In Proceedings, IJCAI'91, Sydney, Australia,
August 24-30, 1991.

TECHNICAL REPORT
R-153
April, 1991

DIRECTED CONSTRAINT NETWORKS: A RELATIONAL FRAMEWORK FOR CAUSAL MODELING*

Rina Dechter
Information & Computer Science
University of California, Irvine
Irvine, CA. 92717
U.S.A.

Judea Pearl
Computer Science Department
University of California, Los Angeles
Los Angeles, CA. 90024
U.S.A.

Abstract

Normally, constraint networks are undirected, since constraints merely tell us which sets of values are compatible, and compatibility is a symmetrical relationship. In contrast, causal models use directed links, conveying cause-effect asymmetries. In this paper we give a relational semantics to this directionality, thus explaining why prediction is easy while diagnosis and planning are hard. We use this semantics to show that certain relations possess intrinsic directionalities, similar to those characterizing causal influences. We also use this semantics to decide when and how an unstructured set of symmetrical constraints can be configured so as to form a directed causal theory.

1. Introduction

Finding a solution to an arbitrary set of constraints is known to be an NP-hard problem. Yet certain types of constraint systems, usually those describing causal mechanisms, manage to escape this limitation and permit us to construct a solution in an extremely efficient way. Consider, for example, the task of computing the output of an acyclic circuit consisting of a large number of logical gates. In theory, each gate is merely a constraint that forbids certain input-output combinations from occurring, and the task of computing the output of the overall circuit (for a given combination of the circuit inputs) is equivalent to that of finding a solution to a set of constraints. Yet contrary to the general constraint problem, this task is remarkably simple; one need only trace the flow of causation and propagate the values of the intermediate variables from the circuit inputs down to the circuit output(s). This forward computation encounters none of the difficulties of the general constraint-satisfaction problems, thus exemplifying the simplicity inherent to causal predictions.

*This work was partially supported by the National Science Foundation, Grant #IRI-8821444 and by the Air Force Office of Scientific Research, Grant #AFOSR-90-0136.

The aim of this paper is to identify and characterize the features that render this class of problems computationally efficient, thus explaining some of the reasons that causal models are so popular in the organization of human knowledge. Note that this efficiency is asymmetric; it only characterizes the forward computation, but fails to hold in the backward direction. For instance, the problem of finding an input combination that yields a given output (a task we normally associate with planning or diagnosis) is as hard as any constraint satisfaction problem. Thus, the second aim of our analysis is to explain how a system of constraints, each defined in terms of the totally symmetric relationship of compatibility, can give rise to such profound asymmetries as those attributed to cause-effect or input-output relationships. At first glance, we might be tempted to attribute the asymmetry to the functional nature of the constraints involved. However, functional dependency in itself cannot explain the directional asymmetry found in the analysis of causal mechanisms such as the logic circuit above. Imagine a circuit containing some faulty components, the output of which may attain one of *several* values. The constraints are no longer functional, yet the asymmetry persists; finding an output compatible with a given input is easy while finding an input compatible with a given output is hard. This asymmetry between prediction and planning seems to be a universal feature of all systems involving causal mechanisms [Shoham, 1988], a feature we must emulate in defining causal theories.

Our starting point is to formulate a necessary and sufficient condition for a system of constraints to exhibit a directional asymmetry similar to that characterizing causal organizations. Basically, the criterion is that of **modularity**: there should exist an ordering of the variables in the system such that imposing constraints on later variables would not further constrain earlier variables. Intuitively, it captures the understanding that predictions are useless for diagnosis; e.g., given a set of findings, we cannot improve the accuracy of our diagnosis by concentrating our analysis on the patient's prospects for recovery. Likewise, in the context of the logic circuit example, modularity asserts that if we wish to add a new gate, then, as long as we do not connect to its output, we can add this gate anywhere in the circuit without perturbing the circuit's behavior. Starting with modularity as a definition of causal theories (Section 2), we show[1] that it is tantamount to enabling **backtrack-free** search (for a feasible solution) along any natural ordering of the theory. We then explore methods of constructing causal specifications for a given relation, that is, specifications that permit objects from the relation to be retrieved backtrack-free along some ordering. Such methods are investigated along two dimensions: inductive and pragmatic. Along the inductive dimension (Section 3), we observe the tuples of some relation ρ, and we seek to represent this set of observations by a causal theory that is as *simple* as possible. We provide a formal definition of simplicity and show that together with the insistence on backtrack-free predictions, it leads to a natural definition of *intrinsic directionality*, matching our perception of causal directionality in logical circuits and other physical devices.

