Causal Graph Justifications of Logic Programs∗

Pedro Cabalar, Jorge Fandinno

Department of Computer Science
University of Corunna, Spain
(*e-mail:* `{cabalar, jorge.fandino}@udc.es`)

Michael Fink

Vienna University of Technology,
Institute for Information Systems
Vienna, Austria
(*e-mail:* `fink@kr.tuwien.ac.at`)

Abstract

In this work we propose a multi-valued extension of logic programs under the stable models semantics where each true atom in a model is associated with a set of justifications. These justifications are expressed in terms of *causal graphs* formed by rule labels and edges that represent their application ordering. For positive programs, we show that the causal justifications obtained for a given atom have a direct correspondence to (relevant) syntactic proofs of that atom using the program rules involved in the graphs. The most interesting contribution is that this causal information is obtained in a purely semantic way, by algebraic operations (product, sum and application) on a lattice of causal values whose ordering relation expresses when a justification is stronger than another. Finally, for programs with negation, we define the concept of *causal stable model* by introducing an analogous transformation to Gelfond and Lifschitz's program reduct. As a result, default negation behaves as "absence of proof" and no justification is derived from negative literals, something that turns out convenient for elaboration tolerance, as we explain with a running example.

KEYWORDS: Answer Set Programming, Causality, Knowledge Representation, Multi-valued Logic Programming

1 Introduction

An important difference between classical models and most Logic Programming (LP) semantics is that, in the latter, true atoms must be founded or justified by a given derivation. Consequently, falsity is understood as absence of proof: for instance, a common informal way reading for default literal *not p* is "there is no way to derive *p*." Although this idea seems quite intuitive, it actually resorts to a concept, the *ways to derive p*, outside the scope of the standard LP semantics. In other words, LP semantics point out whether there exists some derivation for an atom, but do not provide the derivations themselves, if several alternatives exist.

However, such information on justifications for atoms can be of great interest for Knowledge Representation (KR), and especially, for dealing with problems related to causality. In the area of diagnosis, for instance, when a discrepancy between expected and observed behaviour

is found, it may be convenient to not only exhibit a set of malfunctioning components as explanation, but also the way (a causal graph) in which these breakdowns have eventually caused the discrepancies. Another potential application area is legal reasoning where determining a legal responsability usually involves finding out which agent or agents have eventually caused a given result, regardless the chain of effects involved in the process. An important challenge in causal reasoning is the capability of not only deriving facts of the form "*A* has caused *B*," but also being able to represent them and reason about them. As an example, take the assertion:

"If somebody causes an accident, (s)he is legally responsible for that."

This law does not specify the possible ways in which a person may cause an accident. Depending on a representation of the domain, the chain of events from the agent's action(s) to the final effect may be simple (a direct effect) or involve a complex set of indirect effects and defaults like inertia. Regarding representation of the above law, for instance, one might think of an informal rule:

$$responsible(X,Y) \leftarrow action(A),\ person(X),\ accident(Y),\ ``do(A,X)\ \texttt{caused}\ occurs(Y)"\ .$$

If the pseudo-literal "$do(A,X)\ \texttt{caused}\ occurs(Y)$" actually corresponds to an explicit representation of all the possible ways of causing an accident, however, one immediately runs into a problem of *elaboration tolerance* (McCarthy 1998) — adding new rules that causally connect $do(A,X)$ to $occurs(Y)$ (in a direct or indirect way) would force us to build new rules for $responsible(X,Y)$. What is needed instead, and what we actually propose as an eventual aim and future extension of our work, is to introduce, indeed, some kind of new LP literal "*A* caused *B*," with *an associated semantics* capable of revealing causes *A* of a given true atom *B*.

While not straightforward, the rewarding perspective of such a semantic approach is an extension of Answer Set Programming (ASP) (Brewka et al. 2011) with causal literals capable of representing different kinds of causal influences (sufficient cause, necessary cause, etc). In this paper, we tackle the above issue and, as a first step and basic underlying requirement, develop a suitable semantics capable of associating causal justifications with each true atom. To this end, we propose a multi-valued extension of logic programs under the stable model semantics (Gelfond and Lifschitz 1988) where each true atom in a model is associated with a set of justifications in the form of *causal graphs*. To further illustrate our motivation, consider the following example.

Example 1 (*From Cabalar 2011*)
Some country has a law *l* that asserts that driving drunk is punishable with imprisonment. On the other hand, a second law *m* specifies that resisting arrest has the same effect. The execution *e* of a sentence establishes that a punishment implies imprisonment. Suppose that some person drove drunk and resisted to be arrested.   □

We can capture this scenario with the following logic program P_1:

$l:$ $punish \leftarrow drive,\ drunk$	$m:$ $punish \leftarrow resist$	$e:$ $prison \leftarrow punish$
$d:$ $drive$	$k:$ $drunk$	$r:$ $resist$

The least model of this positive program makes atom *prison* true, so we know that there exists a possible derivation for it. In particular, two alternative justifications can be made, corresponding to the graphs in Figure 1(a): driving drunk and, independently, resisting to authority (vertices and edges respectively corresponds with rule labels and their dependences).

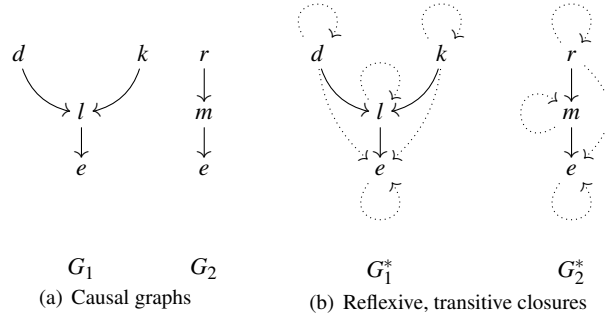More specifically, we summarise our contributions as follows.

(a) Causal graphs   (b) Reflexive, transitive closures

Fig. 1. Derivations G_1 and G_2 justifying atom *prison* in program P_1.

- We define a multi-valued semantics for (normal) logic programs based on causal graphs. An important result is that, despite of this semantic nature, we are able to show that causal values have a direct correspondence to (relevant) syntactic proofs using the program rules involved in the graphs (cf. Section 4).
- We also define an ordering relation that specifies when a cause is *stronger* than another, and show how causal values form a lattice with three associated algebraic operations: a product '$*$' representing conjunction or joint causation; a sum '$+$' representing alternative causes; and a non-commutative product '$\cdot$' that stands for rule application. We study beneficial properties of these operations that allow manipulating and reasoning with causal values in an analytical way (cf. Sections 2 and 3).

Fostered by its algebraic treatment of causal values, our work facilitates the incorporation of dedicated, more specific causal expressions representing causal influence of different kinds.

2 Causes as graphs

In this and subsequent Section 3, we introduce the lattice of causal values in two different steps. In a first step, we focus on the idea of an "individual" cause and then we proceed to explain the concept of causal value that allows collecting different alternative causes.

