Causal Reasoning with Neuron Diagrams*

Martin Erwig
Oregon State University
erwig@eecs.oregonstate.edu

Eric Walkingshaw
Oregon State University
walkiner@eecs.oregonstate.edu

Abstract

The principle of causation is fundamental to science and society and has remained an active topic of discourse in philosophy for over two millennia. Modern philosophers often rely on "neuron diagrams", a domain-specific visual language for discussing and reasoning about causal relationships and the concept of causation itself. In this paper we formalize the syntax and semantics of neuron diagrams. We discuss existing algorithms for identifying causes in neuron diagrams, show how these approaches are flawed, and propose solutions to these problems. We separate the standard representation of a dynamic execution of a neuron diagram from its static definition and define two separate, but related semantics, one for the causal effects of neuron diagrams and one for the identification of causes themselves. Most significantly, we propose a simple language extension that supports a clear, consistent, and comprehensive algorithm for automatic causal inference.

1 Introduction

Identifying causal relationships is a central concern in scientific research, the judicial system, and our everyday lives. Understanding cause and effect helps us to understand the world around us, to change it, and to assign praise or blame to each others' actions. But how do we determine which events are causes of others? And what exactly does this even mean? Philosophers have studied these questions for over two millennia, dating back at least to Plato [15], and it remains an active area of research even today.

Modern philosophers have developed several different notations for discussing causation and analyzing causal relationships. The most widely used of these are *neuron diagrams* [12], which are created using a domain-specific visual language for concisely representing causal structures. Although philosophers rely on this language for precise reasoning, it has never been described in a formal way. A significant contribution of this work is therefore a formal definition of the syntax and semantics of neuron diagrams. We also discuss methods for inferring causes from neuron diagrams, demonstrate known flaws in these approaches, introduce solutions, and motivate a simple language extension to neuron diagrams. Finally, we provide a *causal semantics* for extended neuron diagrams, a new cause inference algorithm that performs better than existing algorithms.

In the next section we introduce the visual notation of neuron diagrams. In Section 3 we introduce the philosophical tool of counterfactual reasoning, which forms the basis of most modern theories of causation, and discuss its limitations. In Section 4 we discuss the transitivity of causation, where it is problematic, and how our language extension can help. We formalize the syntax and execution semantics of neuron diagrams in Section 5, and provide our causal semantics in Section 6. In Section 7 we compare our causal semantics to existing strategies when applied to some well-known causation problems. We discuss other related work in Section 8, and offer conclusions in Section 9.

2 Neuron Diagrams

Philosophers frequently structure their research around thought experiments in the form of simple (and usually morbid) stories. In this spirit, consider the well-known desert traveler problem. A traveler goes on a trip through the desert and takes a bottle of water. Two people try to kill the traveler: the first poisons the water while the second pokes a hole in the bottle. On the trip the traveler gets thirsty, tries to drink from the bottle, finds it empty, and dies of dehydration. The problem, of course, is what is the cause of the traveler's death? Who is guilty of murder? A neuron diagram for the desert traveler problem is given in Figure 1.

A neuron diagram is a directed, acyclic graph (DAG),

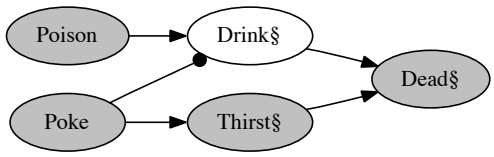

Figure 1. Desert traveler problem.

*This work is partially supported by the National Science Foundation under the grant CCF-0917092.

where each node is called a *neuron*. A neuron can either fire or not, which is indicated visually by the color of the node: either gray or white, respectively. Neurons are highly abstract and do not generally represent biological neurons. They are frequently used to represent all sorts of other real world entities, however, from events—where firing indicates the occurrence of the event and non-firing indicates its failure to occur—to the (binary) state of various objects. In our example, the *Poison* neuron can be interpreted either as the action of poisoning the water, or as the existence of poison in the water (in which case, the act of poisoning is implied but not explicitly represented). The § symbol adorning some neurons is related to our language extension described in Section 4; they can be safely ignored for now.

Neurons in a diagram can be separated into two groups. A neuron with no incoming edges (a source node in the DAG) is called *exogenous*, and its value (whether or not it fires) is predefined. A neuron that is not exogenous is said to be *endogenous*, and its value is determined by the values of its predecessors, according to some function. In our example, the *Poison* and *Poke* neurons are exogenous and all other neurons are endogenous. If we were to change the value of either or both of the exogenous neurons, the values of downstream neurons could potentially change as well.

Neurons can be connected by two different types of edges. Edges with triangular arrowheads are *stimulating* edges, those with circular arrowheads are *inhibiting* edges. A typical endogenous neuron fires if and only if it is stimulated by at least one firing neuron, and inhibited by zero firing neurons. In our example, the *Dead* neuron fires since it is stimulated by the firing *Thirst* neuron and is not inhibited by any neurons. Although the *Drink* neuron is stimulated by the *Poison* neuron, it is also inhibited by the *Poke* neuron, preventing it from firing (poking the bottle drains the liquid from the bottle, preventing it from being drunk). Note that any number of stimulating neurons can be overridden by the firing of a single inhibiting neuron.