Along the pragmatic dimension (Section 4), we start with an unordered collection of constraint specifications, which might represent some stable physical laws, and we seek an ordering of the variables such that the overall system constitutes a causal theory. Clearly, not every system of constraints can turn causal by a clever ordering of the variables. The criterion for the existence of such an ordering depends on both the nature of the constraints and the topology of the subsets of variables upon which the constraints are specified. Some constraint systems are amiable to causal ordering by virtue of their topology alone, ***regardless*** of the content of the individual constraints. These are called acyclic constraint systems, originally studied in the literature of relational databases, [Beeri et al., 1983]. In contrast, Section 4 ascribes causal ordering to a more general set of topologies, but imposes special requirements on the character of the individual constraints.

Our basic requirement for a k-variable constraint to qualify as a description of a primitive causal mechanism, is that at least one set of $k-1$ variables must behave as **inputs** (or **causes**) relative to the remaining k^{th} variable (to be regarded as an **output** or an **effect**), that is, no value combination of these $k-1$ variables can be forbidden, and each such combination must be compatible with at least one value of the k^{th} variable. Additionally, in order for the system as a whole to act as a causal system, mechanisms must be ordered in a way that prevents conflicts among their predictions, hence, we require that no two constraints should designate the same variable as an output. We provide effective procedures for: (1) deciding if such an ordering exists and, (2) identifying such ordering whenever possible. The ordering found can be used to facilitate search and retrieval, and are similar to those used to describe the operation of physical devices [Kuipers, 1984; Iwasaki and Simon, 1986; de-Kleer and Brown, 1986].

[1]Proofs can be found in [Dechter and Pearl, 1991].

2. Definitions and Preliminaries: Constraint Specifications and Causal Theories

Definition 1 (Constraint Specification): A **constraint specification** (CS) consists of a set of n variables $X = \{X_1, \ldots, X_n\}$, each associated with a finite domain, $dom_1, \ldots, dom_n$, and a set of constraints $\{C_1, C_2, \ldots, C_t\}$ on subsets of X. Each **constraint** C_i is a relation on a subset of variables $S_i = \{X_{i_1}, \ldots, X_{i_j}\}$, namely, it defines a subset of the Cartesian product of $dom_{i_1} \times \ldots \times dom_{i_j}$. The **scheme** of a CS is the set of subsets on which constraints are defined, $scheme(CS) = \{S_1, S_2, \ldots, S_t\}$, $S_i \subseteq X$, and each such subset is called a **component**. A **solution** of a given CS is an assignment of values to the variables in X such that all the constraints in the CS are satisfied. A constraint specification CS is said to define an **underlying relation** $rel(CS)$, consisting of all the solutions of CS.

Definition 2 (Causal Theories): Given a constraint specification CS, its underlying relation $\rho = rel(CS)$, and an ordering $d = (X_1, X_2, \ldots, X_n)$, we say that a CS is a **causal theory** (of ρ) relative to d if for all $i \geq 1$ we have

$$\Pi_{X_1, \ldots, X_i}(\rho) = \underset{j(i)}{\bowtie} C_j \qquad (1)$$

where

$$j(i) = \{j : S_j \subseteq \{X_1, \ldots, X_i\}\}. \qquad (2)$$

$\Pi_{X_1, \ldots, X_i}(\rho)$ denotes the projection of ρ on $\{X_1, \ldots, X_i\}$, that is, the set of all subtuples $(x_1, \ldots, x_i)$ for which an extension $(x_i, \ldots, x_i, x_{i+1}, \ldots, x_n)$ exists in ρ, and $\bowtie$ is the *join* operator. Any pair $<d, CS>$ satisfying (1) will be called a **causal theory** (of ρ).