We begin recalling several graph definitions and notation. A (directed) *graph* is a pair $\langle V, E \rangle$ where V is a set of vertices V and E is a set of edges $E \subseteq V \times V$. In the following definitions, let $G = \langle V, E \rangle$ and $G' = \langle V', E' \rangle$ be two graphs. We say that G is a *subgraph* of G', written $G \subseteq G'$, when $V \subseteq V'$ and $E \subseteq E'$. We write $G \cup G'$ to stand for the graph $\langle V \cup V', E \cup E' \rangle$. We represent the reflexive and transitive closure of G as G^*. Finally, we introduce a *concatenation* operation $G \odot G'$ on graphs corresponding to a graph with vertices $V \cup V'$ and edges $E \cup E' \cup \{(x, y) \mid x \in V, y \in V'\}$. Notice that, $G \cup G' \subseteq G \odot G'$, that is, the concatenation extends the union of graphs by adding all possible arcs that go from some node in G to some node in G'.

Definition 1 (*Causal graph*)
Given some set *Lb* of (rule) labels, a *causal graph* (*c-graph*) G is a reflexively and transitively closed directed graph, i.e., $G^* = G$, whose vertices are labels, i.e. $V \subseteq Lb$. We denote by $\mathbf{C}_{Lb}$ the set of all possible causal graphs over *Lb*. □

Intuitively, the vertices correspond to rules involved in a derivation of a given atom (or formula), and the edges point out a (partial) ordering of application of rules in the derivation. Figure 1(a) shows two causal graphs with labels from P_1. Transitivity is crucial for interpreting the subgraph relation $G \subseteq G'$ as a way to express that G' is redundant with respect to G. For instance, a graph

G_3 formed by the single edge (r,e) *is not a subgraph* of G_2 but is simpler (does not require using *m*). This fact is captured by $G_3^* \subseteq G_2^*$. Reflexivity is convenient for simpler definitions. For instance, the causal graph formed by a single label *l* also has a single edge (l,l)—we call this an *atomic* causal graphs and represent it just by its label. For simplicity, we will usually omit transitive and reflexive arcs when depicting a causal graph. For instance, taking G_1 and G_2 in Figure 1(a) as causal graphs actually amounts to considering the graphs shown in Figure 1(b), where previously omitted arcs are shown as dotted lines. We define next a natural ordering relation among them.

Definition 2 (*Sufficient*)
A causal graph *G* is *sufficient* for another causal graph G', written $G \leq G'$, when $G \supseteq G'$. □

Saying that *G* is sufficient for G' intuitively means that *G* contains enough information to yield the same effect than G', but perhaps more than needed (this explains $G \supseteq G'$). For this reason, we sometimes read $G \leq G'$ as "G' is *stronger* than *G*."

Since graphs with the subgraph relation form a poset, the set of causal graphs also constitutes a poset $\langle \mathbf{C}_{Lb}, \leq \rangle$ with a top element corresponding to the empty c-graph $G_\emptyset = \langle \emptyset, \emptyset \rangle$. This stands for a kind of "absolute truth" and is of interest for including rules or facts one does not want to label, that is, their effect will not be traced in the justifications.

Any causal graph can be built up from labels (atomic causal graphs) using two basic operations: the product $G * G' \overset{\text{def}}{=} (G \cup G')^*$ that stands for union of causal graphs or *joint interaction*, and the concatenation $G \cdot G' \overset{\text{def}}{=} (G \odot G')^*$ that captures their *sequential application*. The reason for applying a closure is that the result of $G \cup G'$ and $G \odot G'$ does not need to be closed under transitivity. We can extend the product to any (possibly empty and possibly infinite) set of causal graphs *S* so that $\prod S \overset{\text{def}}{=} \left(\bigcup_{G \in S} G \right)^*$.

Example 2 (*Ex. 1 continued*)
The cause for the body of *l* in P_1 is the product of causes for *drive* and *drunk*, that is $d * k$ formed with vertices $\{d,k\}$ and edges $\{(d,d),(k,k)\}$. As a result, the explanation of the rule head, *punish*, is formed by the concatenation of its body cause $d * k$ with its label, that is $(d * k) \cdot l$. In its turn, this becomes the cause for the body of *e* and so, we get the explanation $(d * k) \cdot l \cdot e$ for atom *prison* represented as G_1 in Figure 1(b). Similarly, G_2 corresponds to $r \cdot m \cdot e$. □

When writing these causal expressions, we assume that '·' has higher priority than '∗'. Furthermore, we will usually omit '·' when applied to consecutive labels, so that $r \cdot m \cdot e$ will be abbreviated as *rme*. It is easy to see that $G * G' = G' * G$ while, in general, $G \cdot G' \neq G' \cdot G$, that is, concatenation is not commutative. Another observation is that $G \cdot G' \leq G * G'$, that is, concatenation is sufficient for the product, but the opposite does not hold in general. Moreover, in our running example, we can check that $(d * k) \cdot l$ is equal to $(d \cdot l) * (k \cdot l)$. In fact, application distributes over products and, as a result, we can identify any causal graph with the product of all its edges. To conclude this section, we note that the set of causal graphs $\mathbf{C}_{Lb}$ ordered by $\leq$ forms a lower semilattice $\langle \mathbf{C}_{Lb}, * \rangle$, where the product constitutes the infimum.

3 Alternative causes

Having settled the case for individual causes, let us now proceed to represent situations in which several alternative and independent causes can be found for an atom *p*. The obvious possibility is just using a *set of causal graphs* for that purpose. However, we should additionally disregard

causes for which a stronger alternative exists. For instance, as we saw before, cause *rme* is sufficient for explaining *punish* and therefore, it is also an alternative way to prove this atom, but redundant in the presence of the stronger cause *rm*. This suggests to choose sets of $\leq$-maximal causal graphs as 'truth values' for our multi-valued semantics. In principle, this is the central idea, although $\leq$-maximal causal graphs incur some minor inconveniences in mathematical treatment. For instance, if we collect different alternative causes by just using the union of sets of maximal causal graphs, the elements in the result need not be maximal. Besides, the operations of product and concatenation are expected to extend to the sets adopted as causal values. To address these issues, a more solid representation is obtained resorting to *ideals* of causal graphs.

Given any poset $\langle A, \leq \rangle$, an *ideal I* is any set $I \subseteq A$ satisfying[1]: if $x \in I$ and $y \leq x$ then $y \in I$. A compact way of representing an ideal I is by using its set of maximal elements S, since the rest of I contains *all elements below* them. The *principal ideal* of an individual element $x \in A$ is denoted as $\downarrow x \stackrel{\text{def}}{=} \{y \in A \mid y \leq x\}$. We extend this notion for any set of elements S so that $\downarrow S \stackrel{\text{def}}{=} \bigcup \{\downarrow x \mid x \in S\} = \{y \in A \mid y \leq x, \text{for some } x \in S\}$. Thus, we will usually represent an ideal I as $\downarrow S$ where S are the maximal [2] elements in I. In fact, maximal elements constitute the relevant information provided by the ideal, while keeping all other elements is convenient for simplicity of algebraic treament (but we do not assign a particular meaning to them).