While the informal firing semantics described above applies unless otherwise noted, it is common for creators of neuron diagrams to use or invent neurons which implement other functions as well. To demonstrate this, we present the two doctors problem, taken from [9]. In this problem, a patient is seriously ill and will die unless two different doctors, *A* and *B*, administer a treatment. Unfortunately, only doctor *A* administers the treatment, and so the patient dies. A possible neuron diagram for this problem is given in Figure 2. In this example, a thick-bordered neuron is a neuron that will fire only if two or more stimulating neurons fire. So the *Cure* neuron will fire (and the *Dead* neuron will be inhibited) only if both the *A* and *B* neurons fire.

It seems clear that doctor *B*'s inaction is the cause of death of the patient, but how do we describe this intuition more precisely? In the next section we discuss counterfactual reasoning as a foundation for identifying causation.

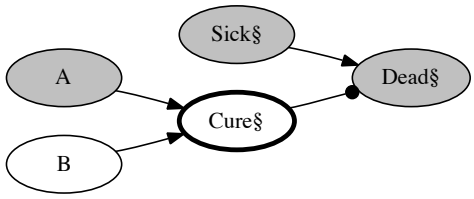

Figure 2. Two doctors problem.

3 Counterfactual Reasoning

The essence of counterfactual reasoning is considering what might have happened had things been different. In the case of the two doctors problem, we know that if doctor *B* had administered the treatment, the patient would not have died, so we conclude that *B*'s failure to administer the treatment is a cause of the patient's death. This simple idea was first explored by philosopher David Lewis [11], and forms the basis of many modern theories of causation [16, 4, 14].

Extending counterfactual reasoning to neuron diagrams is easy. To determine if the value of neuron *c* is the cause of the value of a downstream neuron *e*, simply change the value of *c* and see if the value of *e* also changes. If so, the original value of *c* is a cause of the original value of *e*. For example, the value of neuron *B* in Figure 2 is *false* (non-firing). If we change the value to *true* (firing), the value of *Cure* changes from *false* to *true*, and *Dead* changes from *true* to *false*. Thus, the non-firing of neuron *B* is a cause of the non-firing of *Cure* and the firing of *Dead*.

This approach breaks down relatively rapidly, however, and there are a few standard classes of philosophical problems that demonstrate this. The first is called (*symmetric*) *overdetermination*, and can be seen by considering a variant of the two doctors problem given in Figure 3. In this variant, neither doctor administers the cure, and the patient still dies. Intuitively, it seems that doctor *A* and doctor *B* are both at fault, and that each doctor's failure to administer the cure is a cause of the patient's death. However, notice that if we apply our basic counterfactual strategy by changing either neuron *A* or neuron *B*, the result of *Dead* is unchanged. Getting *Dead* to change requires altering *both* neurons *A* and *B*. In our formal description of causation in Section 6, we extend counterfactual reasoning to identify *sets* of neurons as counterfactual dependencies, solving the overdetermination problem. This also leads to a richer cause structure, where causes are boolean expressions of neurons. We call this extended approach *structured* counterfactual reasoning. This form is known in the philosophy literature [8, 6], but so-called "token" counterfactual reasoning is preferred for its simplicity.

A more fundamental limitation of counterfactual reasoning is revealed by the desert traveler problem from Section 2. Recall that the desert traveler died of thirst on the journey because of the hole poked in the bottle. Thus, it

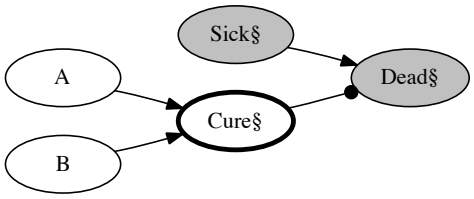

Figure 3. Overdetermined variant of the two doctors problem.

seems that only the hole-poker and not the poisoner should be considered the cause of the traveler's death. However, we can see that the *Dead* neuron is not counterfactually dependent on *Poke* alone, but only on the set including both the *Poke* and *Poison* neurons. This problem is called *asymmetric overdetermination*, or more commonly, *preemption*—the poking of the bottle preempts or "trumps" the poisoning, and so should be considered the sole cause of the traveler's death. Counterfactual reasoning, however, cannot distinguish this relationship from the symmetric overdetermination found in the two doctors variant described above.

In the next section we describe Lewis's definition of causation based on "causal chains", which gracefully copes with the problem of preemption, and forms the basis of our causal semantics presented in Section 6.

4 Transitivity of Causation

The failure of counterfactual reasoning in cases of preemption stems from a disregard for the internal structure of neuron diagrams. Twiddling inputs and observing outputs reduces neuron diagrams to black-box boolean functions, when in fact they provide much more causal information. When adapted to neuron diagrams, Lewis's causal chain description of causation [11] utilizes this information to identify causes in preemption problems that are more consistent with intuition. For example, it identifies Poke as a cause of Dead in the desert traveler problem, but not Poison.