Although condition (1) may seem hard to verify in practice, it nevertheless provides an operational definition for causal theories. To test whether a given CS is causal relative to ordering d, we need to find the set of solutions to the given CS, project back these solutions on the strings of variables $X_1, X_2, \ldots, X_i$, $1 \leq i \leq n$, then check whether each such projection coincides exactly with the set of solutions to a smaller CS, one consisting of only those constraints that are defined on variables taken from $\{X_1, \ldots, X_i\}$. In Section 4 we will show that certain types of specifications possess syntactic features that render them inherently causal, in no need of the elaborate test prescribed by (1). For example, the specifications provided by a collection of logic gates always constitutes a causal theory relative to any ordering compatible with their standard assembly in acyclic circuits (i.e., no variable can serve as an output of two different gates). Similarly, linear inequalities and propositional clauses, under certain conditions, can be assembled into causal theories by finding appropriate orderings of the variables.

From a conceptual viewpoint, Definition 2 formalizes the notion of modularity (see Introduction) and can be given the following temporal interpretation. If we view the variables $X_1, \ldots, X_i$ as past events, the variables $X_{i+1}, \ldots, X_n$ as future events, and the constraints as physical laws, then Eq. (1) asserts that the permissible set of past scenarios is not affected by laws that pertain only to future events. In other words, the set of scenarios we get by ignoring future constraints will remain valid after including such constraints in the analysis. This interpretation is indeed at the very heart of the notion of causation, and is closely related to the principle of *chronological ignorance* described in [Shoham, 1988], although Shoham's definition of causal theories insists on functional dependencies.

We shall now show that causal theories as defined by (1) yield a computationally effective scheme of encoding relations; it guarantees that the tuples of these relations can be generated systematically, without search, by simply instantiating variables along the natural ordering of the theory.

Definition 3 (Backtrack-free): We say that a CS is **backtrack-free** along ordering $d = (X_1, \ldots, X_n)$ if for every i and for every assignment $x_1, \ldots, x_i$ consistent with $\{C_j : S_j \subseteq \{X_1, \ldots, X_i\}\}$ there is a value x_{i+1} of X_{i+1} such that $x_1, \ldots, x_i, x_{i+1}$ satisfies all the constraints in $\{C_j : S_j \subseteq \{X_1, \ldots, X_{i+1}\}\}$. In other words, a CS is backtrack-free w.r.t. d if $rel(CS)$ can be recovered with no dead-ends along the order d.

Definition 3 is an extension of the standard notion of backtrack-free originally stated for binary constraints [Freuder, 1982], and later related to directional consistency [Dechter, 1990]. Note that, given a constraint C_i on a subset S_j of variables, definition 3 does not allow testing whether some partial instantiation of S_j is compatible with C_j. It is possible to weaken this restriction by considering all the constraints projections as part of the problem's scheme. In this paper we do not consider such projections; nevertheless, our analysis is extensible to that case as well.

Theorem 1: A constraint specification CS is backtrack-free along an ordering d if and only if it is causal relative to d.

In the practice of causal modeling, it is common to depict the structure of causal theories using directed acyclic graphs (dags), not total orders. Each such dag, called a causal model, indicates the existence of direct causal influences among sets of variables, but does not specify the precise nature of the influences. We will next give a formal definition of such models, and then explore what properties of the underlying relation are portrayed by the topology of the dag.

Definition 4 (Dags and Families): Given a directed acyclic graph (dag) D, we say that an ordering $d = (X_1, \ldots, X_n)$ of the nodes in the graph **respects** D if all edges in D are directed from lower to higher nodes of d. A dag D defines a set of n **families** $F_1, \ldots, F_n$, each family F_i is a subset consisting of a **son** node, X_i, and all its **parent** nodes, P_i, which are those directed towards X_i in D.

Definition 4′ (Characteristic dag): The **characteristic dag**, D, of the pair (d, CS) is constructed as follows: For each component S_j in *scheme(CS)*, designate the latest variable (according to d) in S_j as a sink and direct the other variables in S_j towards it.

Figure 1 shows the characteristic dag of a CS defined on the subsets AB, AC, BD, CD, CE, DEF, along the ordering $d = (A, B, C, D, E, F)$.