Definition 3 (*Causal Value*)
Given a set of labels *Lb*, a *causal value* is any ideal for the poset $\langle \mathbf{C}_{Lb}, \leq \rangle$. We denote by $\mathbf{V}_{Lb}$ the set of causal values. □

Product, concatenation and the $\leq$-relation are easily extended to any pair U, U' of causal values respectively as: $U * U' \stackrel{\text{def}}{=} U \cap U'$ and $U \cdot U' \stackrel{\text{def}}{=} \downarrow \{ G \cdot G' \mid G \in U \text{ and } G' \in U' \}$ and $U \leq U'$ iff $U \subseteq U'$. We also define addition as: $U + U' \stackrel{\text{def}}{=} U \cup U'$ allowing to collect alternative causes. Using terminology and results from lattice theory in (Stumme 1997) we can prove the following.

Theorem 1
$\mathbf{V}_{Lb}$ forms a free, completely distributive lattice with join $+$ and meet $*$ generated by the lower semilattice $\langle \mathbf{C}_{Lb}, * \rangle$ with the injective homomorphism (or embedding) $\downarrow \colon \mathbf{C}_{Lb} \longrightarrow \mathbf{V}_{Lb}$. □

Essentially, this theorem means that the mapping $\downarrow$ from c-graph G to its principal ideal $\downarrow G$ is preserved for their respective products, $\downarrow (G_1 * G_2) = \downarrow G_1 * \downarrow G_2$, and ordering relations: $G_1 \leq G_2$ (among c-graphs) iff $\downarrow G_1 \leq \downarrow G_2$ (among causal values).

Example 3 (*Ex. 1 continued*)
The interpretation for *punish* has two alternative causes $(d * k) \cdot l$ and *rm* that become the causal values $\downarrow (d * k) \cdot l$ and $\downarrow rm$. The causal value for *punish* is then formed by their addition:
$$\downarrow (d * k) \cdot l \;+\; \downarrow rm \;=\; \downarrow (d * k) \cdot l \;\cup\; \downarrow rm \;=\; \downarrow \{ (d * k) \cdot l, \; rm \}$$
This ideal contains, among others, the cause *rme*, although it is not maximal due to *rm*:
$$\downarrow \{ (d * k) \cdot l, \; rm \} \;\cup\; \downarrow rme = \downarrow \{ (d * k) \cdot l, \; rm, \; rme \} \;=\; \downarrow \{ (d * k) \cdot l, \; rm \} \qquad \square$$

The term completely distributive lattice in Theorem 1 means that meet (resp. join) operation is defined for any infinite subset of $\mathbf{V}_{Lb}$ and distributes over infinite joins (resp. meets). There is also a bottom element, the empty ideal $\emptyset$ (standing for "falsity") that will be denoted as 0, and a top

---

[1] We use terminology from (Stumme 1997). In some texts this is known as *semi-ideal* or *down-set* to differentiate this definition from the stronger case in which ideals are applied on a (full) lattice rather than a semi-lattice.

[2] Note that, in the case of causal graphs, the existence of maximal elements for the $\leq$-relation amounts to the existence of minimal elements for the subgraph relation, and this holds since the latter is well-founded.

element, the ideal formed by the empty causal graph $\downarrow G_\emptyset = \mathbf{C}_{Lb}$ (standing for "absolute truth") that is denoted as 1 from now on. To improve readability, we introduce the syntactic notion of *causal terms*, that allow representing the possible causal values without explicitly resorting to graphs or ideals.

Definition 4 (*Causal term*)
A *(causal) term*, t, over a set of labels Lb, is recursively defined as one of the following expressions $t ::= l \mid \prod S \mid \sum S \mid t_1 \cdot t_2$ where $l \in Lb$, t_1, t_2 are in their turn causal terms and S is a (possibly empty and possible infinite) set of causal terms. When S is finite and non-empty, $S = \{t_1, \ldots, t_n\}$ we write $\prod S$ simply as $t_1 * \cdots * t_n$ and $\sum S$ as $t_1 + \cdots + t_n$. □

We assume that '$*$' has higher priority than '$+$'. The causal value associated to a causal term is naturally obtained by the recursive application of its operators until we reach the level of labels, so that each label l in the term actually stands for the principal ideal $\downarrow l$. When $S = \emptyset$, the union in S corresponds to the bottom causal value that is, $\sum \emptyset = 0$. Analogously, the intersection of elements in $S = \emptyset$ corresponds to top causal value, i.e., $\prod \emptyset = 1$.

From now on, we will use causal terms as compact representations of causal values. Individual causes (i.e. causal graphs) correspond to terms without addition (note that this also excludes 0, the empty sum). Several interesting algebraic properties can be proved for causal values. In particular, Theorem 1 guarantees that they form a free completely distributive lattice with respect to '$*$' and '$+$' satisfying the standard properties such as associativity, commutativity, idempotence, absorption or distributivity on both directions[3]. Besides, as usual, 0 (resp. 1) is the annihilator for '$*$' (resp. '$+$') and the identity for '$+$' (resp. '$*$'). More significantly, the main properties for '$\cdot$' are shown in Figure 2.

Associativity		Absorption			Identity			Annihilator		
$t \cdot (u \cdot w)$	$=$	$(t \cdot u) \cdot w$	t	$=$	$t + u \cdot t \cdot w$	t	$=$	$1 \cdot t$	0	$=$	$t \cdot 0$
			$u \cdot t \cdot w$	$=$	$t * u \cdot t \cdot w$	t	$=$	$t \cdot 1$	0	$=$	$0 \cdot t$

Indempotence	Addition distributivity			Product distributivity			
$l \cdot l = l$	$t \cdot (u + w)$	$=$	$(t \cdot u) + (t \cdot w)$	$c \cdot d \cdot e$	$=$	$(c \cdot d) * (d \cdot e)$	with $d \neq 1$
	$(t + u) \cdot w$	$=$	$(t \cdot w) + (u \cdot w)$	$c \cdot (d * e)$	$=$	$(c \cdot d) * (c \cdot e)$	
				$(c * d) \cdot e$	$=$	$(c \cdot e) * (d \cdot e)$	

Fig. 2. Properties of the '$\cdot$' operator (c, d, e are terms without '$+$' and l is a label).

