entropy

MDPI

Article

Integrated Information Theory and Isomorphic Feed-Forward Philosophical Zombies

Jake R. Hanson [1,2] and Sara I. Walker [1,2,3,*]

[1] School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287, USA; jake.hanson@asu.edu
[2] Beyond Center for Fundamental Concepts in Science, Arizona State University, Tempe, AZ 85281, USA
[3] ASU–SFI Center for Biosocial Complex Systems, Arizona State University, Tempe, AZ 85281, USA
[*] Correspondence: sara.i.walker@asu.edu

Received: 1 August 2019; Accepted: 29 October 2019; Published: 2 November 2019

Abstract: Any theory amenable to scientific inquiry must have testable consequences. This minimal criterion is uniquely challenging for the study of consciousness, as we do not know if it is possible to confirm via observation from the outside whether or not a physical system knows what it feels like to have an inside—a challenge referred to as the "hard problem" of consciousness. To arrive at a theory of consciousness, the hard problem has motivated development of phenomenological approaches that adopt assumptions of what properties consciousness has based on first-hand experience and, from these, derive the physical processes that give rise to these properties. A leading theory adopting this approach is Integrated Information Theory (IIT), which assumes our subjective experience is a "unified whole", subsequently yielding a requirement for physical feedback as a necessary condition for consciousness. Here, we develop a mathematical framework to assess the validity of this assumption by testing it in the context of isomorphic physical systems with and without feedback. The isomorphism allows us to isolate changes in Φ without affecting the size or functionality of the original system. Indeed, the only mathematical difference between a "conscious" system with $\Phi > 0$ and an isomorphic "philosophical zombie" with $\Phi = 0$ is a permutation of the binary labels used to internally represent functional states. This implies Φ is sensitive to *functionally arbitrary aspects of a particular labeling scheme*, with no clear justification in terms of phenomenological differences. In light of this, we argue any quantitative theory of consciousness, including IIT, should be invariant under isomorphisms if it is to avoid the existence of isomorphic philosophical zombies and the epistemological problems they pose.

Keywords: consciousness; integrated information theory; Krohn–Rhodes decomposition

1. Introduction

The scientific study of consciousness walks a fine line between physics and metaphysics. On the one hand, there are observable consequences to what we intuitively describe as consciousness. Sleep, for example, is an outward behavior that is uncontroversially associated with a lower overall level of consciousness. Similarly, scientists can decipher what is intrinsically experienced when humans are conscious via verbal reports or other outward signs of awareness. By studying the physiology of the brain during these specific behaviors, scientists can study "neuronal correlates of consciousness" (NCCs), which point to where in the brain conscious experience is generated and what physiological processes correlate with it [1]. On the other hand, NCCs cannot be used to explain *why* we are conscious or to predict whether or not another system demonstrating similar properties to NCCs is conscious. Indeed, NCCs can only tell us the physiological processes that correlate with what are assumed to be the functional consequences of consciousness and, in principle, may not actually correspond to a measurement of what it is like to have subjective experience [2]. In other words, we can objectively measure behaviors we assume accurately reflect consciousness but, currently, there exist no scientific tools permitting testing our assumptions. As a result, we struggle to differentiate whether a system is truly conscious or is instead simply going through the motions and giving outward signs of, or even actively reporting, an internal experience that does not exist (e.g., [3]).

This is the "hard problem" of consciousness [2] and it is what differentiates the study of consciousness from all other scientific endeavors. Since consciousness is subjective (by definition), there is no objective way to prove whether or not a system experiences it readily accessible to science. Addressing the hard problem, therefore, necessitates an inversion of the approach underlying NCCs: rather than starting with observables and deducing consciousness, one must start with consciousness and deduce observables. This has motivated theorists to develop phenomenological approaches that adopt rigorous assumptions of what properties consciousness must include based on human experience, and, from these, "derive" the physical processes that give rise to these properties. The benefit to this approach is not that the hard-problem is avoided, but rather, that the solution appears self-evident given the phenomenological axioms of the theory. In practice, translating from phenomenology to physics is rarely obvious, but the approach remains promising.

The phenomenological approach to addressing the hard problem of consciousness is exemplified in Integrated Information Theory (IIT) [4,5], a leading theory of consciousness. Indeed, IIT is a leading contender in modern neuroscience precisely because it takes a phenomenological approach and offers a well-motivated solution to the hard problem of consciousness [6]. Three phenomenological axioms form the backbone of IIT: information, integration, and exclusion. The first, *information*, states that by taking on only one of the many possibilities a conscious experience generates information (in the Shannon sense, e.g., via a reduction in uncertainty [7]). The second, *integration*, states each conscious experience is a single "unified whole". The third, *exclusion*, states conscious experience is exclusive in that each component in a system can take part in at most one conscious experience at a time (simultaneous overlapping experiences are forbidden). Given these three phenomenological axioms, IIT derives a mathematical measure of integrated information—Φ—that is designed to quantify the extent to which a system is conscious based on the logical architecture (i.e., the "wiring") underlying its internal dynamics.

In constructing Φ as a phenomenologically-derived measure of consciousness, IIT must assume a connection between its phenomenological axioms and the physical processes that embody those axioms. It is important to emphasize that this assumption is nothing less than a proposed solution to the hard problem of consciousness, as it connects subjective experience (axiomatized as integration, information, and exclusion) and objective (measurable) properties of a physical system. As such, it is possible for one to accept the phenomenological axioms of the theory without accepting Φ as the correct quantification of these axioms and, indeed, IIT has undergone several revisions in an attempt to better reflect the phenomenological axioms in the proposed construction of Φ [5,8,9]. Experimental falsification of IIT or any other similarly constructed theory is a matter of sufficiently violating our intuitive understanding of what a measure of consciousness should predict in a given situation which, outside of a few clear cases (e.g., that humans are conscious while awake), varies across individuals and adds a level of subjectivity to assessing the validity of measures derived based on phenomenology. Given that IIT is a phenomenological theory, it is therefore only natural that the bulk of epistemic justification for the theory comes in the form of carefully constructed logical arguments, rather than directly from empirical observation [10]. For this reason, it is extremely important to isolate and understand the logical assumptions that underlie any potentially controversial deductions that come from the theory, as this plays an important role in assessing the foundations of the theory.