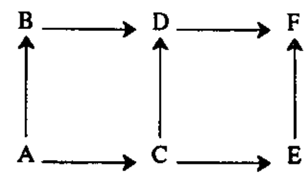


Figure 1: The characteristic dag of a CS

Lemma 1: If D is the characteristic dag of the pair (d, CS) then it is also the characteristic dag of (d', CS) whenever d' respects D and, furthermore, if $<d, CS>$ is a causal theory, then so is $<d', CS>$. □

We now define causal theories and models using dags:

Definition 5: A pair $<D, CS>$ is a causal theory if $<d, CS>$ is a causal theory for all d respecting D.

Definition 6 (Causal model): Given a relation ρ and an arbitrary dag D, D is a **causal model** of ρ if there exists a constraint specification CS such that $<D, CS>$ is causal theory of ρ.

It is easy to see that not every dag D could be a causal model of a given relation ρ. For example, the relation defined by the pair of logical clauses $\{X \vee Z, Y \vee Z\}$ can be modeled by either $X \rightarrow Z \leftarrow Y$ or $X \leftarrow Z \rightarrow Y$, but not by $X \rightarrow Y \leftarrow Z$. The reason is that while the former two dags form causal theories with the specification above, no such theory can be formed for the third dag, because to determine the permissible values of X and Z we must consult a later variable, Y.

To determine whether a dag D is a causal model of a given relation ρ, one need not enumerate the space of specifications for ρ. The condition is simply that D should decompose ρ by the following rule: Fixing its parents in D, each variable must remain unaffected by all other variables, except possibly by its descendants in D. This rule reflects another common feature of causation: once we learn the current status of its direct causal factors, no other information is needed for predicting the state of a given variable.

Formally, D is a causal model of ρ if for some ordering $X_1, \ldots, X_n$ respecting D and for all i, we have $\Pi_{X_1, \ldots, X_{i-1}}(\rho) \bowtie \Pi_{P_i \cup \{X_i\}}(\rho) = \Pi_{X_1, \ldots, X_i}(\rho)$ where P_i stands for the parents of X_i. This result follows from the theory of graphoids, as applied to database dependencies [Pearl, 1988]. The complexity of the test above is polynomial in the size of ρ, but may be exponential in the number of variables. Once D qualified as a causal model of ρ, a causal theory $<D, CS>$ (of ρ) can be formed by simply pairing D with the projections of ρ on the families of D.

3. Synthesizing Causal Theories and Uncovering Causal Directionality

Our ultimate goal is to construct causal theories for the information we possess. In this section we analyze two tasks. First, we assume that the information we have is a database tabulating explicitly the tuples of some relation ρ, and our task is to replace the table by a more economical representation, one that enjoys the computational advantage of causal organizations. Such a task would be useful in machine learning applications, where the tuples represent a stream of observations and the causal theory forms a convenient model of the environment, facilitating modular organization and fast predictions. In our second task, the information will be given in the form of a preformulated constraint specification CS, and the problem will be to construct a causal theory without explicating the underlying relation of CS.

Task 1: (decomposition) Given a relation ρ and an ordering d, find a causal theory for ρ along d.

Barring additional requirements, a causal theory can be obtained by a trivial construction. For instance, the complete dag generated by directing an edge from each lower variable to every higher variable is clearly a causal model of ρ, and the desired causal theory can be obtained by projecting ρ onto the complete families $F_i = \{X_1, X_2, \ldots, X_i\}$. We next present a scheme for constructing a causal theory on top of an **edge-minimal** model of ρ, that is, a dag D from which no edge can be deleted without destroying its capability to support a causal theory of ρ.

The algorithm that follows constructs an edge-minimal causal model of ρ.

build-causal-1 (ρ, d):
1. Begin
2. For $i = n$ to 2 by -1 do:
3. Find a minimal subset $P_i \subseteq \{X_1, \ldots, X_{i-1}\}$ such that $\Pi_{X_1, \ldots, X_{i-1}}(\rho) \bowtie \Pi_{P_i \cup \{X_i\}}(\rho) = \Pi_{X_1, \ldots, X_i}(\rho)$
4. Return a dag D generated by directing an arc from each node in P_i towards X_i.
5. End.