According to Lewis, given events c, d, and e, if e is counterfactually dependent on d, and d is counterfactually dependent on c, then c is a cause of e, regardless of whether or not e is counterfactually dependent on c. In other words, Lewis considers causation to be fundamentally *transitive*. The events c, d, and e are said to form a *causal chain*. Looking again at the desert traveler problem, we see that *Dead* is counterfactually dependent on *Thirst*, *Thirst* is counterfactually dependent on *Poke*, and so we conclude that *Poke* is a cause of *Dead*, even though *Dead* is not counterfactually dependent on *Poke*. Further, we do not identify *Poison* as a cause of *Dead*; since the firing of *Dead* is not counterfactually dependent on the non-firing of *Drink*, there is no causal chain leading from *Dead* to *Poison*.

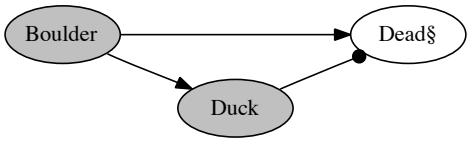

Figure 4. Boulder problem.

The causal chain approach is not without flaws. Since identifying the causes of each neuron/event relies on token counterfactual reasoning, it does not handle symmetric overdetermination well. This can be resolved by adding a logical structure to causes, which we do in our own causal semantics. More troubling for the causal chain approach, is that our intuitive sense of causation does not always seem to be transitive. A classic example of this is the boulder problem, shown in Figure 4. In this problem a boulder falls off a ledge toward a hiker below. The hiker sees the boulder crashing down, ducks, and is unharmed. Had the hiker not ducked, she would have been killed. Applying the causal chain method we see that the non-firing of *Dead* is counterfactually dependent on the firing of *Duck*. Thus, ducking is a cause of surviving—so far so good. However, *Duck* is also counterfactually dependent on *Boulder*; since *Boulder*, *Duck*, and *Dead* form a causal chain, the falling of the boulder is *also* identified as a cause of surviving. It seems very odd to say that the very thing which threatened the life of the hiker is a cause of the hiker's survival. In this case, it seems that causation is not transitive.

Hitchcock argues that examples like this demonstrate that causation is fundamentally not transitive, and reveal a fundamental flaw in Lewis's causal chains [8]. Hall takes a different approach, arguing that causation only sometimes seems to be transitive because there are really two different kinds of causation that are conflated in current research: counterfactual causation, which is not transitive, and productive causation, which is [4]. In the boulder problem, the falling of the boulder is a productive cause of the hiker's survival (essentially by the causal chain analysis), but it is not a counterfactual cause (by basic counterfactual reasoning).

The structural equations model developed by Halpern and Pearl [6, 14] retains a single definition of causation and copes gracefully with almost (see Section 7) every problem presented so far, including transitivity. In many ways, it can be considered the state of the art with regard to formal causation analysis. The structural equations approach places a much greater emphasis on problem modeling, which Hall complains ultimately amounts to little more than building the solution into the model [5]. We agree with Halpern and Pearl's claim that this is a feature and not a bug—causation is complicated and depends crucially and subtly on the internal structure of causal stories; it is therefore important to have a language which can capture this structure explicitly.

Unfortunately, the richness of structural equations does

not map well onto neuron diagrams, and the structural equation notation itself can be quite opaque. While structural equations support superior causation analyses, neuron diagrams are simpler, more concise, and easier to understand. In particular, determining if two events are related corresponds to looking for an edge between neurons in a neuron diagram vs. scanning for co-occurring variables in a potentially large set of structural equations. The continued widespread use of neuron diagrams, particularly in informal settings [9], further suggests their utility as explanatory tools and thinking aids. With a simple extension, we believe their usefulness as tools for cause inference can be significantly improved as well.

The extension we propose is to split neurons into two kinds: *laws*, decorated with a § symbol, and *actions*, which are unadorned. These allow the modeler to indicate whether a neuron should be considered a potential cause or not. Laws represent facts or relationships that are considered immutable. They often provide context or are structural in nature, and should not be considered as potential causes. Examples include the *Cure* and *Sick* neurons in the two doctors problem. In this example, we are not interested in identifying the firing of *Sick* as a cause of the patient's death. That the patient is sick is considered given; we are interested in which of the doctors is responsible for the patient's death. Similarly, we do not want to identify the non-firing of *Cure* as a cause of the patient's death, but instead the ultimate action neurons responsible for this event.

A crucial piece of our causal semantics is that the backward-directed search for causes (that is, the construction of causal chains) terminates at action neurons. This provides a simple solution to the transitivity problem. Since *Duck* is an action in the boulder problem, we will return this as the sole cause of the hiker's survival, rather than continuing the search for earlier causes and also identifying *Boulder* as a cause. Together with imposing a logical structure on causes, we are able to correctly identify causes in all of the traditionally difficult cases presented so far.

5 Formalization of Neuron Diagrams

While the nodes and edges of a neuron diagram define an abstract causal structure, the shading of neurons show the *execution* or *instantiation* of that structure for some set of inputs. For example, Figures 2 and 3 represent the same causal structure instantiated with two different settings of exogenous neurons A and B. In our formalization, we make a distinction between *neuron graphs*, which capture an abstract neuron structure, and *neuron diagrams*, which represent an execution of a neuron graph for some set of inputs. In this section we begin by formally describing neuron graphs in Section 5.1. We then describe the execution semantics of neuron graphs in Section 5.2, and combine these to describe neuron diagrams in Section 5.3.