Here, we focus on a particularly controversial aspect of IIT, namely, the fact that philosophical zombies are permitted by the theory. By definition, a philosophical zombie is an unconscious system capable of perfectly emulating the outward behavior of a conscious system. In addition to being epistemologically problematic [11,12], such systems are thought to indicate problems with the logical foundations of any theory that admits them [13], as it is difficult to imagine how a difference in subjective experience can be scientifically justified without any apparent difference in the outward functionality of the system [14]. In IIT, philosophical zombies arise as a direct consequence of IIT's proposed translation of the integration axiom. IIT assumes the subjective experience of a unified whole (the integration axiom) requires feedback in the physical substrate that gives rise to consciousness as a necessary (but not sufficient) condition. This implies any strictly feed-forward logical architecture has $\Phi = 0$ and is unconscious by default, despite the fact that the logical architecture of an "integrated" system with $\Phi > 0$ can always be unfolded [15] or decomposed [16,17] into a system with $\Phi = 0$ without affecting the outward behavior of the system.

In what follows, we demonstrate the existence of a fundamentally new type of feed-forward philosophical zombie, namely one that is *isomorphic* to its conscious counterpart in its state-transition diagram. To do so, we implement techniques based on Krohn–Rhodes decomposition from automata theory to isomorphically decompose a system with $\Phi > 0$ onto a feed-forward system with $\Phi = 0$. The result is a feed-forward philosophical zombie capable of perfectly emulating the behavior of its conscious counterpart *without increasing the size of the original system*. Given the strong mathematical equivalence between isomorphic systems, our framework suggests the presence or absence of feedback is not associated with observable differences in function or other properties such as efficiency or autonomy. Our formalism translates into a proposed mathematical criterion that any observationally verifiable measure of consciousness should be invariant under physical isomorphisms. That is, we suggest conscious systems should form an equivalence class of physical implementations with structurally equivalent state-transition diagrams. Enforcement of this criterion serves as a necessary, but not sufficient, condition for any theory of consciousness to be free from philosophical zombies and the epistemological problems they pose.

2. Methods

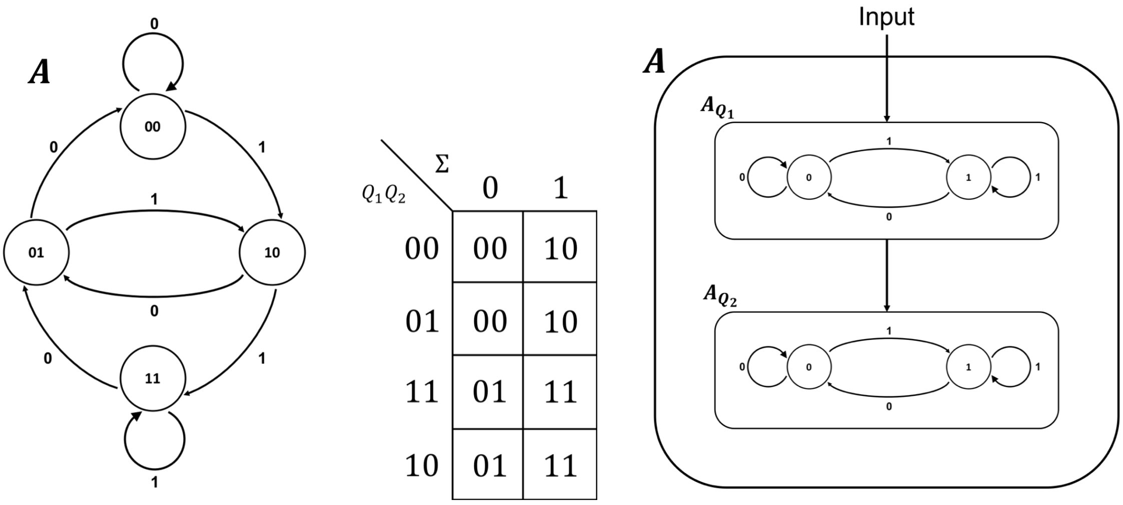
Our methodology is based on automata theory [18,19], where the concept of philosophical zombies has a natural interpretation in terms of "emulation" [20]. The goal of our methodology is to demonstrate that it is possible to isomorphically emulate an integrated finite-state automaton ($\Phi > 0$) with a feed-forward finite-state automaton ($\Phi = 0$) using techniques closely related to the Krohn–Rhodes theorem [16,21].

2.1. Finite-State Automata

Finite-state automata are abstract computing devices, or "machines", designed to model a discrete system as it transitions between states. Automata theory was invented to address biological and psychological problems [22,23] and it remains an extremely intuitive choice for modeling neuronal systems. This is because one can define an automaton in terms of how specific abstract inputs lead to changes within a system. Namely, if we have a set of potential inputs Σ and a set of internal states Q, we define an automaton A in terms of the tuple $A = (\Sigma, Q, \delta, q_0)$ where $\delta : \Sigma \times Q \to Q$ is a map from the current state and input symbol to the next state, and $q_0 \in Q$ is the starting state of the system. To simplify notation, we write $\delta(s, q) = q'$ to denote the transition from q to q' upon receiving the input symbol $s \in \Sigma$.

For example, consider the "right-shift automaton" A shown in Figure 1. This automaton is designed to model a system with a two-bit internal register that processes new elements from the input alphabet $\Sigma = \{0, 1\}$ by shifting the bits in the register to the right and appending the new element on the left [24]. The global state of the machine is the combined state of the left and right register, thus $Q = \{00, 01, 10, 11\}$ and the transition function δ specifies how this global state changes in response to each input, as shown in Figure 1b.

In addition to the global state transitions, each individual bit in the register of the right-shift automaton is itself an automaton. In other words, the global functionality of the system is nothing more than the combined output from a system of interconnected automata, each specifying the state of a single component or "coordinate" of the system. Specifically, the right-shift automaton is comprised of an automaton A_{Q1} responsible for the left bit of register and an automaton A_{Q2} responsible for the right bit of the register. By definition, A_{Q_1} copies the input from the environment and A_{Q_2} copies the state of A_{Q1}. Thus, $\Sigma_{Q_1} = \{0,1\}$ and $\Sigma_{Q_2} = Q_1 = \{0,1\}$ and the transition functions for the coordinates are $\delta_{Q_1} = \delta_{Q_2} = \{\delta(0,0) = 0; \delta(0,1) = 0; \delta(1,0) = 1; \delta(1,1) = 1\}$. This fine-grained view of the right-shift automaton specifies its *logical architecture* and is shown in Figure 1c. The logical architecture of the system is the "circuitry" that underlies its behavior and, as such, is often specified explicitly in terms of logic gates, with the implicit understanding that each logic gate is a component automaton.



Q_1Q_2 \ Σ	0	1
00	00	10
01	00	10
11	01	11
10	01	11

(**a**) (**b**) (**c**)

Figure 1. The "right-shift automaton" A in terms of its: state-transition diagram (**a**); transition function δ (**b**); and logical architecture (**c**).