4 Positive programs and minimal models

Let us now reconsider logic programs and provide a semantics based on the causal values we have just defined. For the syntax, we recall standard LP definitions, just slightly extending it by introducing rule labels. A *signature* is a pair $\langle At, Lb \rangle$ of sets that respectively represent a set of *atoms* (or *propositions*) and a set of *labels*. As usual, a *literal* is defined as an atom p (positive literal) or its default negation *not p* (negative literal). In this paper, we will concentrate on programs without disjunction in the head (leaving its treatment for future work).

Definition 5 (*Causal logic program*)
Given a signature $\langle At, Lb \rangle$, a *(causal) logic program* P is a (possible infinite) set of rules of the form:
$$t : H \leftarrow B_1, \ldots, B_n, \tag{1}$$
where $t \in Lb \cup \{1\}$, H is an atom (the *head* of the rule) and $B_1, \ldots, B_n$ are literals (the *body*). □

---

[3] The term "free lattice" in Theorem 1 means that *any* equivalence with $*$ and $+$ can be derived from these properties.

For any rule R of the form (1) we define $label(R) \stackrel{\text{def}}{=} t$. We denote by $head(R) \stackrel{\text{def}}{=} H$ its *head*, and by $body(R) \stackrel{\text{def}}{=} \{B_1, \ldots, B_n\}$ its *body*. When $n = 0$ we say that the rule is a *fact* and omit the symbol '←.' When $t \in Lb$ we say that the rule is labelled; otherwise $t = 1$ and we omit both t and ':'. By these conventions, for instance, an unlabelled fact p is actually an abbreviation of $(1 : p \leftarrow)$. A logic program P is *positive* if it contains no default negation. A program is *uniquely labelled* if no pair of labelled rules share the same label, and *completely labelled* if, additionally, all rules are labelled. For instance, P_1 is completely labelled.

Given a signature $\langle At, Lb \rangle$ a *causal interpretation* is a mapping $I : At \longrightarrow \mathbf{V}_{Lb}$ assigning a causal value to each atom. For any interpretations I, J, we say that $I \leq J$ when $I(p) \leq J(p)$ for each atom $p \in At$. Hence, there is a $\leq$-bottom (resp. $\leq$-top) interpretation $\mathbf{0}$ (resp. $\mathbf{1}$) that stands for the interpretation mapping each atom p to 0 (resp. 1). The value assigned to a negative literal *not p* by an interpretation I, denoted as $I(not\ p)$, is defined as: $I(not\ p) \stackrel{\text{def}}{=} 1$ if $I(p) = 0$; and $I(not\ p) \stackrel{\text{def}}{=} 0$ otherwise. An interpretation is *two-valued* if it maps all atoms to $\{0, 1\}$. Furthermore, for any causal interpretation, its corresponding two-valued interpretation, written I^{cl}, is defined so that for any atom p: $I^{cl}(p) \stackrel{\text{def}}{=} 0$ if $I(p) = 0$; and $I^{cl}(p) \stackrel{\text{def}}{=} 1$ otherwise.

Definition 6 (*Causal model*)
Given a positive causal logic program P, a causal interpretation I is a *causal model*, in symbols $I \models P$, if and only if, for each rule $R \in P$ of the form (1), the following condition holds:

$$\big(I(B_1) * \ldots * I(B_n) \big) \cdot t \leq I(H) \qquad \qquad \square$$

Example 4 (*Ex. 1 continued*)
Take rule l from program P_1 and let I be such that $I(drive) = d$ and $I(drunk) = k$. Then I will be a model of l when $(d * k) \cdot l \leq I(punish)$. In particular, this holds when $I(punish) = (d * k) \cdot l + r \cdot m$ which was the value we expected for that atom. But it would also hold when, for instance, $I(punish) = l + m$ or $I(punish) = 1$. The inequality in Definition 6 is important to accommodate possible additional facts such as $(l : punish)$ or even $(1 : punish)$ in the program. $\square$

The fact that any $I(punish)$ greater than $(d * k) \cdot l + r \cdot m$ also becomes a model clearly points out the need for selecting *minimal* models. In fact, as it is the case for non-causal programs, positive programs have a $\leq$-least model that can be computed by iterating an extension of the well-known *direct consequences operator* (van Emden and Kowalski 1976).

Definition 7 (*Direct consequences*)
Given a positive logic program P over signature $\langle At, Lb \rangle$, the operator of *direct consequences* is a function $T_P : \mathbf{I} \longrightarrow \mathbf{I}$ such that, for any causal interpretation I and any atom $p \in At$:

$$T_P(I)(p) \stackrel{\text{def}}{=} \sum \big\{ \big(I(B_1) * \ldots * I(B_n) \big) \cdot t \mid (t : p \leftarrow B_1, \ldots, B_n) \in P \big\}$$

Theorem 2
Let P be a (possibly infinite) positive logic program with n causal rules. Then, (*i*) $lfp(T_P)$ is the least model of P, and (*ii*) $lfp(T_P) = T_P \uparrow^{\omega} (\mathbf{0}) = T_P \uparrow^{n} (\mathbf{0})$. $\square$

The proof of this theorem relies on an encoding of causal logic programs into *Generalized Annotated Logic Programming* (GAP) (Kifer and Subrahmanian 1992) and applying existing results for that general multi-valued LP framework. Theorem 2 just guarantees that the least fixpoint of T_P is well-behaved, but does not explain the nature of the obtained causal values. We illustrate next that these values have a direct relation to the syntactic idea of *proof* in a positive program.

Definition 8

Given a positive program P, a *proof* $\pi(p)$ of an atom p can be recursively defined as a derivation:

$$\pi(p) \quad \stackrel{\text{def}}{=} \quad \frac{\pi(B_1) \ \ldots \ \pi(B_n)}{p} \ (R),$$

where $R \in P$ is a rule with $head(R) = p$ and $body(R) = \{B_1, \ldots, B_n\}$. When $n = 0$, the derivation antecedent $\pi(B_1) \ \ldots \ \pi(B_n)$ is replaced by $\top$ (corresponding to the empty body). ▫

Each derivation in a proof is a particular application of Modus Ponens where, once the body (conjunction of literals B) of a rule R ($p \leftarrow B$) has been proved, then the head p can be concluded.

$$\frac{\dfrac{\top}{drive}(d) \quad \dfrac{\top}{drunk}(k)}{\dfrac{punish}{prison}(l)}(e) \qquad \frac{\dfrac{\top}{resist}(r)}{\dfrac{punish}{prison}(m)}(e) \qquad \frac{\dfrac{\dfrac{\top}{drive}(d) \quad \dfrac{\top}{drunk}(k)}{\dfrac{punish}{\dfrac{sentence}{punish}(s)}(l)}(n)}{prison}(e)$$

Fig. 3. Some proofs for atom *prison* (the rightmost proof is redundant).

