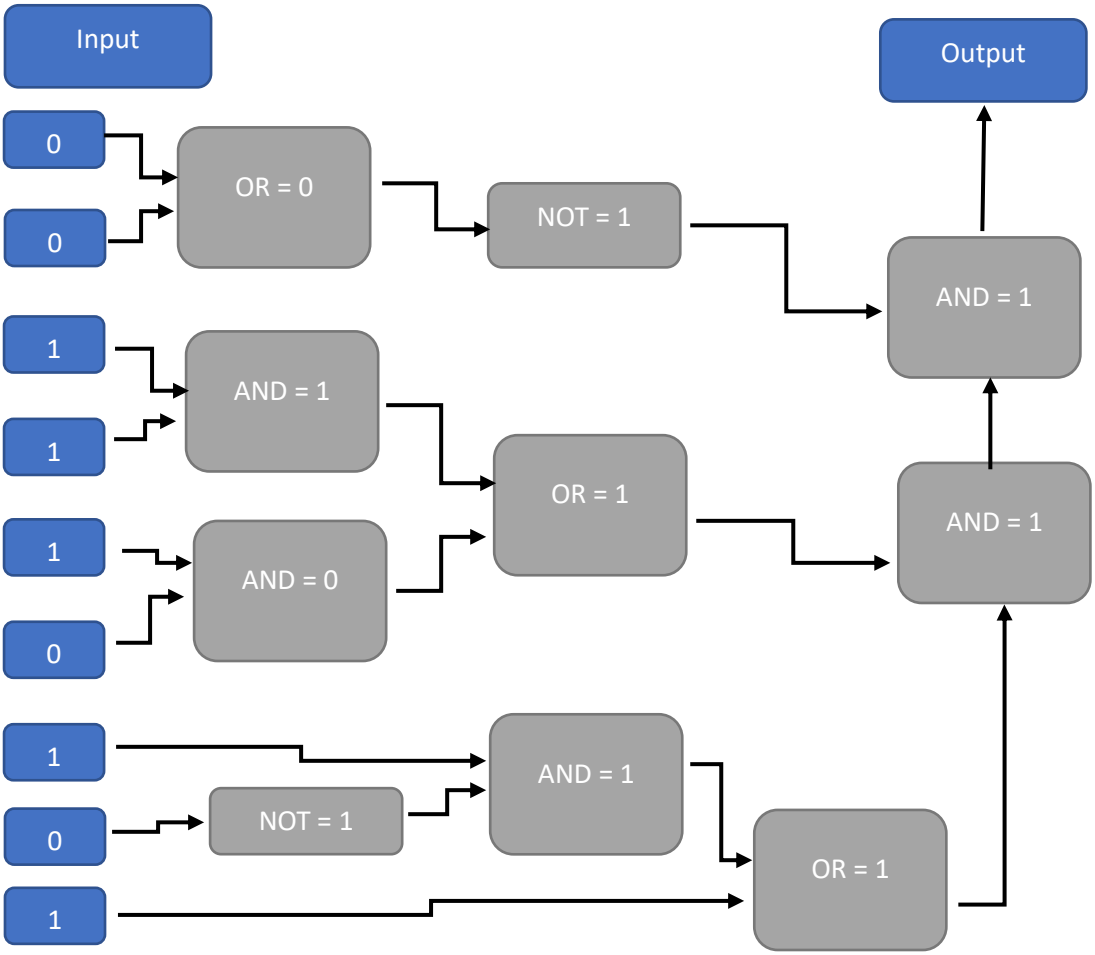

Figure 15: Example of a Boolean circuit being evaluated using plaintext1 as input.

Next, we feed plaintext2 in the same Boolean circuit.