5.1 Neuron Graphs

A core feature of neuron graphs is their extensibility. New types of neurons can be (and frequently are) invented for use in a single problem. We thus allow a neuron to implement an arbitrary boolean function on its inputs. We write $\mathbb{F}^{k,m}$ for the set of boolean functions with k inputs and m outputs, $\mathbb{B}^k \to \mathbb{B}^m$ (where $\mathbb{F}^{0,1} = \mathbb{B}$ and $\mathbb{B} = \{T, F\}$). We let $\mathbb{F} = \cup_{k>0}\mathbb{F}^{k,1}$ represent all functions with one or more inputs and exactly one output.

The abstract syntax of a neuron graph is given by a node-labeled DAG [3], $G = (N, E, \kappa, \varphi)$, where each node represents a neuron. The sources and sinks of G are denoted by $src(G)$ and $snk(G)$, respectively. Given a node $n \in N$, if $n \in src(G)$, the corresponding neuron is exogenous, otherwise it is endogenous. The mapping $\kappa : N \to K$, where $K = \{\S, !\}$, marks each neuron as either a law or action. The mapping $\varphi : N \to \mathbb{F}$ defines the function each endogenous neuron implements. We require that functions given by φ are consistent with the structure of the graph; that is, for a node n with $k > 0$ predecessors, $\varphi(n) \in \mathbb{F}^{k,1}$.

Note that we assume a fixed ordering on E. In particular, we assume that the list of predecessors of a node $n \in N$ are given in a fixed order, and we write $\pi(n)$ to denote this list. This ordering is required since $\varphi(n)$ is applied to the values of $\pi(n)$, and $\varphi(n)$ is, in general, non-commutative. A fixed ordering on E can be easily derived, for example, from the names of nodes.

Also note that the distinction between stimulating and inhibiting edges is not captured explicitly in this representation. This functionality is pushed into the function each neuron implements. A standard neuron n with two predecessors $\pi(n) = [a, b]$, where a is connected to n by a stimulating edge and b is connected to n by an inhibiting edge, is implemented by the function $\varphi(n) = \lambda(x, y).\ x \wedge \neg y$.

5.2 Firing Semantics

The *firing semantics* of a neuron graph G with k sources and m sinks is given by a function from the set $\mathbb{F}^{k,m}$ and can be defined by composing the functions of each node, following the graph structure. We write $\mathscr{F}(n)$ for the firing semantics of node n. $\mathscr{F}$ is defined inductively as follows. Note that we abbreviate a list $x_1, \ldots, x_k$ as x^k.

Let i^k (= $i_1, \ldots, i_k$) be the list of all sources (inputs) and o^m be the list of all sinks (outputs) of G. For each source i_j we define $\mathscr{F}(i_j)$ to be a simple projection onto the jth argument; that is, $\mathscr{F}(i_j) = \lambda(x^k).\ x_j$. For each node n with $l > 0$ predecessors $n_1, \ldots, n_l$, we obtain a function $\mathbb{F}^{k,m}$ that maps the k sources to a boolean value based on n's function.

$$\mathscr{F}(n) = \lambda(x^k).\ \varphi(n)(\mathscr{F}(n_1)(x^k), \ldots, \mathscr{F}(n_l)(x^k))$$

This definition says that the firing semantics of n first computes the firing-semantics values for its predecessors, then applies its own function $\varphi(n)$ to those values.

The firing semantics of the whole neuron graph G is then given by the following function ($\in \mathbb{F}^{k,m}$) that maps a k-tuple of input values to an m-tuple boolean output.

$$\mathscr{F}(G) = \lambda(x^k).(\mathscr{F}(o_1)(x^k), \ldots, \mathscr{F}(o_m)(x^k))$$

5.3 Neuron Diagrams

A neuron graph defines a structure from which neuron diagrams can be derived by assigning values to the source nodes and using the firing semantics to propagate those values to all other nodes in the graph. Therefore, we define a neuron diagram as a node-labeled DAG, $D = (N, E, \kappa, \phi)$, where $\phi : N \to \mathbb{B}$ defines the value of each node.

For each neuron graph with k sources, the firing semantics gives rise to 2^k neuron diagrams. These are obtained by applying the firing semantics to each possible boolean input k-tuple. We capture this fact through the mapping $\mathscr{D}$ as follows. First, we define the instantiation of a neuron graph to a neuron diagram for one particular input tuple b^k. Since the nodes, edges, and node kinds do not change, all we have to do is define the node labeling ϕ, which can be directly obtained by the firing semantics, leading to the following definition.

$$\mathscr{D}((N, E, \kappa, \varphi), b^k) = (N, E, \kappa, \phi)$$
$$\text{where } \phi(n) = \mathscr{F}(n)(b^k)$$

Next we define the set of all neuron diagrams that can be instantiated from one neuron graph as follows.

$$\mathscr{D}(G) = \{\mathscr{D}(G, b^k) \mid b^k \in \mathbb{B}^k\}$$

6 Formalization of Causation

In Sections 3 and 4 we introduced counterfactual reasoning and cause propagation as pieces of our approach to cause inference. In this section we formalize and apply these ideas to develop a causal semantics for neuron diagrams. In Section 6.1, we begin by formalizing the notion of counterfactual dependency. We formalize the relationship between counterfactuals and causes in Section 6.2, and define the structure of causes. We use these definitions in the description of the causal semantics in Section 6.3.