It is important to note that not all automata require multiple input symbols and it is common to find examples of automata with a single-letter input alphabet. In fact, any deterministic state-transition diagram can be represented in this way, with a single input letter signaling the passage of time. In this case, the states of the automaton are the states of the system, the input alphabet is the passage of time, and the transition function δ is given by the transition probability matrix (TPM) for the system. Because Φ is a mathematical measure that takes a TPM as input, this specialized case provides a concrete link between IIT and automata theory. Non-deterministic TPMs can also be described in terms of finite-state automata [24,25] but, for our purposes, this generalization is not necessary.

2.2. Cascade Decomposition

The idea of decomposability is central to both IIT and automata theory. As Tegmark [26] pointed out, mathematical measures of integrated information, including Φ, quantify the inability to decompose a transition probability matrix M into two independent processes M_A and M_B. Given a distribution over initial states p, if we approximate M by the tensor factorization $\hat{M} \approx M_A \otimes M_B$, then Φ, in general, quantifies an information-theoretic distance D between the regular dynamics Mp and the dynamics under the partitioned approximation $\hat{M}p$ (i.e., $\Phi = D(Mp||\hat{M}p)$). In the latest version of IIT [5], only *unidirectional* partitions are implemented (information can flow in one direction across the partition) which mathematically enforces the assumption that feedback is a necessary condition for consciousness.

Decomposition in automata theory, on the other hand, has historically been an engineering problem. The goal is to decompose an automaton A into an automaton A' which is made of simpler physical components than A and maps *homomorphically* onto A. Here, we define a homomorphism h as a map from the states, stimuli, and transitions of A' onto the states, stimuli, and transitions of A

such that for every state and stimulus in A' the results obtained by the following two methods are equivalent [22]:

1. Use the stimulus of A' to update the state of A' then map the resulting state onto A.
2. Map the stimulus of $A'$and the state of A' to the corresponding stimulus/state in A then update the state of A using the stimulus of A.

In other words, the map h is a homomorphism if it *commutes* with the dynamics of the system. The two operations (listed above) that must commute are shown schematically in Figure 2. If the homomorphism h is bijective, then it is also an *isomorphism* and the two automata necessarily have the same number of states.

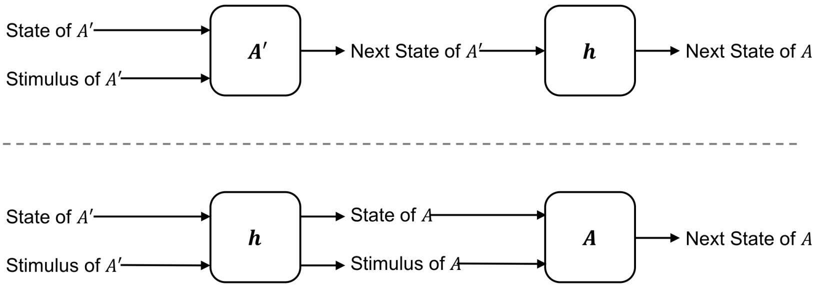


Figure 2. For the map h to be a homomorphism from A' onto A, updating the dynamics then applying h (**top**) must yield the same state of A as applying h then updating the dynamics (**bottom**).

From an engineering perspective, homomorphic/isomorphic logical architectures are useful because they allow flexibility when choosing a logical architecture to implement a given computation (i.e., the homomorphic system can perfectly emulate the original). Mathematically, the difference between homomorphic automata is the internal labeling scheme used to encode the states/stimuli of the global finite-state machine, which specifies the behavior of the system. Thus, the homomorphism h is a dictionary that translates between different representations of the same computation. Just as the same sentence can be spoken in different languages, the same computation can be instantiated using different encodings. Under this view, what gives a computational state meaning is not its binary representation (label) but rather its causal relationship with other global states/stimuli, which is what the homomorphism preserves.

Because we are interested in isolating the role of feedback, the specific type of decomposition we seek is a feed-forward or *cascade decomposition* of the logical architecture of a given system. Cascade decomposition takes the automaton A and decomposes it into a homomorphic automaton A' comprised of several elementary automata "cascaded together". By this, what is meant is that the output from one component serves as the input to another such that the flow of information in the system is strictly unidirectional (Figure 3). The resulting logical architecture is said to be in "cascade" or "hierarchical" form and is functionally identical to the original system (i.e., it realizes the same global finite-state machine).

At this point, the connection between IIT and cascade decomposition is readily apparent: if an automaton with feedback allows a homomorphic cascade decomposition, then the behavior of the resulting system can emulate the original but utilizes only feed-forward connections. Therefore, there exists a unidirectional partition of the system that leaves the dynamics of the new system (i.e., the transition probability matrix) unchanged such that $\Phi = 0$ for all states.

In the language of Oizumi et al. [5], we can prove this by letting $C_{\rightarrow}$ be the constellation that is generated as a result of any unidirectional partition and C be the original constellation. Because $C_{\rightarrow}$ has no effect on the TPM, we are guaranteed that $C_{\rightarrow} = C$ and $\Phi^{MIP} = D(C|C_{\rightarrow}) = 0$. We can repeat this process for every possible subsystem within a given system and, since the flow of information is always unidirectional, $\Phi^{MIP} = 0$ for all subsets so $\Phi^{Max} = 0$. Thus, $\Phi = 0$ for all states and subsystems of a cascade automaton.

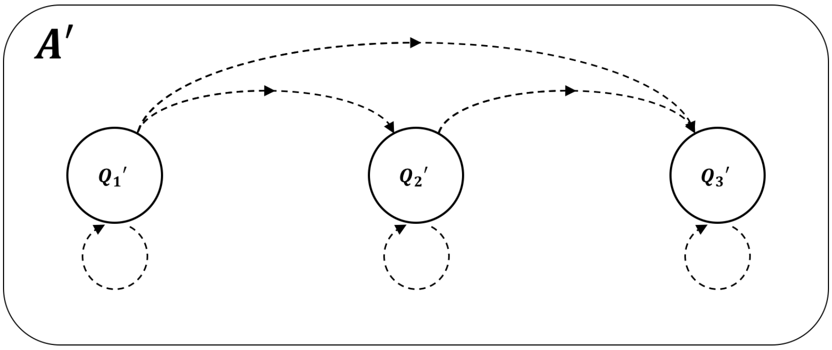


Figure 3. An example of a fully-connected three-component system in cascade form. Any subset of the connections drawn above meets the criteria for cascade form because all information flows unidirectionally.