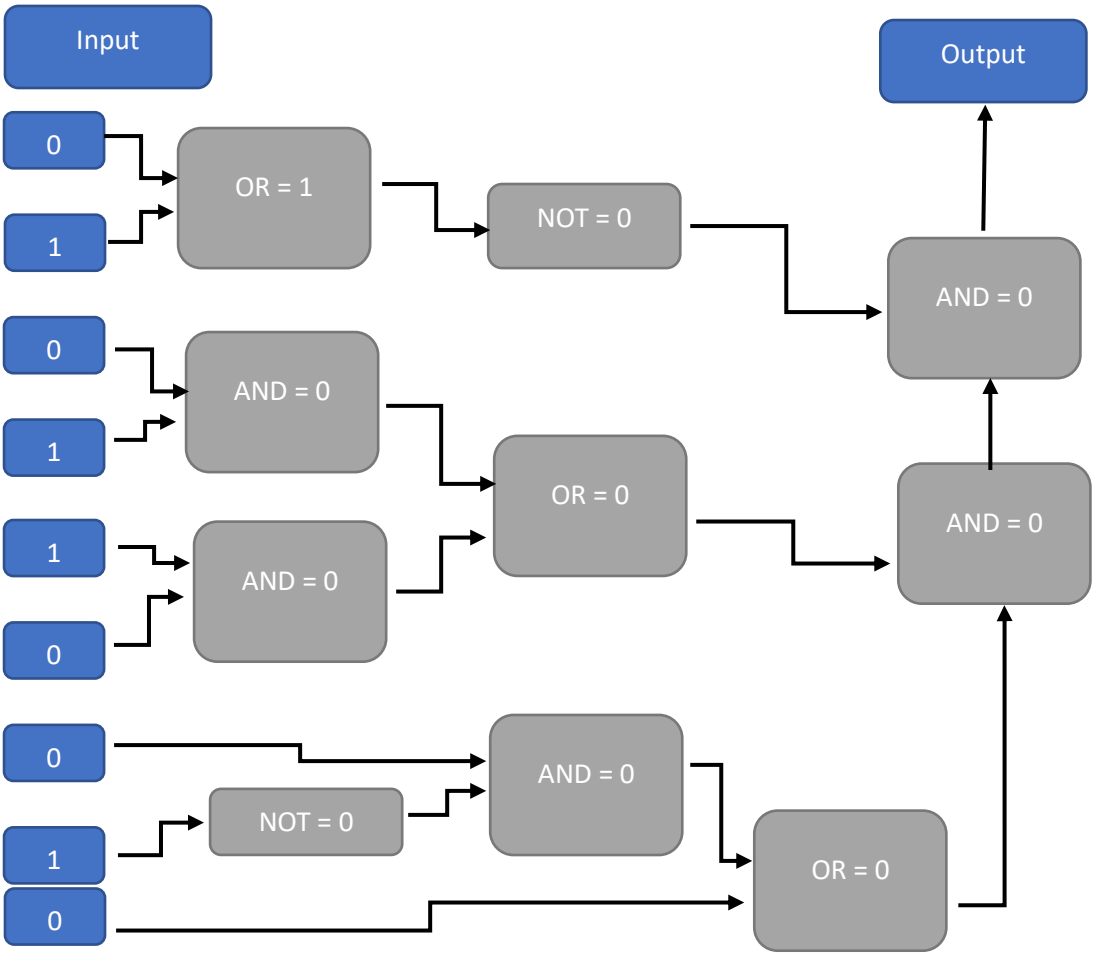

Figure 16: The Boolean circuit of figure 15 is being evaluated using plaintext2 as input.

Finally, we want to verify that decryption after application of the Boolean circuit on the ciphertext, results in the same output as running the Boolean circuit on the plaintext. The gates are evaluated as described in equations B1, B2 and B3.