6.1 Counterfactual Dependencies

We begin by observing that the value of a neuron n in some diagram $D \in \mathscr{D}(G)$ can be obtained by applying $\varphi(n)$ to the values of n's predecessors, a fact which is expressed by the following lemma.

Lemma 1 $D \in \mathscr{D}(G) \wedge \pi(n) = [n_1, \ldots, n_l] \implies$
$\phi(n) = \varphi(n)(\phi(n_1), \ldots, \phi(n_l))$

This follows directly from the definitions of $\mathscr{F}$ and $\mathscr{D}$ in the previous section. Note that in the above and throughout this section, we assume that symbols are implicitly defined in the obvious way; that is, in Lemma 1, φ is the node-to-function map in G, and ϕ is the node-to-value map in D.

To simplify the following discussion and technical development, we introduce several notational conventions. First, we write $n{:}b$ to express the fact that $\phi(n) = b$. Second, we associate a neuron's function with its name and write more succinctly $n(b^l)$ instead of $\varphi(n)(b^l)$. Third, we introduce the concept of an *argument record*, where the arguments to a function are annotated by the corresponding neuron names. Putting these all together, we can write $n(n_1{:}b_1, \ldots, n_l{:}b_l)$ instead of $\varphi(n)(\phi(n_1), \ldots, \phi(n_l))$.

An argument record is essentially a mapping from names to values. Each neuron n in a neuron diagram has an associated argument record ρ_n, which is given by the names and values of its predecessor nodes; that is, $\rho_n = (n_1{:}b_1, \ldots, n_l{:}b_l)$ where $\pi(n) = [n_1, \ldots, n_l]$. An *alteration* of an argument record is a variation in which the names of the record's arguments are the same, but some subset of the boolean values are changed. An alteration of record ρ_n is given by a subset of its argument names $M \subseteq \pi(n)$, and is defined as $\rho_n/M = (n_1{:}\tilde{b}_1, \ldots, n_l{:}\tilde{b}_l)$ where $\tilde{b}_i = \neg b_i$ if $n_i \in M$ and $\tilde{b}_i = b_i$ otherwise.

Argument records provide a convenient mechanism for identifying counterfactual dependency. Recall that the basic idea of counterfactual reasoning is to try to find combinations of inputs such that when the values of those inputs are changed, the result also changes. Thus, we say that neuron n is *counterfactually dependent* (or *cf-dependent* for short) on M if $n(\rho_n/M) \neq n(\rho_n)$. We say that n is *minimally cf-dependent* (or *mcf-dependent*) on M if it is cf-dependent on M and if it is not cf-dependent on any proper subset $M' \subset M$. Finally, it is possible for n to be mcf-dependent on several different subsets of inputs, and we write $n \leftarrow \{M_1, \ldots, M_n\}$ to express that neuron n is mcf-dependent on each $M_i \in \{M_1, \ldots, M_n\}$.

As an example, recall the diagram of the desert traveler problem in Figure 1. Since $Drink(Poison{:}T, Poke{:}T) = F$ and $Drink(Poison{:}T, Poke{:}F) = T$, in this diagram, *Drink* is cf-dependent on $\{Poke\}$. Since no other alterations of the argument record produce a change in the output, *Drink* is also mcf-dependent on $\{Poke\}$ and this is its only mcf-dependency. So we have $Drink \leftarrow \{\{Poke\}\}$.

It is important to note that (m)cf-dependency is a property of a neuron within a neuron diagram, not a neuron graph. It is common for the same neuron to have different cf-dependencies in two different diagrams instantiated from the same graph. Sometimes a neuron is cf-dependent on the same neurons in two different diagrams, but in a different way. These differences are revealed by the structure of the dependencies. To demonstrate this, recall the instance of the two doctors problem in Figure 3. In this example, $\varphi(Cure) = \wedge$, and $Cure(A{:}F, B{:}F) = F$. Since we must change both *A and B* to change the outcome of *Cure*, *Cure* is

mcf-dependent on the set $\{A,B\}$—that is, $Cure \leftarrow \{\{A,B\}\}$. If we consider instead the instance in which A:T and B:T, then $Cure(A{:}T,A{:}T) = T$. Now, *Cure* will change if either A *or* B changes, so we have $Cure \leftarrow \{\{A\},\{B\}\}$.