Example 5 (*Ex. 1 continued*)

Program P_1 is positive and, in fact, completely labelled, so we can identify each rule with its label. Atom *prison* can be derived in P_1 using the two proofs on the left in Figure 3. These two proofs have a clear correspondence to causes $(d * k) \cdot le$ and *rme* depicted in Figure 1(b). In fact, the least model I of P_1 assigns causal value $I(punish) = (d * k) \cdot le + rme$. ▫

Let P be a positive, completely labelled program. Given a proof π, we define its graph G_π as follows. For each sub-derivation in π of the form $\pi(p)$ in Definition 8 we include an edge (l_i, m) where m is the label of rule R and l_i is the label of the top-level rule in $\pi(B_i)$, for all $i = 1, \ldots, n$. The vertices in G_π exclusively collect the labels in those edges. We define $graph(\pi) \stackrel{\text{def}}{=} G_\pi^*$. The two left proofs in Figure 3 are then obviously mapped to the causal graphs in Figure 1(b). If Π is a set of proofs, we define $graph(\Pi) \stackrel{\text{def}}{=} \{graph(\pi) \mid \pi \in \Pi\}$.

A proof can be sometimes redundant, in the sense that some of its derivation steps could be removed. A natural way of defining a non-redundant proof is resorting to its associated graph. We say that a proof $\pi(p)$ of an atom p in a positive, completely labelled program P is *redundant* if there exists another proof of p, $\pi'(p)$, such that $graph(\pi(p)) \leq graph(\pi'(p))$, in other words, we can build another proof π' with a smaller associated graph.

Example 6

Suppose that we introduce an atom *sentence* which acts as a synonym for *punish*. Furthermore, assume law m mentions *sentence* as its head now, instead of *punish*. Hence, let P_2 be program:

$$
\begin{aligned}
l &: \quad punish \leftarrow drive, drunk & d &: \quad drive & n &: \quad punish \leftarrow sentence \\
m &: \quad sentence \leftarrow resist & k &: \quad drunk & s &: \quad sentence \leftarrow punish \\
e &: \quad prison \leftarrow punish & r &: \quad resist
\end{aligned}
$$

Then, the rightmost proof shown in Figure 3 together with its associated graph $(d * k) \cdot lsne$ is redundant, since the (still valid) leftmost proof in Figure 3 for *prison* has an associated stronger cause (or smaller graph) $(d * k) \cdot le$. Considering the positive loop formed by n and s, one may wonder why it does not spoil the computation of T_{P_2} to iterate forever (adding more and more

concatenations of n and s). The reason is that, at a given point, subsequent iterations yield redundant graphs subsumed by previous steps. In particular, the iteration of T_{P_2} yields the steps:

i	$drive$	$drunk$	$resist$	$sentence$	$punish$	$prison$
1	d	k	r	0	0	0
2	d	k	r	rm	$(d*k)\cdot l$	0
3	d	k	r	$rm+(d*k)\cdot ls$	$(d*k)\cdot l+rmn$	$(d*k)\cdot le$
4	d	k	r	$rm+(d*k)\cdot ls$	$(d*k)\cdot l+rmn$	$((d*k)\cdot l+rmn)\cdot e$

reaching a fixpoint at step 4. The value for *sentence* at step 4 would actually be the sum of rm (derived from *resist*) with the value of *punish* in the previous step, $(d*k)\cdot l+rmn$ followed by s. This corresponds to:

$$rm+(\underbrace{(d*k)\cdot l+rmn}_{punish})\cdot s = rm+(d*k)\cdot ls+rmns \qquad \text{distributivity}$$

$$= rm+rmns+(d*k)\cdot ls \qquad \text{commutativity}$$

$$= rm+rm\cdot ns+(d*k)\cdot ls \qquad \text{associativity}$$

$$= rm+1\cdot rm\cdot ns+(d*k)\cdot ls \quad \text{identity}$$

$$= rm+(d*k)\cdot ls \qquad \text{absorption for '}+\text{' and '}\cdot\text{'}$$

That is, iterating the loop $rmns$ is redundant since a stronger cause rm was obtained before. $\square$

Theorem 3
Let P be a positive, completely labelled program, and Π_p the set of non-redundant proofs of some atom p with respect to P. If I denotes the least model of P, then:

$$G \in graph(\Pi_p) \quad \text{iff} \quad G \text{ is a maximal causal graph in } I(p) \qquad \square$$

Note the importance of this result: it reveals that the information we obtain by a purely semantic treatment of causal values (computing the least model by algebraic operations) has a one-to-one correspondence to syntactic proofs obtained by modus ponens that are further guaranteed to be non-redundant (they do not contain unnecessary steps). Completely labelled programs are interesting for establishing the correspondence in the theorem above, but there are several scenarios in which one may be interested in disregarding the effect of rules in a program or in identifying a group of rules under the same label.

Example 7
Let P_3 be the following variation of P_2:

$z:$	$sentence \leftarrow drive, drunk$		$d:$	$drive$	$punish \leftarrow sentence$
$z:$	$punish \leftarrow resist$		$k:$	$drunk$	$sentence \leftarrow punish$
$e:$	$prison \leftarrow punish$		$r:$	$resist$	

where l and m in P_2 are now just two cases of a common law z, and *punish* and *sentence* depend on each other through unlabelled rules. $\square$

Removing the labels in the positive cycle between *sentence* and *punish* captures the idea that, since they are synonyms, whenever we have a cause for *sentence*, it immediately becomes a cause for *punish* and vice versa. By iterating the T_P operator, it is not difficult to see that the least causal model I_3 makes the assignments $I_3(sentence) = I_3(punish) = (d*k)\cdot z+rz$ (that is *sentence* and *punish* are equivalent) and $I_3(prison) = (d*k)\cdot ze+rze$. This result could also be computed from the least model I_2 for P_2 by replacing l and m by z and "removing" n and s (that

is, replacing them by 1). This is, in fact, a general property we formalise as follows. Given two causal terms t, u and a label l, we define $t[l \mapsto u]$ as the result of replacing label l in t by term u.

Theorem 4
Let P be a positive causal logic program and P' be the result of replacing a label l in P by some u, where u is any label or 1. Furthermore, let I and I' be the least models of P and P', respectively. Then, $I'(p) = I(p)[l \mapsto u]$ for any atom p. □

In particular, in our example, $I_3(p) = I_2(p)[l \mapsto z][m \mapsto z][n \mapsto 1][s \mapsto 1]$, for any atom p. If we remove all labels in a program, we eventually get a standard, unlabelled program. Obviously, its least model will be two-valued, since removing all labels in causal terms, eventually collapses all of them to $\{0, 1\}$. As a result, we can easily establish the following correspondence.

Theorem 5
Let P be a causal positive logic program and P' its unlabelled version. Furthermore, let I be the least causal model of P and I' the least classical model of P'. Then $I' = I^{cl}$. □

5 Default negation

To introduce default negation, let us consider the following variation of our running example.