To form a causal theory, we simply pair this dag with the projections of ρ on its families.

The construction above shows that a causal theory can be found for any arbitrary ordering. However, we will next show that certain orderings possess features that render them more natural for a given relation. It is these features, we conjecture, which give rise to the perception that certain relations possess "intrinsic" directionalities.

Definition 7 (Model Preference): A causal model D_2 is said to be at least **as expressive** as D_1, denoted $D_1 \leq D_2$, if for any causal theory $<D_1, CS_1>$ there exists a causal theory $<D_2, CS_2>$ such that $rel(CS_1) = rel(CS_2)$. A dag D is said to be a **minimal** causal model of ρ if it is not strictly more expressive than any other causal model of ρ. In other words, the set of relations modeled by D is not a superset of any set of relations, containing ρ, that can be modeled by some other dag.

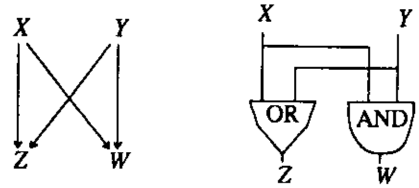
Clearly, every minimal model must be edge-minimal, but not the converse. For example, the complete dag $Z \to X$, $Z \to Y$, $X \to Y$ is an edge-minimal causal model of the relation given by the formula $Z = X \vee Y$, but it is not a minimal model, because it is strictly more expressive than the dag $X \to Z \leftarrow Y$; the latter can model only relations where X does not constrain Y. Polynomial graphical methods for testing preference and equivalence between causal models are described in [Pearl et al., 1990]. However, finding a minimal model for a given relation may be exponentially hard.

Definition 8 (Intrinsic Directionality): Given a relation ρ, a variable X is said to be **a direct cause** of variable Y, if there exists a directed edge from X to Y in all minimal causal models of ρ.

Example 1. Consider a relation ρ specified by the table of Figure 2(a). The table is small enough to verify that the dag in 2(b) is the only minimal causal model of ρ. For example, the arrow from X to Z cannot be reversed, because ρ cannot be expressed as a set of constraints on the families of the resulting dag, $\{YZ, ZX, XYW\}$. Adding an arc $Y \to X$ to the resulting dag would permit a representation of ρ (using the scheme $\{YZ, YZX, YXW\}$), but would no longer be minimal. It is strictly more expressive than the one in 2(b), because, unlike the latter, it also models relations in which some X Y pairs are forbidden. The causal theory corresponding to the dag of 2(b) is shown in 2(c), matching our intuition about the causal relationships embedded in 2(a). Note that the same minimal model ensues (though not the same theory) were we to destroy the functional dependencies by adding the tuple 1100 to the table in 2(a). However, it is no longer unique.

X	Y	Z	W
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

(a)



(b) (c)

Figure 2: The directionality shown in (b) is intrinsic to the relation in (a), because (b) is a unique minimal causal model of (a).

Verma and Pearl [1990] have used minimal model semantics to construct a probabilistic definition of causal directionality. They have also developed a proof theory which, under certain conditions provides efficient algorithms for determining causal directionality without examining the vast space of minimal models [Pearl and Verma, 1991]. Whether similar conditions exist in the relational framework remains an open problem.

Task 2: Given a constraint specification CS, find a dag D and a constraint specification CS', s.t. $<D, CS'>$ is a causal theory of $rel(CS)$.

This task can be solved by executing algorithm *Adaptive-consistency* [Dechter and Pearl, 1987] along an arbitrary ordering d:

build-causal-2 (CS, d)
1. Begin
 2. Execute *adaptive-consistency* w.r.t order d.
 3. Take the graph induced by *adaptive consistency* and direct edges from lower to higher variables, Call this dag D.
 4. Return the set of constraints CS' induced by *adaptive consistency*.
5. End

The resulting pair $<D, CS'>$ is a causal theory of CS. Algorithm *adaptive-consistency* is known to be exponential in the induced width W^* of $scheme(CS)$ [Dechter and Pearl, 1987], hence, it is a practical procedure only for sparse constraint topologies.