Clearly, the nested set structure of cf-dependencies is interpreted as a disjunction of conjunctions. To emphasize this, we replace the set notation with a corresponding boolean-formula notation. We combine neurons within a dependency with $\wedge$ and combine alternative dependencies with $\vee$. Moreover, we include the value labels of neurons to emphasize the context in which dependencies hold. Thus, for the above examples we obtain the following notation.

$$\begin{aligned} Drink{:}F &\leftarrow Poke{:}T \\ Cure{:}F &\leftarrow A{:}F \wedge B{:}F \\ Cure{:}T &\leftarrow A{:}T \vee B{:}T \end{aligned}$$

In general, the function of a neuron n, $\varphi(n)$, determines a corresponding function $\overleftarrow{n}$ that maps arguments neurons into a structure describing the counterfactual dependency based on the values for the argument neurons. For our examples, we have:

$$\begin{aligned} \overleftarrow{Drink}_{TT}(Poison,Poke) &= Poke \\ \overleftarrow{Cure}_{FF}(A,B) &= A \wedge B \\ \overleftarrow{Cure}_{TT}(A,B) &= A \vee B \end{aligned}$$

The notion of cause is directly linked to cf-dependencies. In the next section we make this relationship explicit.

6.2 Relationship of Counterfactuals and Causes

The relationship between counterfactuals and causes is best seen by example. Consider again the overdetermined two doctors problem, where $Cure{:}F \leftarrow A{:}F \wedge B{:}F$. In other words, both A *and* B must be changed to change *Cure*. In turn, this means that either A *or* B alone is a sufficient *cause* of *Cure*. Similarly, for the instance where $Cure{:}T \leftarrow A{:}T \vee B{:}T$, the value of *Cure* will change if either of its two predecessors change, so only together are A and B a sufficient cause of *Cure*.

Converting an arbitrary cf-dependency into a cause requires considering three cases. First, when n is mcf-dependent on a single node n', we say that n' is the *cause* of n, written $n' \rightsquigarrow n$. Second, if n is mcf-dependent on a conjunction of neurons $n_1 \wedge \ldots \wedge n_k$, then all of these neurons must change to change n, so any one of $n_1,\ldots,n_k$ alone is a sufficient cause of n. The corresponding cause is then a disjunction of neurons, written $n_1 \vee \ldots \vee n_k \rightsquigarrow n$. Third, if n is mcf-dependent on a disjunction of neurons $n_1 \vee \ldots \vee n_k$, then the changing of any single neuron will change n, so only together are they a sufficient cause of n. The corresponding cause is therefore a conjunction of the original neurons, written $n_1 \wedge \ldots \wedge n_k \rightsquigarrow n$.

Finally, we have to formulate these relationships for the general case when we are already given a (complex) cause of a neuron n. Such a cause is obtained through the function $\overleftarrow{n}$ and yields a disjunction of conjunctions of causes. We regroup such a nested cause as follows. We add each conjunct from the first disjunct to each regrouped cause obtained from the remaining disjunction. Formally, this can be expressed by the counterfactual conversion function $\lfloor\cdot\rfloor$.

$$\begin{aligned} \lfloor n \rfloor &= n \\ \lfloor (c_1 \wedge \ldots \wedge c_k) \vee C \rfloor &= c_1 \wedge C_1 \vee \ldots \vee c_k \wedge C_1 \vee \ldots \\ &\quad \vee c_1 \wedge C_m \vee \ldots \vee c_k \wedge C_m \\ &\quad \text{where } C_1 \vee \ldots \vee C_m = \lfloor C \rfloor \end{aligned}$$

With this transformation, the general relationship between counterfactuals and causes can be captured in the following lemma.

Lemma 2 (Counterfactual-Causation Correspondence) $n \leftarrow c \iff \lfloor c \rfloor \rightsquigarrow n$

6.3 Causal Semantics

With the direct correspondence between counterfactuals and causes, we can nearly define a function $\mathscr{C}$ that computes the causes for an arbitrary neuron in a diagram. However, we must still consider the distinction between law and action neurons. The description in Section 4 leads to the following definition: Each action is its own cause, but the cause of a law is obtained from the causes of its predecessors through counterfactual conversion.

$$\mathscr{C}(n) = \begin{cases} n{:}\phi(n) & \text{if } \kappa(n) = ! \\ \lfloor \overleftarrow{n}_{\phi(n^k)}(\mathscr{C}(n^k)) \rfloor & \text{if } \kappa(n) = \S \wedge \pi(n) = [n^k] \end{cases}$$

Here we write more succinctly $\phi(n^k)$ for the sequence of boolean values $\phi(n_1),\ldots,\phi(n_k)$ and, similarly, $\mathscr{C}(n^k)$ for the sequence of predecessor causes $\mathscr{C}(n_1),\ldots,\mathscr{C}(n_k)$.

7 Evaluation

In this section we compare our causal semantics to other well-known cause inference algorithms. We present a table for each of the examples used throughout this paper: (1) the two doctors problem, (2) the desert traveler problem, and (3) the boulder problem. Each table corresponds to the neuron graph for one of these problems, each row to a neuron diagram instantiated from that graph, and each column to a cause inference algorithm. A row is labeled according to the settings of the exogenous action neurons in the diagram, where the values are assigned from top to bottom. For example, the *TF* row in Table 1 (two doctors problem) corresponds to the diagram in Figure 2 where the firing status of neuron A is set to *true*, and neuron B is set to *false*. Exogenous laws are assumed to be fixed, since studying the alternatives yields no interesting results.