Pertinently, the Krohn–Rhodes theorem proves that every automaton can be decomposed into cascade form [16,17], which implies *every system for which we can measure non-zero* Φ *allows a feed-forward decomposition with* $\Phi = 0$. These feed-forward systems are "philosophical zombies" in the sense that they lack subjective experience according to IIT (i.e., $\Phi = 0$), but they nonetheless perfectly emulate the behavior of conscious systems. However, the Krohn–Rhodes theorem does not tell us *how* to construct such systems. Furthermore, the map between systems is only guaranteed to be homomorphic (many-to-one) which allows for the possibility that Φ is picking up on other properties (e.g., the efficiency and/or autonomy of the computation) in addition to the presence or absence of feedback [5].

To isolate what Φ is measuring, we must go one step further and insist that the decomposition is isomorphic (one-to-one) such that the original and zombie systems can be considered to perform the same computation [20] (same global state-transition topology) under the same resource constraints. In this case, the feed-forward system has the *exact same number of states* as its counterpart with feedback. Provided the latter has $\Phi > 0$, this implies Φ is not a measure of the efficiency of a given computation, as both systems require the same amount of memory. This is not to say that feedback and Φ do not *correlate* with efficiency because, in general, they do [27]. For certain computations, however, the presence of feedback is not associated with increased efficiency but only increased interdependence among elements.

It is these specific corner cases that are most beneficial if one wants to assess the validity of the theory, as they allow one to understand whether or not feedback is important in absence of the benefits typically associated with its presence. In other words, IIT's translation of the integration axiom is that *feedback* is a minimal criterion for the subjective experience of a unified whole; however, Φ is described as quantifying "the amount of information generated by a complex of elements, above and beyond the information generated by its parts" [4], which seems to imply feedback enables something "extra" feed-forward systems cannot reproduce. An isomorphic feed-forward decomposition allows us to carefully track the mathematical changes that destroy this additional information, in a way that lets us preserve the efficiency and functionality of the original system. This, in turn, provides the clearest possible case to assess whether or not this additional information is likely to correspond to a phenomenological difference between systems.

2.3. Feed-Forward Isomorphisms via Preserved Partitions

The special type of computation that allows an isomorphic feed-forward decomposition is one in which the global state-transition diagram is amenable to decomposition via a nested sequence of preserved partitions. A preserved partition is a way of partitioning the state space of a system into blocks of states (macrostates) that transition together. Namely, a partition P is preserved if it breaks the state space S into a set of blocks $\{B_1, B_2, ..., B_N\}$ such that every state within each block transitions to a state within the same block [22,28]. If we denote the state-transition function $f : S \to S$, then a block B_i is preserved when:

$$\exists j \in \{1, 2, ..., N\} \text{ such that } f(x) \in B_j \forall x \in B_i$$

In other words, for B_i to be preserved, $\forall x$ in B_i x must transition to some state in a single block B_j ($i = j$ is allowed). Conversely, B_i is *not* preserved if there exist two or more states in B_i that transition to different blocks (i.e., $\exists\, x_1, x_2 \in B_i$ such that $f(x_1) = B_j$ and $f(x_2) = B_k$ with $j \neq k$). In order for the entire partition P_i to be preserved, each block within the partition must be preserved.

For an isomorphic cascade decomposition to exist, we must be able to iteratively construct a hierarchy or "nested sequence" of preserved partitions such that each partition P_i evenly splits the partition P_{i-1} above it in half, leading to a more and more refined description of the system. For a system with 2^n states where n is the number of binary components in the original system, this implies that we need to find exactly n nested preserved partitions, each of which then maps onto a unique component of the cascade automaton, as demonstrated in Section 2.3.1.

If one cannot find a preserved partition made of disjoint blocks or the blocks of a given partition do not evenly split the blocks of the partition above it in half, then the system in question does not allow an isomorphic feed-forward decomposition. It will, however, still allow a *homomorphic* feed-forward decomposition based on a nested sequence of preserved *covers*, which forms the basis of standard Krohn–Rhodes decomposition techniques [20,22,29]. Unfortunately, there does not appear to be a way to tell a priori whether or not a given computation will ultimately allow an isomorphic feed-forward decomposition, although a high degree of symmetry in the global state-transition diagram is certainly a requirement.

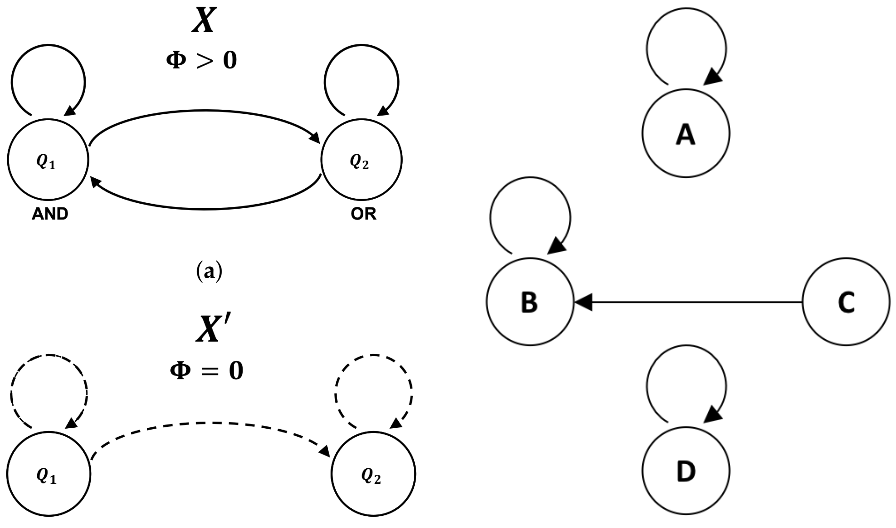
2.3.1. Example: AND/OR ≅ COPY/OR

As an example, we isomorphically decompose the feedback system X, comprised of an AND gate and an OR gate, as shown in Figure 4a. As it stands, X is not in cascade form because information flows bidirectionally between the components Q_1 and Q_2. While this feedback alone is insufficient to guarantee $\Phi > 0$, one can readily check that X does indeed have $\Phi > 0$ for all possible states (e.g., [30]). The global state-transition diagram for the system X is shown in Figure 4c. Note that we have purposefully left off the binary labels that X uses to instantiate these computational states, as the goal is to relabel them in a way that results in a different (feed-forward) instantiation of the same underlying computation. In general, one typically starts from the computation and derives a single logical architecture but, here, we must start and end with fixed (isomorphic) logical architectures—passing through the underlying computation in between. The general form of the feed-forward logical architecture X' that we seek is shown in Figure 4b.