4. Finding Causal Ordering

In this section we characterize sufficient conditions under which causal theories can be assembled from pre-existing constraint specifications. Part of these conditions relate to the nature of the individual constraints; each must permit a causal relationship to exist between variables designated as *inputs* and that designated as the *output*. Additionally, to avoid conflicts in assembling the overall theory, we shall insist that no two constraints designate the same variable as their output. Whether a given CS can comply with the latter restriction depends only on the topological property of $scheme(CS)$ and is captured in the notion of an **ordered** CS.

Definition 9: An ordered constraint specification (OCS) is a pair (D, CS) consisting of a dag D, and a constraint specification CS in which every component is a family of D.

For example, a CS with constraints defined on $\{AB, AC, BCD, CE, DEF\}$ forms an OCS with the dag of Figure 1. However, if instead of BCD, we had two separate constraints, on BD and CD, this dag could no longer be paired with the CS to form an OCS and, in fact, no such dag exists.

A *Scheme* that permits the formation of an OCS is called a WEB and can be identified in quadratic time [Dalkey, 1991]. The following procedure[1] for identifying OCS's parallels the notion of WEB's unfolding in [Dalkey, 1991]:

Procedure build-dag-3 (CS)
1. While $scheme(CS)$ is non-empty do
 2. If there is a component $S \in scheme(CS)$ containing a **lonely** variable X (i.e., one that participates in only one constraint), then
 3. direct edges from all variables in $S - X$ towards X, and remove S from $scheme(CS)$.
 4. else, return failure.
5. end while
6. return the dag D generated.

Theorem 2: Procedure **build-dag-3** (CS) returns a dag D iff (D, CS) is an OCS, else, no such dag exists. □

Task 3: Given a CS, find, whenever possible, a dag D s.t. $<D, CS>$ is a causal theory of $rel(CS)$.

In general, this task may require insurmountable amount of computations. The task becomes easier when the CS can be paired with a dag D to form an OCS by the procedure above. Still, not every OCS pair (D, CS) corresponds to a causal theory $<D, CS>$ according to criterion (1). We next show that if the constraints residing in a given OCS meet certain conditions, then the OCS always yields a causal theory. Such constraints will be called **causal**.

Definition 10 (Causal constraints): A constraint, C, on a set of variables $U = \{X_1, \ldots, X_{k-1}, X_k\}$ is said to be causal with respect to a subset O of its variables if the following two conditions are met:

i). Any assignment of values to $U - O$ is legal. Formally, if $O = \{X_{i_1}, \ldots, X_{i_{|O|}}\} \subset U$, then

[1] A similar procedure was used to order variables in ThinkLab [Borning, 1981].

$\Pi_{U-O}(C) = dom_{i_1} \times, \ldots, \times dom_{i_{|U-O|}}$.

ii). Let O_j denote the set $\{X_{i_j}\} \cup U-O$, then $C = \Pi_{O_1}(C) \bowtie, \ldots, \bowtie \Pi_{O_{|O|}}(C)$. In other words, C can be **losslessly**[1] decomposed into $|O|$ smaller constraints, each defined on $U-O$ plus a single variable from O.

We say that $U-O$ and O are **input** and **output** sets, respectively. Note that a subset of an output set O, may or may not qualify as an output.

Example 2: Consider the constraint C specified in Example 1. The sets $\{X, Y\}$, $\{X\}$, $\{Y\}$, $\{Z\}$, and $\{W\}$ qualify as input sets ($U-O$) according to condition (i), since C permits all possible assignments to their constituents. However, only $\{X, Y\}$ qualifies as an input set by requirement (ii), since C can be losslessly decomposed only into the scheme $\{XYZ, XYW\}$. Hence, C is *causal* w.r.t. $O = \{Z, W\}$, and indeed, it matches the functional description of C as shown in Figure 2(c).

Theorem 3: Given an OCS (D, CS) in which all constraints are causal. If for every component S, the sons of all families contained in S are an output set, then $<D, CS>$ is a causal theory.[2] □

Thus, an unordered set of *causal* constraints can be assembled into a causal theory if: 1. There is a dag D that will render it an OCS and, 2. For each constraint C, the sons of all the families contained in C must coincide with an output set of C.

We call a dag satisfying conditions 1 and 2 a **legal dag**.

Task 4: Given a CS in which all constraints are causal, find, whenever possible, a legal dag D.