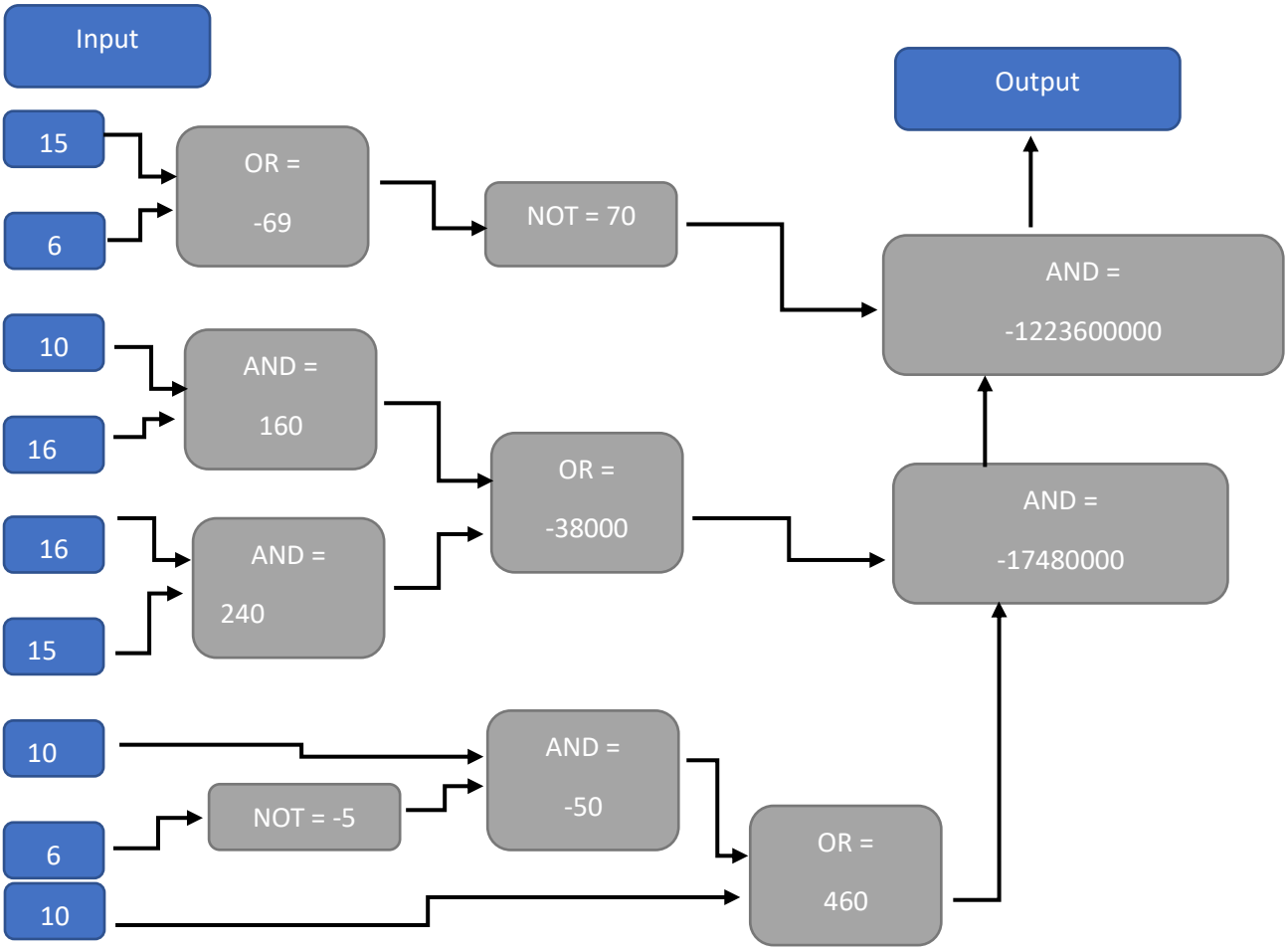

Figure 17: The Boolean circuit of figure 15 and 16 is being evaluated using the ciphertext as input.

Indeed, when we decrypt -1223600000 using the appropriate keys, we arrive at an output with value 1 for plaintext1 and an output with value 0 for plaintext2.

Appendix C – Interpreting physical processes as algorithms

Here we show that assumption Φ leads to a situation similar to the Hyper-Panpsychism scenario.

Physical processes can be interpreted as computations. In any physical system the initial values of some measurable quantities of the system can be interpreted as an input file D and the values of the same quantities after some time T can be interpreted as an output file D*. The physical processes happening during the time T can be interpreted as the algorithm F.