Example 8
Assume now that law e is a default and that there may be exceptional cases in which punishment is not effective. In particular, some of such exceptions are a pardon, that the punishment was revoked, or that the person has diplomatic immunity. A possible program P_4 encoding this variant of the scenario is:

$l:$	$punish \leftarrow drive, drunk$		$d:$	$drive$	$abnormal \leftarrow pardon$
$m:$	$punish \leftarrow resist$		$k:$	$drunk$	$abnormal \leftarrow revoke$
$e:$	$prison \leftarrow punish, not\ abnormal$		$r:$	$resist$	$abnormal \leftarrow diplomat$

This program has a unique stable model which still keeps *prison* true, since *no proof for abnormal* could be obtained, i.e. no exception occurred. □

From a causal perspective, saying that the lack of an exception is part of a cause (e.g., for imprisonment) is rather counterintuitive. It is not the case that we go to prison because of not receiving a pardon, not having a punishment revocation, not being a diplomat, or whatever possible exception that might be added in the future[4]. Instead, as nothing violated default e, the justifications for *prison* should be those shown in Figure 1(a). In this way, falsity becomes the *default situation* that is broken when a cause is found[5]. This interpretation carries over to negative literals, so that the presence of *not p* in a rule body does not propagate causal information, but instead is a check for the absence of an exception. To capture this behaviour, we proceed to extend the traditional program reduct (Gelfond and Lifschitz 1988) to causal logic programs.

---

[4] A case of the well-known *qualification problem* (McCarthy 1977), i.e., the impossibility of listing all the possible conditions that prevent an action to cause a given effect. Appendix B (available online) contains a more elaborated example showing how the qualification problem may affect causal explanations when inertia is involved.

[5] The paper (Hitchcock and Knobe 2009) contains an extended discussion with several examples showing how people ordinarily understand causes as deviations from a norm.

Definition 9 (*Program reduct*)
The *reduct* of program P with respect to causal interpretation I, in symbols P^I, is the result of:

1. removing from P all rules R, s.t. $I(B) \neq 0$ for some negative literal $B \in body(R)$;
2. removing all negative literals from the remaining rules of P.   □

An interpretation I is a *causal stable model* of program P iff I is the least causal model of P^I.

Example 9 (*Ex. 8 continued*)
Suppose that we add atoms $(p : pardon)$ and $(d : diplomat)$ to program P_4. The only stable model I of this extended program makes $I(prison) = 0$ and $I(abnormal) = p + d$ as expected.   □

Theorem 6 (*Correspondence to non-causal stable models*)
Let P be a causal logic program and P' its unlabelled version. Then:

1. If I is a causal stable model of P, then I^{cl} is a stable model of P'.
2. If I' is a stable model of P' then there is a unique causal stable model I of P s.t. $I' = I^{cl}$. □

This theorem also shows a possible method for computing causal stable models of a program P. We may first run a standard ASP solver on the unlabelled version of P to obtain a stable model I'. This stable model I' has a corresponding causal stable model I, such that $I' = I^{cl}$ and both interpretations coincide in their assignment of 0's. Therefore, $P^I = P^{I'}$ and we can use the latter to iterate the T_P operator and obtain the least causal model of this reduct, which will mandatorily be a causal stable model due to Theorem 6.

6 Related Work

Cabalar (2011) already introduced the main motivations of our work, but used *ad hoc* operations on proof trees without resorting to algebraic structures. A preliminary version (Cabalar and Fandinno 2013) of the current approach relied on chains of labels but was actually *weaker*, missing basic properties we can derive now from causal graphs.

There exists a vast literature on causal reasoning in Artificial Intelligence. Papers on reasoning about actions and change (Lin 1995; McCain and Turner 1997; Thielscher 1997) have been traditionally focused on using causal inference to solve representational problems (mostly, the frame, ramification and qualification problems) without paying much attention to the derivation of cause-effect relations. Perhaps the most established AI approach for causality is relying on *causal networks* (Pearl 2000; Halpern and Pearl 2005; Halpern 2008). In this approach, it is possible to conclude cause-effect relations like "A has caused B" from the behaviour of structural equations by applying the counterfactual interpretation from Hume (1748): "had A not happened, B would not have happened." As discussed by Hall (2004), the counterfactual-based definition of causation corresponds to recognising some kind of *dependence* relation in the behaviour of a non-causal system description. As opposed to this, Hall considers a different (and incompatible) definition where causes must be connected to their effects via *sequences of causal intermediates*, something that is closer to our explanations in terms of causal graphs.

Apart from the different AI approaches and attitudes towards causality, from the technical point of view, the current approach can be classified as a *labelled deductive system* (Broda et al. 2004). In particular, the work that has had a clearest and most influential relation to the current proposal is the *Logic of Proofs* (**LP**) by Artëmov (2001). We have borrowed from that formalism part of the notation for our causal terms and rule labellings and the fundamental idea of keeping track of justifications by considering rule applications.

Focusing on LP, our work obviously relates to explanations as provided by approaches to debugging in ASP (Gebser et al. 2008; Pontelli et al. 2009; Schulz et al. 2013; Damásio et al. 2013). Pereira et al. (1991) and Denecker and De Schreye (1993) also define different semantics in terms of justifications, but do not provide calculi for them. In these works, explanations usually contain all possible ways to derive an atom or to prevent its derivation, including paths through negation. This differs from a KR orientation where only the cause-effect relations that "break the norm" should be considered relevant. This point of view is also shared, e.g., by the counterfactual-based causal LP approach (Vennekens 2011). Fages (1991) characterised stable models in terms of loop-free justifications expressed as partial order relations among atoms in positive bodies. We conjecture that the causal values obtained in our semantics formally capture Fages' justifications. A more far-fetched resemblance exists to work on the analysis of tabled Prolog computations. There, the goal is to identify potential causes for non-termination of program evaluations, which can be achieved examining so-called *forest logs*, i.e., a log of table operations for a computation. By adding unique labels for rules (with the original intention to disambiguate analysis results, cf. Liang and Kifer (2013), however not as an explicit means for representing knowledge), in principle a forest log implicitly contains the information necessary to read of the causal model of a completely labelled positive causal logic program.

7 Conclusions

In this paper we have provided a multi-valued semantics for normal logic programs whose truth values form a lattice of causal graphs. A causal graph is nothing else but a graph of rule labels that reflects some order of rule applications. In this way, a model assigns to each true atom a value that contains justifications for its derivation from the existing rules. We have further provided three basic operations on the lattice: an addition, that stands for alternative, independent justifications; a product, that represents joint interaction of causes; and a concatenation that reflects rule application. We have shown that, for positive programs, there exists a least model that coincides with the least fixpoint of a direct consequences operator, analogous to van Emden and Kowalski (1976). With this, we are able to prove a direct correspondence between the semantic values we obtain and the syntactic idea of proof. These results have been extrapolated to stable models of programs with default negation, understanding the latter as "absence of cause." Although, for space reasons, we have not dealt with programs with variables, their semantics is obtained from their (possibly infinite) grounding, as usual.