The columns are labeled as follows:

	CF	CC	SE	Hall	SCF	$\mathscr{C}$
FF	$\{\}$	$\{\}$	$\{A,B\}$	$\{\}$	$A \vee B$	$A \vee B$
FT	$\{A\}$	$\{A\}$	$\{A\}$	$\{A\}$	A	A
TF	$\{B\}$	$\{B\}$	$\{B\}$	$\{B\}$	B	B
TT	$\{A,B\}$	$\{A,B\}$	$\{A,B\}$	$\{A,B\}$	$A \wedge B$	$A \wedge B$

Table 1. Two doctors problem causes.

	CF	CC	SE	Hall	SCF	$\mathscr{C}$
FF	$\{P,K\}$	$\{P,K\}$	$\{P,K\}$	$\{P,K\}$	$P \wedge K$	$P \wedge K$
FT	$\{K\}$	$\{K\}$	$\{K\}$	$\{K\}$	K	K
TF	$\{P\}$	$\{P,K\}$	$\{P\}$	$\{P\}$	P	$P \wedge K$
TT	$\{\}$	$\{K\}$	$\{K\}$	$\{K\}$	$P \vee K$	K

Table 2. Desert traveler problem causes.

- CF - (token) counterfactual reasoning as described in Section 3 and originally formulated by Lewis in [11].
- CC - Lewis's causal chains, described in Section 4 and also originally in [11].
- SE - structural equations as described by Halpern and Pearl in [6].
- Hall - an approach sketched by Hall in [5] to deal with perceived weaknesses of structural equations.
- SCF - structured counterfactual reasoning, formalized in Sections 6.1 and 6.2.
- $\mathscr{C}$ - our causal semantics, defined in Section 6.3.

For the counterfactual algorithms, we perform counterfactual reasoning on exogenous actions. For structural equations, we consider a direct translation of each neuron diagram into a structural equations model. Hall's approach is described only informally for certain classes of neuron diagrams and so required some extrapolation on our part. Finally, we filter all laws out of causes identified by any method. This essentially retrofits our action/law extension to the existing algorithms, enabling a fairer and more direct comparison with our own causal semantics.

Table 1 shows the identified causes for the two doctors problem. Incorrectly identified causes are shown in gray. Note the flat structure of causes identified by methods from the philosophy literature. This is especially significant in the two doctors problem—although structural equations identifies the correct set of causal events for both the *FF* and *TT* diagrams, the causes in each case appear identical while they are distinct in our results. Intuitively, since *both* doctors' treatments are required to prevent the patient from dying, *both* are the cause of the patient's survival in the *TT* case. In the *FF* case, since *either* doctor's failure to treat is sufficient for the patient's death, *either* doctor can be considered the cause. Several researchers have acknowledged this distinction [8, 6], but it has not been incorporated into philosophical accounts of causation.

The other interesting feature of Table 1 is the failure of three approaches in the *FF* case. This is the symmetric overdetermination problem described in Section 3, for which the correct causes can only be identified by methods that can reason about more than one neuron at a time.

The results of the desert traveler problem are shown in Table 2. *P* represents the *Poison* neuron, and *K* represents *Poke*. Note the failure of both counterfactual reasoning approaches in the *TT* case. This demonstrates the classic preemption problem, which pure counterfactual reasoning cannot distinguish from overdetermination.

	CF	CC	SE	Hall	SCF	$\mathscr{C}$
F	$\{\}$	$\{B\}$	$\{B\}$	$\{B\}$	none	B
T	$\{\}$	$\{B,D\}$	$\{D\}$	$\{B,D\}$	none	D

Table 3. Boulder problem causes.

Our coding of the results for the *TF* diagram is potentially contentious. In this instance of the desert traveler problem, the poisoner puts poison in the bottle but the poker does not poke the bottle; the traveler eventually drinks the poison and dies. Several of the methods identify only *Poison* as the sole cause of death, and this seems to agree with intuition. It also seems consistent with the *FT* and *TT* cases—when the traveler dies of thirst, *Poke* alone is the cause of death. However, we believe that intuition is incorrect here, and that this is a rare case where the structural equations approach fails (the only such failure in our survey). The case where the traveler dies of poison is fundamentally different from the cases where the traveler dies by poking since the problem is asymmetric—poking always trumps poisoning. When the traveler dies of thirst, it does not matter if we would have poisoned or not, the traveler will still die of thirst. But when the traveler dies by poisoning, if we would have poked, the traveler would have instead died of thirst. Thus, the non-poking in the *TF* case *is* a significant factor in the cause of the traveler's death, and so $Poison{:}T \wedge Poke{:}F \rightsquigarrow Dead{:}T$.

Chockler and Halpern distinguish causality from the related concepts of blame and responsibility [1]. We believe that this is related to the disagreement of intuition and actual causation in the *TF* desert traveler example. In this case, we argue that the poisoner is the only one *to blame* for the traveler's death, but this does not preclude the non-poking from being a *cause*. Thus, blame seems to involves a value judgment while cause does not. A different assignment of real-world meanings to the neurons in a neuron diagram could yield different blames, but the causes would be unchanged.