Given the global state-transition diagram shown in Figure 4c, we let our first preserved partition be $P_1 = \{B_0, B_1\}$ with $B_0 = \{A, D\}$ and $B_1 = \{B, C\}$. It is easy to check that this partition is preserved, as one can verify that every element in B_0 transitions to an element in B_0 and every element in B_1 transitions to an element in B_1 (shown topologically in Figure 5a). We then assign all the states in B_0 a first coordinate value of 0 and all the states in B_1 a first coordinate value of 1, which guarantees the state of the first coordinate is independent of later coordinates. If the value of the first coordinate is 0, it will remain 0 and, if the value of the first coordinate is 1, it will remain 1, because states within a given block transition together. Because 0 goes to 0 and 1 goes to 1, the logic element (component automaton) representing the first coordinate Q'_1 is a COPY gate receiving its previous state as input.

The second preserved partition P_2 must evenly split each block within P_1 in half. Letting $P_2 = \{\{B_{00}, B_{01}\}, \{B_{10}, B_{11}\}\}$ we have $B_{00} = \{A\}$, $B_{01} = \{B\}$, $B_{10} = \{C\}$, and $B_{11} = \{D\}$. At this stage, it is trivial to verify that the partition is preserved because each block is comprised of only one state which is guaranteed to transition to a single block. As with Q'_1, the logic gate for the second coordinate (Q'_2) is specified by the way the labeled blocks of P_2 transition. Namely, we have $B_{00} \to B_{00}$, $B_{01} \to B_{01}$, $B_{10} \to B_{01}$, and $B_{11} \to B_{11}$. Note that the transition function δ_{Q2} is completely deterministic given input from the first two coordinates (as required) and is given by $\delta_{Q2} = \{00 \to 0; 01 \to 1; 10 \to 1; 11 \to 1\}$. This implies Q_2' is an `OR` gate receiving input from both Q_1' and Q_2'.

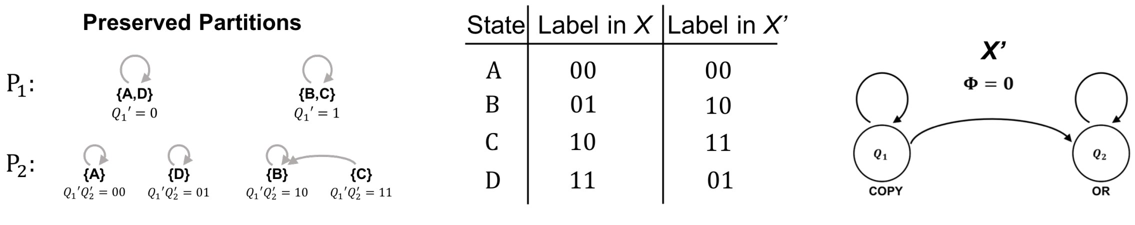
At this point, the isomorphic cascade decomposition is complete. We have constructed an automaton for Q_1' that takes input from only itself and an automaton for Q_2' that takes input only from itself and earlier coordinates (i.e., Q_1' and Q_2'). The mapping between the states of X and the states of X', shown in Figure 5b, is specified by identifying the binary labels (internal representations) each system uses to instantiate the abstract computational states A, B, C, D of the global state-transition diagram. Because X and X' operate on the same support (the same four binary states) the fact that they are isomorphic implies the difference between representations is nothing more than a permutation of the labels used to instantiate the computation. By choosing a specific labeling scheme based on isomorphic cascade decomposition, we can induce a logical architecture that is guaranteed to be feedback-free and has $\Phi = 0$. In this way, we have "unfolded" the feedback present in X without affecting the size/efficiency of the system.



(**a**)

(**b**) (**c**)

Figure 4. The goal of an isomorphic cascade decomposition is to decompose the integrated logical architecture of the system X (**a**) so that it is in cascade form X' (**b**) without affecting the state-transition topology of the original system (**c**).



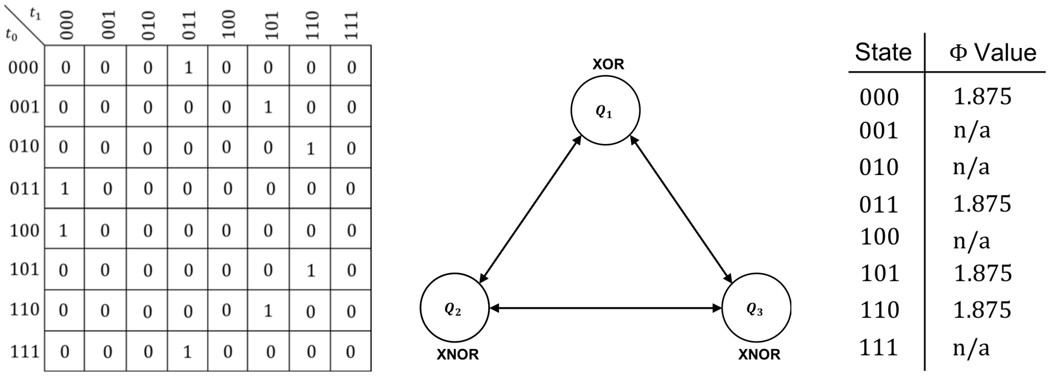
State	Label in X	Label in X'
A	00	00
B	01	10
C	10	11
D	11	01

(**a**) (**b**) (**c**)

Figure 5. The nested sequence of preserved partitions in (**a**) yields the isomorphism (**b**) between X and X' which can be translated into the strictly feed-forward logical architecture with $\Phi = 0$ shown in (**c**).

3. Results

We are now prepared to demonstrate the existence of isomorphic feed-forward philosophical zombies in systems similar to those found in Oizumi et al. [5]. To do so, we decompose the integrated system Y shown in Figure 6 into an isomorphic feed-forward philosophical zombie Y' of the form shown in Figure 3. The system Y, comprised of two `XNOR` gates and one `XOR` gate, clearly contains feedback between components and has $\Phi > 0$ for all states for which Φ can be calculated (Figure 6c). As in Section 2.3, the goal of the decomposition is an isomorphic relabeling of the finite-state machine representing the global behavior of the system, such that the induced logical architecture is strictly feed-forward.



t_0 \ t_1	000	001	010	011	100	101	110	111
000	0	0	0	1	0	0	0	0
001	0	0	0	0	0	1	0	0
010	0	0	0	0	0	0	1	0
011	1	0	0	0	0	0	0	0
100	1	0	0	0	0	0	0	0
101	0	0	0	0	0	0	1	0
110	0	0	0	0	0	1	0	0
111	0	0	0	1	0	0	0	0

(**a**)

(**b**)

State	Φ Value
000	1.875
001	n/a
010	n/a
011	1.875
100	n/a
101	1.875
110	1.875
111	n/a

(**c**)

Figure 6. The transition probability matrix (**a**); logical architecture (**b**); and all available Φ values (**c**) for the example system Y (n/a implies Φ is not defined for a given state because it is unreachable).