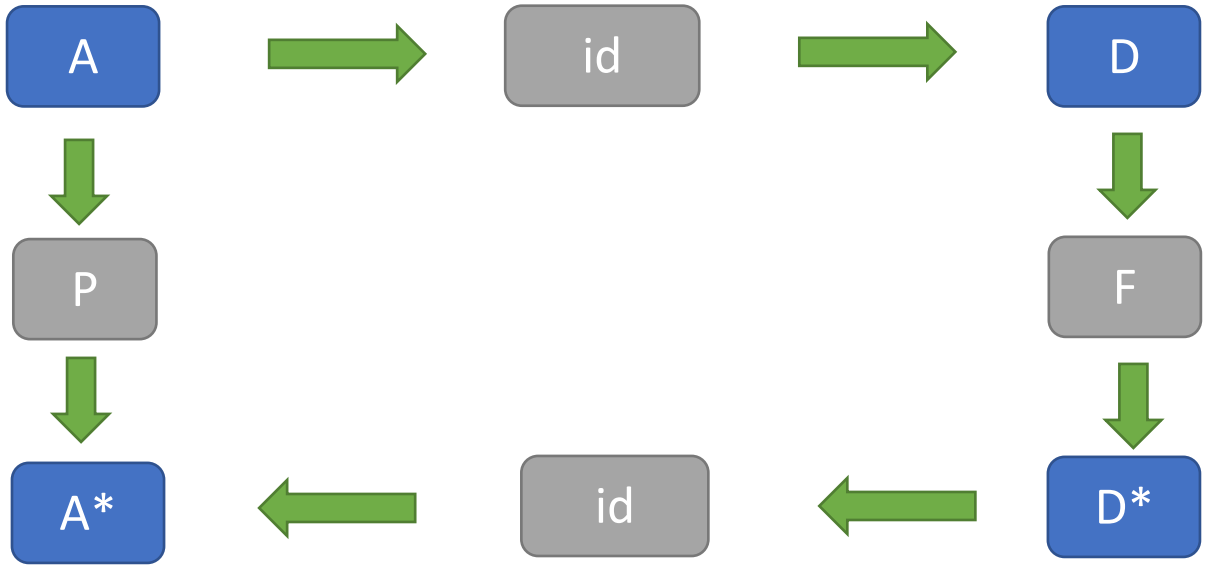

Figure 18: The situation depicted in this figure is similar to the one depicted in figure 7. Encryption and decryption are replaced by simple identity maps. F is an arbitrary physical process. P is a simulation of human behavior.

How likely is it, that the relationship of figure 18 exists for a physical process? To investigate this question, we assume that the physical process F is in this case completely random (e.g. a shaking box full of sand). We further assume that the size of the inputs and outputs is identical. The size of these data files is denoted by n. "Encryption" is a simple process now, since we just need to select any n physically measurable binary quantities (for example a particular bit in the location variable of a specific sand grain) that initially have the same number of ones and zeros as A and order them in a way so that D is identical to A. Since we assumed that time evolution is completely random, the bits of D will flip randomly as time passes. Let us assume that one random bit-flip takes time t. Let us further assume that 50% of the bits in A equal zero and 50% equal one, such that (for large n) we can ignore the problem of selecting the right number of ones and zeros. If we have a total of N one-bit-variables from which to choose n, then the time that needs to pass in order to have a better than 50% chance of arriving at A* at some point in the time evolution and for some parameter combination is equal to:

$$T(\text{Probabilty}(D^* = A^*) > 0.5) = \frac{t \log(0.5)}{\log(1-0.5^n)} \frac{n!(N-n)!}{N!} \quad \text{(C1)}$$

Which converges to zero for large n and N=kn with sufficiently large k (k=2 is sufficiently large, k=1 is too small).

This means that the relationship depicted in figure 18 will hold for a small but non-zero subset of the time-evolution of physical processes. Importantly, the computation as which we can interpret the physical process that connects D to D* is not random, despite the fact that we analyze it using statistical methods. For macroscopic systems (like grains of sand) time evolution is governed by

deterministic physical laws. The corresponding computations could be performed on a conventional computer. Therefore, assumption Φ implies that many physical processes (e.g. computers running simulations) create all kinds of conscious experiences. This situation is similar to the Hyper-Panpsychism scenario used in the main argument, which doesn't allow for meaningful simulation of conscious experiences.

Acknowledgements

We thank Carlos Franke for very useful comments.