The final example is the boulder problem, the results of which are given in Table 3. Only structural equations and our causal semantics correctly handle the transitivity problem in the case where the boulder falls. Both counterfactual

reasoning strategies fail on both diagrams since the hiker does not die in either case. Hall calls this counterfactual-foiling pattern *short-circuiting* [5].

In addition to the examples above, we have tested our causal semantics against 13 problematic test cases presented by Hitchcock in [10]. Some of these examples involve atypical "switch" neurons with non-local effects; we remodeled these using standard neurons. Our approach identifies the same causes as structural equations in all cases.

8 Related Work

The works of Lewis [11, 12], Hall [5], and Halpern and Pearl [6, 14] are most closely related to our own, and have been discussed extensively throughout the paper. In this section we continue this discussion and briefly acknowledge related work in other fields as well.

We have so far considered the study of causation to be the realm of philosophers, but it is also an active area of research in the field of artificial intelligence (in fact, Halpern and Pearl are computer scientists). Identifying and modeling causal relationships is critical to planning, learning, knowledge representation, fault detection, and many other problems in AI [14]. Neuron diagrams are not really suitable for large, complex artificial intelligence problems. While there is no principle limitation to their scalability, they contain no abstraction or modularity mechanisms. Additionally, most AI research focuses on probabilistic causation, which neuron diagrams do not represent. Other visual notations have been developed for explaining probabilistic causation systems, however, for example in [2]. Also relevant is the work of Helmert [7], who develops a graph-based representation for causal reasoning in planning problems.

Returning to the philosophy literature, the structural equations approach includes its own graph-based notation [14], but this is quite impoverished compared to neuron diagrams. A "causal graph" is associated with a model, but indicates only which variables are related; *how* they are related is defined only by the associated structural equations. In earlier work, Pearl describes a stand-alone graphical notation for describing causal structures and an associated cause inference algorithm [13].

Neuron diagrams have become a lingua franca for causation research, but the notation is not without detractors. Hitchcock discusses several shortcomings of neuron diagrams, including both usability and technical issues. Our work resolves several of the technical issues. For example, Hitchcock argues that the ability to add new types of neurons ad hoc makes neuron diagrams: (1) difficult to reason about, and (2) less useful as exploratory tools since configurations of neurons cannot be enumerated. With our formal definition, reasoning about new neuron types is easy since all neurons are represented generically as arbitrary functions. It is also straightforward to list all boolean functions for a certain number of inputs, allowing us to enumerate all configurations of a finite number of neurons.

9 Conclusions

The two major contributions of this paper are: (1) a *formalization*, for the first time, of the syntax and semantics of neuron diagrams, and (2) a new *algorithm* for cause inference over neuron diagrams that performs better than existing alternatives. These theoretical contributions create a foundation for future research on the development and analysis of causal structures, and for the creation of supporting tools. This paper also demonstrates that formal approaches to (visual) language design can contribute significantly to the improvement of such languages.

References

[1] C. Chockler and J. Halpern. Responsibility and Blame: A Structural Model Approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[2] T. Dean and K. Kanazawa. A Model for Reasoning About Persistence and Causation. *Computational Intelligence*, 5(3):142–150, 1989.

[3] M. Erwig. Abstract Syntax and Semantics of Visual Languages. *Journal of Visual Languages and Computing*, 9(5):461–483, 1998.

[4] N. Hall. Two Concepts of Causation. In J. Collins, N. Hall, and L. Paul, editors, *Causation and Counterfactuals*, pages 225–276. MIT Press, Cambridge, MA, 2004.

[5] N. Hall. Structural Equations and Causation. *Philosophical Studies*, 132:109–136, 2007.

[6] J. Halpern and J. Pearl. Causes and Explanations: A Structural-Model Approach, Part I: Causes. *British Journal of Philosophy of Science*, 56(4):843–887, 2005.

[7] Helmert, M. A Planning Heuristic Based on Causal Graph Analysis. In *Int. Conf. on Automated Planning and Scheduling*, pages 161–170, 2004.

[8] C. Hitchcock. The Intransitivity of Causation Revealed in Equations and Graphs. *The Journal of Philosophy*, 98(6):273–299, 2001.

[9] C. Hitchcock. What's Wrong with Neuron Diagrams. In J. K. Campbell, M. O'Rourke, and H. Silverstein, editors, *Causation and Explanation*, pages 69–92. MIT Press, Cambridge, MA, 2007.

[10] C. Hitchcock. Structural Equations and Causation: Six Counterexamples. *Philosophical Studies*, 144:391–401, 2009.

[11] D. Lewis. Causation. *Philosophy*, 70(17):556–567, 1973.

[12] D. Lewis. Postscripts to 'Causation'. In *Philosphical Papers, Vol. II*, pages 196–210. Oxford University Press, New York, NY, 1987.

[13] J. Pearl. Causal Diagrams for Empirical Research. *Biometrika*, 82(4):669–688, 1995.

[14] J. Pearl. *Causality: Models, Reasoning and Inference (2nd ed.)*. Cambridge University Press, Cambridge, UK, 2009.

[15] D. Ruben. *Explaining Explanation*. Routledge, London, UK, 1990.

[16] J. Woodward. *Making Things Happen*. Oxford University Press, New York, NY, 2003.