We first evenly partition the state space of Y into two blocks $B_0 = \{A, C, G, H\}$ and $B_1 = \{B, D, E, F\}$. Under this partition, B_0 transitions to B_1 and B_1 transitions to B_0, which implies the automaton representing the first coordinate in the new labeling scheme is a NOT gate. Note that this choice is not unique, as we could just as easily have chosen a different preserved partition such as $B_0 = \{A, D, E, H\}$ and $B_1 = \{B, C, F, G\}$, in which case the first coordinate would be a COPY gate; as long as the partition is preserved, the choice here is arbitrary and amounts to selecting one of several different feed-forward logical architectures—all in cascade form. For the second preserved partition, we let $P_2 = \{\{B_{00}, B_{01}\}, \{B_{10}, B_{11}\}\}$ with $B_{00} = \{C, G\}$, $B_{01} = \{A, H\}$, $B_{10} = \{B, F\}$. and $B_{11} = \{D, E\}$. The transition function for the automaton representing the second coordinate, given by the movement of these blocks, is: $\delta_{Q2'} = \{00 \rightarrow 0; 01 \rightarrow 1; 10 \rightarrow 0; 11 \rightarrow 1\}$, which is again a COPY gate receiving input from itself. The third and final partition P_3 assigns each state to its own unique block. As is always the case, this last partition is trivially preserved because individual states are guaranteed to transition to a single block. The transition function for this coordinate, read off the bottom row of Figure 7, is given by:

$$\delta_{Q3'} = \{000 \rightarrow 0; 001 \rightarrow 0; 010 \rightarrow 1; 011 \rightarrow 1; 100 \rightarrow 0; 101 \rightarrow 0; 110 \rightarrow 1; 111 \rightarrow 1\}$$

Using Karnuagh maps [31], one can identify δ_{Q3} as a COPY gate receiving input from Q_2'. With the specification of the logic for the third coordinate, the cascade decomposition is complete and the new labeling scheme is shown in Figure 7. A side-by-side comparison of the original system Y and the feed-forward system Y' is shown in Figure 8. As required, the feed-forward system has $\Phi = 0$ but executes the same sequence of state transitions as the original system, modulo a permutation of the labels used to instantiate the states of the global state-transition diagram.

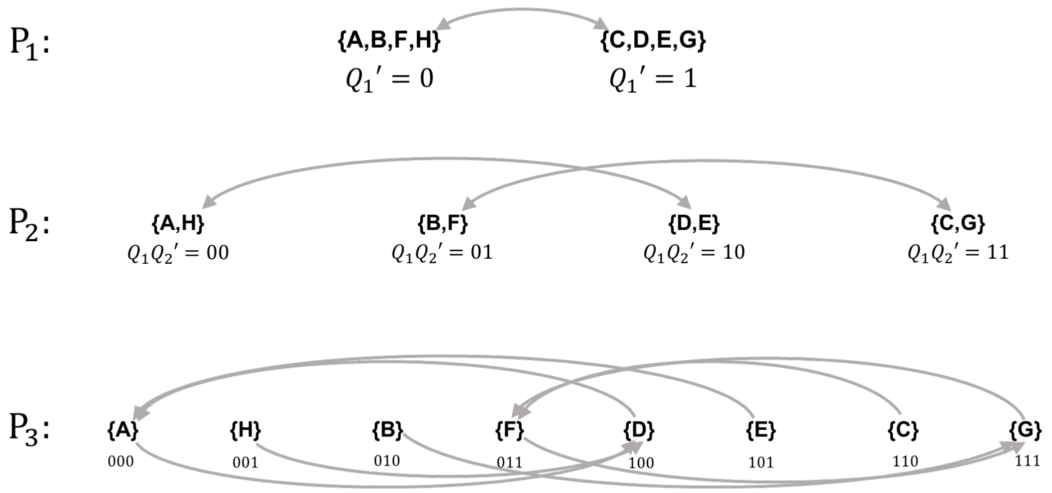
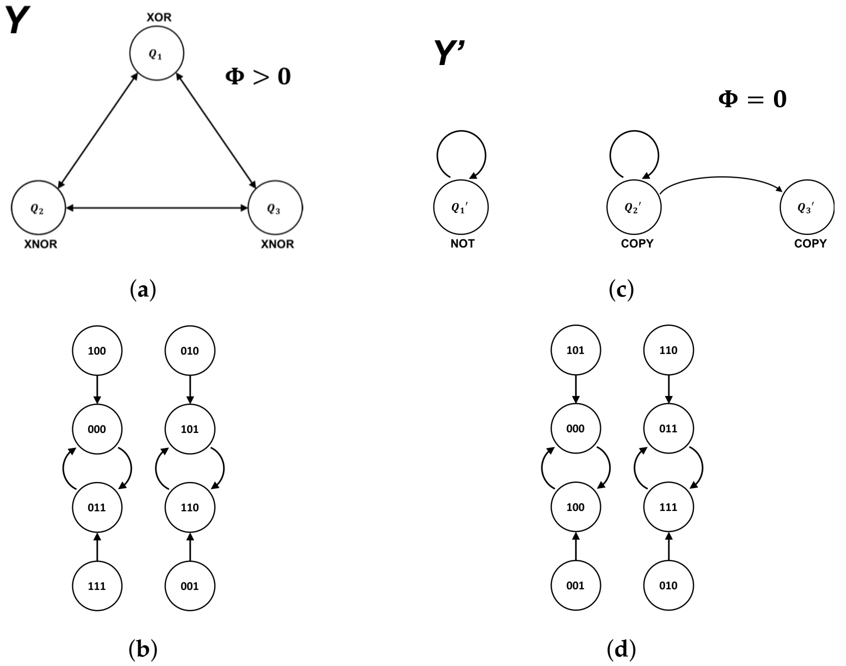


Figure 7. Nested sequence of preserved partitions used to isomorphically decompose Y into cascade form.



(**a**)

(**c**)

(**b**)

(**d**)

Figure 8. Side-by-side comparison of the feedback system Y with $\Phi > 0$ (**a**) and its isomorphic feed-forward counterpart Y' with $\Phi = 0$ (**c**). The global state-transition diagrams (**b**,**d**, respectively) differ only by a permutation of labels.

4. Discussion

Behavior is most frequently described in terms of *abstract* states/stimuli, which are not tied to a specific representation (binary or otherwise). Examples include descriptors of mental states, such as being asleep or awake, etc.: these are representations of system states that must be defined by either an external observer or internally in the system performing the computation by its own logical implementation, but are not necessarily an intrinsic attribute of the computational states themselves (e.g., these states could be labeled with any binary assignment consistent with the state transition diagram of the computation). The analysis presented here is based on this premise that such that behavior is defined by the topology of the state-transition diagram, independent of a particular labeling scheme. Indeed, it is this premise that enables Krohn–Rhodes decomposition to be useful from an engineering perspective, as one can swap between logical architectures without affecting the operation of a system in any way.