Several topics remain open for future study. An interesting issue is to replace the syntactic definition by a reduct in favour of a logical treatment of default negation, as has been done for (non-causal) stable models and their characterisation in terms of Equilibrium Logic (Pearce 2006). Regarding the representation of causal information, a natural next step would be the consideration of syntactic operators for more specific knowledge like the influence of a particular event or label in a conclusion, expressing necessary or sufficient causes, or even dealing with counterfactuals. Further ongoing work is focused on implementation, complexity assessment, and an extension to disjunctive programs, respectively the introduction of strong negation. Exploring related areas of KR and reasoning, such as, e.g., Paraconsistent Reasoning and Belief Revision, seems promising with respect to extending the range of problems to which our approach may effectively be applied.

References

ARTËMOV, S. N. 2001. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic 7,* 1, 1–36.

BREWKA, G., EITER, T., AND TRUSZCZYNSKI, M. 2011. Answer set programming at a glance. *Commun. ACM 54,* 12, 92–103.

BRODA, K., GABBAY, D., LAMB, L., AND RUSSO., A. 2004. *Compiled Labelled Deductive Systems: A Uniform Presentation of Non-Classical Logics.* Research Studies Press.

CABALAR, P. 2011. Logic programs and causal proofs. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.* AAAI.

CABALAR, P. AND FANDINNO, J. 2013. An algebra of causal chains. In *Proc. of the 6th Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP'13).*

DAMÁSIO, C. V., ANALYTI, A., AND ANTONIOU, G. 2013. Justifications for logic programming. In *Proc. of the 12th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning, (LPNMR'13).* Lecture Notes in Computer Science, vol. 8148. Springer, 530–542.

DENECKER, M. AND DE SCHREYE, D. 1993. Justification semantics: A unifiying framework for the semantics of logic programs. In *Proc. of the Logic Programming and Nonmonotonic Reasoning Workshop.* 365–379.

FAGES, F. 1991. A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. *New Generation Computing 9,* 3-4, 425–443.

GEBSER, M., PÜHRER, J., SCHAUB, T., AND TOMPITS, H. 2008. Meta-programming technique for debugging answer-set programs. In *Proc. of the 23rd Conf. on Artificial Inteligence (AAAI'08).* 448–453.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Logic Programming: Proc. of the Fifth International Conference and Symposium (Volume 2),* R. A. Kowalski and K. A. Bowen, Eds. MIT Press, Cambridge, MA, 1070–1080.

HALL, N. 2004. *Two concepts of causality.* 181–276.

HALPERN, J. Y. 2008. Defaults and normality in causal structures. In *Proc. of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR 2008).* 198–208.

HALPERN, J. Y. AND PEARL, J. 2005. Causes and explanations: A structural-model approach. part I: Causes. *British Journal for Philosophy of Science 56,* 4, 843–887.

HITCHCOCK, C. AND KNOBE, J. 2009. Cause and norm. *Journal of Philosophy 11,* 587–612.

HUME, D. 1748. An enquiry concerning human understanding. Reprinted by Open Court Press, LaSalle, IL, 1958.

KIFER, M. AND SUBRAHMANIAN, V. S. 1992. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming 12.*

LIANG, S. AND KIFER, M. 2013. A practical analysis of non-termination in large logic programs. *TPLP 13,* 4-5, 705–719.

LIN, F. 1995. Embracing causality in specifying the indirect effects of actions. In *Proc. of the Intl. Joint Conf. on Artificial Intelligence (IJCAI),* C. S. Mellish, Ed. Morgan Kaufmann, Montreal, Canada.

LIN, F. AND SOUTCHANSKI, M. 2011. Causal theories of actions revisited. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning.*

MCCAIN, N. AND TURNER, H. 1997. Causal theories of action and change. In *Proc. of the AAAI-97.* 460–465.

MCCAIN, N. C. 1997. Causality in commonsense reasoning about actions. Tech. rep.

MCCARTHY, J. 1977. Epistemological problems of Artificial Intelligence. In *Proc. of the Intl. Joint Conf. on Artificial Intelligence (IJCAI).* MIT Press, Cambridge, MA, 1038–1044.

MCCARTHY, J. 1998. Elaboration tolerance. In *Proc. of the 4th Symposium on Logical Formalizations of Commonsense Reasoning (Common Sense 98).* London, UK, 198–217. Updated version at `http://www-formal.stanford.edu/jmc/elaboration.ps`.

PEARCE, D. 2006. Equilibrium logic. *Ann. Math. Artif. Intell. 47,* 1-2, 3–41.

PEARL, J. 2000. *Causality: models, reasoning, and inference*. Cambridge University Press, New York, NY, USA.

PEREIRA, L. M., APARÍCIO, J. N., AND ALFERES, J. J. 1991. Derivation procedures for extended stable models. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, J. Mylopoulos and R. Reiter, Eds. Morgan Kaufmann, 863–869.

PONTELLI, E., SON, T. C., AND EL-KHATIB, O. 2009. Justifications for logic programs under answer set semantics. *Theory and Practice of Logic Programming 9,* 1, 1–56.

SCHULZ, C., SERGOT, M., AND TONI, F. 2013. Argumentation-based answer set justification. In *Proc. of the 11th Intl. Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*.

STUMME, G. 1997. Free distributive completions of partial complete lattices. *Order 14*, 179–189.

THIELSCHER, M. 1997. Ramification and causality. *Artificial Intelligence Journal 1-2,* 89, 317–364.

VAN BELLEGHEM, K., DENECKER, M., AND THESEIDER-DUPRÉ, D. 1998. A constructive approach to the ramification problem. In *Proceedings of ESSLLI*. Citeseer, 1–17.

VAN EMDEN, M. H. AND KOWALSKI, R. A. 1976. The semantics of predicate logic as a programming language. *J. ACM 23,* 4, 733–742.

VENNEKENS, J. 2011. Actual causation in cp-logic. *TPLP 11,* 4-5, 647–662.

Appendix A. Auxiliary figures

Associativity		Commutativity		Absorption	
$t + (u+w)$	$=$	$(t+u) + w$	$t + u$ $=$ $u + t$	t $=$ $t + (t*u)$
$t * (u*w)$	$=$	$(t*u) * w$	$t * u$ $=$ $u * t$	t $=$ $t * (t+u)$

Distributive		Identity	Idempotence	Annihilator
$t + (u*w)$ $=$ $(t+u) * (t+w)$		t $=$ $t + 0$	t $=$ $t + t$	1 $=$ $1 + t$
$t * (u+w)$ $=$ $(t*u) + (t*w)$		t $=$ $t * 1$	t $=$ $t * t$	0 $=$ $0 * t$

Fig. 4. Sum and product satisfy the properties of a completely distributive lattice.

Appendix B. An example of causal action theory

In this section we consider a more elaborated example from Pearl (2000).