It turns out that deciding whether a legal dag exists does not require enumerating all dags that might be returned by procedure build-dag-3, a simple modification of steps 2 and 3 suffices. Procedure **legal-dag-4** is identical to build-dag-3 when only step 2 is modified.

Procedure legal-dag-4 (step 2-3)

2: If there is a component $S \in$ *scheme*(CS) for which an output set O contains only lonely variables, then, for each $X \in O$,

3: directed edges from all variables in $S-O$ towards X, and remove S from *scheme*(CS).

Theorem 4: Given a CS, procedure **legal-dag-4** returns a legal dag D iff such a dag exists. □

An interesting and rather common type of causal constraints are those in which every singleton variable can serve as an output.

Definition 11 (Symmetric constraints): A causal constraint is said to be **symmetric** if it is causal with respect to each of its singleton variables.

For example, the constraint on $\{X, Y, Z\}$ given by the linear inequality $X + Y + Z \le a$, is causal and symmetric, since any one of the three variables qualifies as an output, with the other two as inputs. It is easy to verify that linear equalities, e.g. $X + Y + Z = a$ and propositional clauses, e.g. $X \vee Y \vee Z$ are also symmetric. Clearly, when all constraints are symmetric any CS that forms an OCS has a legal dag.

Corollary: A set of propositional clauses that forms an OCS is always satisfiable and, moreover, a satisfying assignment can be found in linear time. □

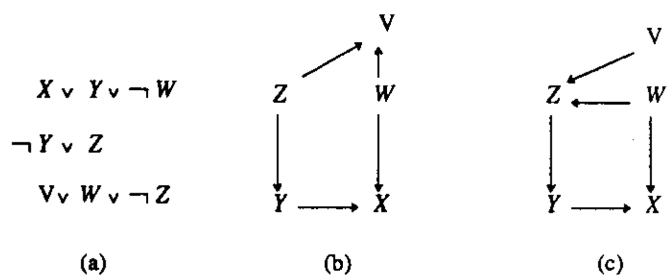


Figure 3: A constraint specification (a) with two of it legal dags (b) and (c).

Example 3: Consider a constraint specification CS given by the three clauses of Figure 3 (a). Letting **build-dag-3** select lonely variables in the orders (V, X, Y) yields the dag of Figure 3 (b). This, as well as any other dag constructed by **build-dag-3**, forms a causal theory with CS, since all clauses are symmetrical causal constraints. Now assume we change the last clause to $Z = V \vee W$ which is still causal, but no longer symmetric; Z is the only variable that qualifies as output. Running **legal-dag-4** on the new CS, we see that the only lonely variable that can be selected initially is X, because V does not qualify as an output. After removing the constraint $X \vee Y \vee \neg W$, the next lonely variable must be Y and then Z. This order yields the legal dag of Fig.3 (c).

A subclass of OCS's that is causal w.r.t. any specification of the constraints is the well known **acyclic** CS [Dechter and Pearl, 1989] which is closely related to acyclic databases [Beeri, 1983]. It can be shown that enforcing local consistency between adjacent constraints is sufficient for rendering any acyclic CS backtrack-free, hence, a causal theory.

[1]A relation ρ is said to be *losslessly decomposed* into $\rho_1, \ldots, \rho_t$ if $\rho = \rho_1 \bowtie \rho_2 \bowtie, \ldots, \bowtie \rho_t$.

[2]Strictly speaking, the causal theory is $<D, CS'>$ where CS' is a decomposition of CS according to step (ii) of Definition 10.

In case algorithm **legal-dag-4**, fails, we know that the specifications do not lend themselves to causal modeling by straightforward variable ordering. It is still feasible though that causal theories could be formed by treating clusters of variables as single objects. For example, the ordering scheme of [Iwasaki and Simon, 1986] illustrates how such clusters can be organized systematically (and uniquely) whenever the constraints are linear equalities having a unique solution. The basic **build-dag-3** algorithm can be used to identify promising candidates of variable clusters, and to assemble more general causal theories than those treated in this paper.