Phenomenologically, consciousness is often associated with the concept of "top-down causation", where "higher level" mental states exert causal control over lower level implementation [32]. Under this view, the "additional information" provided by consciousness above and beyond non-conscious systems is considered to be functionally relevant by affecting how states transition to other states. Typically, this is associated with a macroscale intervening on a microscale, which historically has been problematic due to the issue of supervenience, whereby a system can be causally overdetermined if causation operates across multiple scales [33]. Ellis and others described functional equivalence classes [32,34] as one means of implementing top-down causation without causal overdetermination because what does the actual causal work is a functional equivalence class of microstates, which as a class have the same causal consequences. Building on the idea of functional equivalence classes, our formalism introduces a different kind of top-down causation that also avoids issues of supervenience. In our formalism, consciousness is most appropriately thought of as a computation having to do with the topology (causal architecture) of global state transitions, rather than the labels of the states or a specific logical architecture. Thus, the computation/function describes a functional equivalence class of logical architectures that all implement the same causal relations among states, i.e. the functional equivalence class is the computational abstraction (macrostate), which can be implemented in any of a set of isomorphic physical architectures (microstates/physical implementations). There is no additional "room at the bottom" for a particular logical architecture to exert more causal influence when it instantiates a particular abstraction than another architecture instantiating the same abstraction, because the causal structure of the abstraction remains unchanged. Any measure of consciousness that changes under the isomorphism we introduce here, such as Φ, cannot therefore account for "additional information" related to executing a particular function, because of the existence of zombie systems within the same functional equivalence class.

It is important to recognize there exists an alternative perspective where one defines differences relevant to consciousness not in terms of abstract computation, but in terms of specific logical implementations, as IIT adopts. However, this does not address, in our view, whether isomorphic systems ultimately experience a phenomenological difference as there is no way to test that assumption other than accepting it axiomatically. In particular, for the examples we consider here, there is only one input signal, meaning there are not multiple ways to encode input from the environment. Therefore, there is no physical mechanism by which the environment can dictate a privileged internal representation. Instead, the choice of internal representation is arbitrary with respect to the environment and depends only on the physical constraints of the architecture of the system performing the computation. For a system as complex as the human brain, there are presumably many possible logical architectures that define an equivalence class capable of performing the same computation given the same input, differing only in how the states are internally represented (e.g., by how neurons are wired together). This could, for example, explain why humans brains all have the potential to be conscious despite differences in the particular wiring of their neurons. Why the internal representations that have evolved were selected for in the first place is likely important for understanding why consciousness emerged in the universe.

Historically within IIT, the presence of feedback is associated with *efficiency*, such that unconscious feed-forward systems, e.g., those presented by Oizumi et al. [5] and Doerig et al. [15], operate under drastically different resource constraints than their conscious counterparts with feedback. This motivates arguments for an evolutionary advantage toward efficient representation and, by proxy, Φ/consciousness [27]. Under isomorphic decomposition, however, systems with feedback can be assumed to have equivalent efficiency to their counterparts without feedback, because the size of the system and its state transitions are equivalent—demonstrating that Φ is fundamentally distinct from efficiency. This, in turn, implies there are no inherent evolutionary benefit to the presence or absence of Φ because it is not selectable as being distinctive to a particular computation an organism must perform for survival, but only how that computation is internally represented.

Given that isomorphic systems exhibit the same behavior, meaning that they take the exact same trajectory through state space, modulo a permutation of the labels used to represent the states, they can also be considered to be equally autonomous. This is because the internal states of isomorphic systems are in one-to-one correspondence (Figure 5b), meaning the presence/absence of autonomy does not affect the transitions in the global state diagram of the system (e.g., Figure 8b,d). In our examples, the future state of the system as a whole is completely determined its current state for both the zombie system and its conscious counterpart. Similarly, the future state of each individual component within a given system is completely determined by its inputs (i.e., it is a deterministic logic gate). This presents a challenge for understanding in what sense a system with $\Phi > 0$ can be said to be dictating its own future from within, while the system with $\Phi = 0$ is not, as IIT suggests. The notion of autonomy that IIT adopts to address this is one of interdependence—autonomous systems rely on bi-directional information exchange between components while non-autonomous systems do not. However, given that each component can store only one bit of information, components cannot store *where* the information came from (e.g., whether or not they are part of an integrated architecture). Since the information stored by the system as a whole is nothing more than the combined information stored by individual components, it is unclear to us why feedback between elements should result in autonomy while feed-forward connections between elements should not, given isomorphic state-transition diagrams.

This leads us to the central question of this manuscript: What is experienced as the isomorphic system with $\Phi > 0$ cycles through its internal states that is not experienced by its counterpart with $\Phi = 0$? Since in our examples the environment is not dictating the representation of the input, and all state transitions are isomorphic, the representation and therefore the logic is arbitrary so long as a logical architecture is selected with the proper input-output map under all circumstances. In light of this, our formalism suggests any mathematical measure of consciousness, phenomenologically motivated or otherwise, must be invariant with respect to isomorphic state-transition diagrams. This minimal criterion implies measurable differences in consciousness are always associated with measurable differences in the computation being performed by the system (though the inverse need not be true), which is nothing more than precise mathematical enforcement of the precedent set by Turing [14]. From this perspective, measures of consciousness should operate on the topology of the state-transition diagram, rather than the logic of a particular physical implementation. That is, they should probe the computational capacity of the system without being biased by a particular logical architecture—allowing identifying equivalence classes of physical systems that could have the same or similar conscious experience.

Our motivation in this work is to provide new roads to address the hard problem of consciousness by raising new questions. Our framework focuses attention on the fact that we currently lack a sufficiently formal understanding of the relationship between physical implementation and computation to truly address the hard problem. The logical architectures in Figure 8a,c are radically different, and yet, they perform the same computation. The fact that this computation allows a feed-forward decomposition is a consequence of redundancies that allow a compressed description in terms of a feed-forward logical architecture. There are symmetries present in the computation that allow one to take advantage of shortcuts and reduce the computational load. This, in turn, shows up as flexibility in the logical architecture that can generate the computation. In other words, the computation in question does not appear to require the maximum computational power of a three-bit logical architecture. For sufficiently complex eight-state computations, however, the full capacity of a three-bit architecture is required, as there is no redundancy to compress. Such systems cannot be generated without feedback, as the presence of feedback is accompanied by indispensable functional consequences. In this case, the *computation* is special because it cannot be efficiently represented without feedback—a relationship that can, in principle, be understood but is only tangentially accounted for in current formalisms. It is up to the community to decide if the causal mechanisms of consciousness are at the level of particular logical architectures or the computations they instantiate, our goal in this work is simply to point out where the distinction between the two sets of ideas is very apparent and clear-cut mathematically, so that additional progress can be made.