5. Conclusions

This paper presents a relational semantics for the directionality associated with cause-effect relationships, explaining why prediction is easy while diagnosis and planning are hard. We used this semantics to show that certain relations possess intrinsic directionalities, similar to those characterizing causal influences. We also provided an effective procedure for deciding when and how an unstructured set of constraints can be configured so as to form a directed causal theory.

These results have several applications. First, it is often more natural for a person to express causal relationships as directional, rather than symmetrical constraints. The semantics presented in this paper permits us to interpret and process directional relationships in a consistent way and to utilize the computational advantages latent in causal theories. Second, the notion of intrinsic directionality suggests automated procedures for discovering causal structures in raw observations or, at the very least, for organizing such observations into structures that enjoy the characteristics of causal theories. Finally, the set of constraint specifications that can be configured to form causal theories constitutes another "island of tractability" in constraint satisfaction problems. The procedure provided for identifying such specifications can be used to order computational sequences in qualitative physics and scheduling applications.

References

[Beeri et al., 1983] Beeri, C., Fagin, R., Maier, D., and Yannakakis, M., "On the desirability of acyclic database schemes," *Journal of ACM*, Vol. 30, No. 2, July, 1983, pp. 479-513

[Borning, 1981] Borning, Alan H., "The programming language aspects of ThinkLab: A constraint-oriented simulation laboratory." *ACM Transactions on Programming Languages and Systems*, Vol. 3, No. 4, October, 1981, pp. 353-387.

[Dalkey, 1991]. Dalkey, N.C., "Modeling probability distributions with WEB structures." Cognitive Systems Laboratory, University of California, Los Angeles. Technical Report R-164, 1991.

[de-Kleer and Brown, 1986] de-Kleer, J. and Brown, J.S., "Theories of causal ordering," *Artificial Intelligence*, Vol. 29, No. 1, 1986, pp. 33-62.

[Dechter, 1990] Dechter, R., "From local to global consistency," In *Proceedings, Eighth Canadian Conference on AI, CSCSI-90*, May 23-25, 1990, pp. 231-237.

[Dechter and Pearl, 1987] Dechter, R. and Pearl, J., "Network-based heuristics for constraint-satisfaction problems," *Artificial Intelligence*, Vol. 34, No. 1, December, 1987, pp. 1-38.

[Dechter and Pearl, 1989] Dechter, R. and Pearl, J., "Tree clustering for constraint networks," in *Artificial Intelligence*, 1989, pp. 353-366.

[Dechter and Pearl, 1991] Dechter, R. and Pearl, J., "Directed constraint networks," Cognitive Systems Laboratory, University of California, Los Angeles. Technical Report R-153-L, 1991.

[Freuder, 1982] Freuder, E.C., "A sufficient condition for backtrack-free search." *JACM*, 29(1), 1982, pp. 24-32.

[Iwasaki and Simon, 1986] Iwasaki, Y. and Simon, H.A., "Causality in device behavior," *Artificial Intelligence*, Vol. 29, No. 1, 1986, pp. 3-32.

[Kuipers, 1984] Kuipers, B. "Common sense reasoning about causality: Deriving behavior from structure," *Artificial Intelligence*, Vol. 24, No. 1-3, 1984, pp 169-203.

[Pearl, 1988] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Palo Alto, CA., Morgan Kaufmann, 1988.

[Pearl et al., 1990] Pearl, J., Geiger, D., and Verma, T., "The logic of influence diagrams," in R.M. Oliver and J.Q. Smith (Eds), *Influence Diagrams, Belief Nets and Decision Analysis*, Sussex, England: John Wiley & Sons, Ltd., 1989, pp. 67-87.

[Pearl and Verma, 1991] Pearl, J. and Verma T., "A theory of inferred causation," in Allen, J.A., Fikes, R., and Sandewall, E. (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*. San Mateo, CA: Morgan Kaufmann, (April, 1991), pp. 441-452.

[Shoham, 1988] Shoham, Y., *Reasoning About Change*, Cambridge, Massachusetts: The MIT Press, 1988.

[Verma and Pearl, 1990] Verma, T. and Pearl, J., "Equivalence and synthesis of causal models," in *Proceedings, Sixth Conference on Uncertainty in AI*, Cambridge, Massachusetts, July 27-29, 1990, pp. 220-227.