Author Contributions: Conceptualization, J.R.H. and S.I.W.; formal analysis, J.R.H.; funding acquisition, S.I.W.; investigation, J.R.H.; methodology, J.R.H.; project administration, S.I.W.; resources, S.I.W.; supervision, S.I.W.; visualization, J.R.H.; writing—original draft preparation, J.R.H. and S.I.W.; and writing—review and editing, J.R.H. and S.I.W.

Acknowledgments: The authors would like to thank Doug Moore for his assistance with the Krohn–Rhodes theorem and semigroup theory, as well as Dylan Gagler and the rest of the emergence@asu lab for thoughtful feedback and discussions. S.I.W. also acknowledges funding support from the Foundational Questions in Science Institute.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rees, G.; Kreiman, G.; Koch, C. Neural correlates of consciousness in humans. *Nat. Rev. Neurosci.* **2002**, *3*, 261. [CrossRef] [PubMed]
2. Chalmers, D.J. Facing Up to the Problem of Consciousness. *J. Conscious. Stud.* **1995**, *2*, 200–219.
3. Searle, J.R. Minds, brains, and programs. *Behav. Brain Sci.* **1980**, *3*, 417–424. [CrossRef]
4. Tononi, G. Consciousness as integrated information: A provisional manifesto. *Biol. Bull.* **2008**, *215*, 216–242. [CrossRef]
5. Oizumi, M.; Albantakis, L.; Tononi, G. From the phenomenology to the mechanisms of consciousness: Integrated information theory 3.0. *PLoS Comput. Biol.* **2014**, *10*, e1003588. [CrossRef]
6. Tononi, G.; Boly, M.; Massimini, M.; Koch, C. Integrated information theory: From consciousness to its physical substrate. *Nat. Rev. Neurosci.* **2016**, *17*, 450. [CrossRef]
7. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
8. Tononi, G. An information integration theory of consciousness. *BMC Neurosci.* **2004**, *5*, 42. [CrossRef]
9. Balduzzi, D.; Tononi, G. Integrated information in discrete dynamical systems: Motivation and theoretical framework. *PLoS Comput. Biol.* **2008**, *4*, e1000091. [CrossRef]
10. Godfrey-Smith, P. *Theory and Reality: An Introduction to the Philosophy of Science*; University of Chicago Press: Chicago, IL, USA, 2009.
11. Marcus, E. Why Zombies Are Inconceivable. *Australas. J. Philos.* **2004**, *82*, 477–490. [CrossRef]
12. Kirk, R. Zombies 2003. Available online: https://plato.stanford.edu/entries/zombies/ (accessed on 24 October 2019).
13. Harnad, S. Why and how we are not zombies. *J. Conscious. Stud.* **1995**, *1*, 164–167.
14. Turing, A. Computing Machinery and Intelligence. *Mind* **1950**, *59*, 433–433. [CrossRef]
15. Doerig, A.; Schurger, A.; Hess, K.; Herzog, M.H. The unfolding argument: Why IIT and other causal structure theories cannot explain consciousness. *Conscious. Cognit.* **2019**, *72*, 49–59. [CrossRef] [PubMed]
16. Krohn, K.; Rhodes, J. Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. *Trans. Am. Math. Soc.* **1965**, *116*, 450–464. [CrossRef]
17. Zeiger, H.P. Cascade synthesis of finite-state machines. *Inf. Control* **1967**, *10*, 419–433. [CrossRef]
18. Hopcroft, J.E.; Motwani, R.; Ullman, J.D. Automata theory, languages, and computation. *Int. Ed.* **2006**, *24*, 19.
19. Ginzburg, A. *Algebraic Theory of Automata*; Academic Press: Cambridge, MA, USA, 2014.
20. Egri-Nagy, A.; Nehaniv, C.L. Computational holonomy decomposition of transformation semigroups. *arXiv* **2015**, arXiv:1508.06345.
21. Zeiger, P. Yet another proof of the cascade decomposition theorem for finite automata. *Theory Comput. Syst.* **1967**, *1*, 225–228. [CrossRef]
22. Arbib, M.; Krohn, K.; Rhodes, J. *Algebraic Theory of Machines, Languages, and Semi-Groups*; Academic Press: Cambridge, MA, USA, 1968.
23. Shannon, C.E.; McCarthy, J. *Automata Studies (AM-34)*; Princeton University Press: Princeton, NJ, USA, 2016; Volume 34.
24. DeDeo, S. Effective Theories for Circuits and Automata. *Chaos (Woodbury, N.Y.)* **2011**, *21*, 037106. [CrossRef]

25. Maler, O. A decomposition theorem for probabilistic transition systems. *Theor. Comput. Sci.* **1995**, *145*, 391–396. [CrossRef]
26. Tegmark, M. Improved measures of integrated information. *PLoS Comput. Biol.* **2016**, *12*, e1005123. [CrossRef] [PubMed]
27. Albantakis, L.; Hintze, A.; Koch, C.; Adami, C.; Tononi, G. Evolution of Integrated Causal Structures in Animats Exposed to Environments of Increasing Complexity. *PLoS Comput. Biol.* **2014**, *10*, e1003966. [CrossRef] [PubMed]
28. Hartmanis, J. *Algebraic Structure Theory of Sequential Machines*; Prentice-Hall International Series in Applied Mathematics; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1966.
29. Egri-Nagy, A.; Nehaniv, C.L. Hierarchical coordinate systems for understanding complexity and its evolution, with applications to genetic regulatory networks. *Artif. Life* **2008**, *14*, 299–312. [CrossRef] [PubMed]
30. Mayner, W.G.; Marshall, W.; Albantakis, L.; Findlay, G.; Marchman, R.; Tononi, G. PyPhi: A toolbox for integrated information theory. *PLoS Comput. Biol.* **2018**, *14*, e1006343. [CrossRef] [PubMed]
31. Karnaugh, M. The map method for synthesis of combinational logic circuits. *Trans. Am. Inst. Electr. Eng. Part I Commun. Electron.* **1953**, *72*, 593–599. [CrossRef]
32. Ellis, G. *How Can Physics Underlie the Mind*; Springer: Berlin, Germany, 2016.
33. Kim, J. Concepts of supervenience. In *Supervenience*; Routledge: Abingdon, UK, 2017; pp. 37–62.
34. Auletta, G.; Ellis, G.F.; Jaeger, L. Top-down causation by information control: From a philosophical problem to a scientific research programme. *J. R. Soc. Interface* **2008**, *5*, 1159–1172. [CrossRef]