Example 10

Consider the circuit in Figure 5 with two switches, *a* and *b*, and a lamp *l*. Note that *a* is the main switch, while *b* only affects the lamp when *a* is up. Additionally, when the light is on, we want to track which wire section, *v* or *w*, is conducting current to the lamp. □

As commented by Pearl (2000), the interesting feature of this circuit is that, seen from outside as a black box, it behaves exactly as a pair of independent, parallel switches, so it is impossible to detect the causal dependence between *a* and *b* by a mere observation of performed actions and their effects on the lamp. Figure 5 also includes a possible representation for this scenario; let us call it program P_5. It uses a pair of fluents $up(X)$ and $down(X)$ for the position of switch *X*, as well as *on* and *off* to represent the state of the lamp. Fluents $up(X)$ and $down(X)$ (respectively, *on* and *off*) can be seen as the strong negation of each other, although we do not use an operator for that purpose[6]. Action $m(X,D)$ stands for "move switch *X* in direction $D \in \{u,d\}$" (*up* and *down*, respectively). Actions between state *t* and $t+1$ are located in the resulting state. Finally, we have also labelled inertia laws (by *i*) to help keeping track of fluent justifications inherited by persistence.

Suppose we perform the following sequence of actions: we first move down both switches, next switch *b* is moved first up and then down, and finally we move up switch *a*. Assume also that each action occurrence is labelled with the action name so that, for instance, moving *b* up in Situation 1 corresponds to the program fact $m(b,u)_1 : m(b,u)_1$. The table in Figure 6 shows the resulting temporal projection. Note how the lamp turns on in Situation 1 but only because of *v*, that is, moving *a* down. Movements of *b* at 2 and 3 do not affect the lamp, and its causal explanation ($down(a)$) is maintained by inertia. In Situation 4, the lamp is still on but the reason has changed. The explanation this time is that we had closed down *b* at 3 (and this persisted by inertia) while we have just moved *a* up, firing rule *w*.

This example also illustrates why we are not interested in providing negative justifications through default negation. This would mean to explicitly include non-occurrences of actions that might

---

[6] Notice how strong negation would point out the cause(s) for a boolean fluent to take value false, whereas default negation represents the absence of cause.
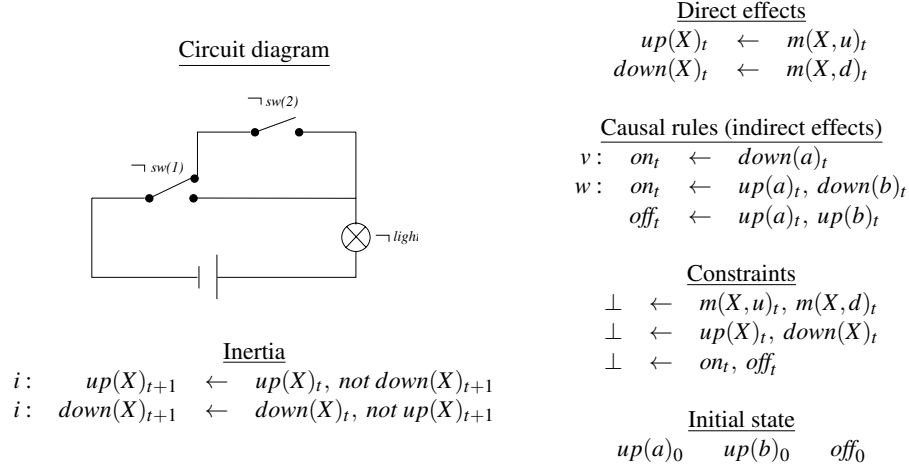
Circuit diagram

Direct effects

$$up(X)_t \quad \leftarrow \quad m(X,u)_t$$
$$down(X)_t \quad \leftarrow \quad m(X,d)_t$$

Causal rules (indirect effects)

$$v: \quad \underline{on_t \quad \leftarrow \quad down(a)_t}$$
$$w: \quad on_t \quad \leftarrow \quad up(a)_t, down(b)_t$$
$$off_t \quad \leftarrow \quad up(a)_t, up(b)_t$$

Constraints

$$\perp \quad \leftarrow \quad m(X,u)_t, m(X,d)_t$$
$$\perp \quad \leftarrow \quad up(X)_t, down(X)_t$$
$$\perp \quad \leftarrow \quad on_t, off_t$$

Inertia

$$i: \quad up(X)_{t+1} \quad \leftarrow \quad up(X)_t, not\ down(X)_{t+1}$$
$$i: \quad down(X)_{t+1} \quad \leftarrow \quad down(X)_t, not\ up(X)_{t+1}$$

Initial state

$$up(a)_0 \qquad up(b)_0 \qquad off_0$$

Fig. 5. A circuit with two switches together with a possible representation.

t	0	1	2	3	4
Actions		$m(a,d)_1, m(b,d)_1$	$m(b,u)_2$	$m(b,d)_3$	$m(a,u)_4$
$up(a)_t$	1	0	0	0	$m(a,u)_4$
$down(a)_t$	0	$m(a,d)_1$	$m(a,d)_1 \cdot i$	$m(a,d)_1 \cdot i$	0
$up(b)_t$	1	0	$m(b,u)_2$	0	0
$down(b)_t$	0	$m(b,d)_1$	0	$m(b,d)_3$	$m(b,d)_3 \cdot i$
on_t	0	$m(a,d)_1 \cdot v$	$m(a,d)_1 \cdot iv$	$m(a,d)_1 \cdot iv$	$(\,m(b,d)_3 \cdot i * m(a,u)_4\,) \cdot w$
off_t	1	0	0	0	0

Fig. 6. Temporal projection of a sequence of actions for program P_5.

otherwise have violated inertia. For instance, the explanation for on_2 would include the fact that we did not perform $m(a,u)_2$. Including this information for one transition is perhaps not so cumbersome, but suppose that, from 2 we executed a high number of transitions without performing any action. The explanation for on_3 would *additionally* collect that we did not perform $m(a,u)_3$ either. The explanation for on_4 should also collect the negation of further possibilities: moving a up at 4; three movements of a up, down and up; moving b at 3 and both switches at 4; moving both switches at 3 and b at 4; etc. It is easy to see that negative explanations grow exponentially: at step t we would get the negation of *all possible plans* for making on_t false, while indeed, *nothing has actually happened* (everything persisted by inertia).

Example 11 (*The gear wheels*)
Consider a gear mechanism with a pair of wheels, each one powered by a separate motor. Each motor has a switch to start and stop it. There is another switch to connect or disconnect the wheels. (McCain 1997). □

This example can be captured by the logic program P_6 formed by the following causal rules:

$$\begin{aligned}
motor(W)_t &\leftarrow start(W)_t & turn(1)_t &\leftarrow \quad turn(2)_t,\ coupled_t \\
\neg motor(W)_t &\leftarrow stop(W)_t & turn(2)_t &\leftarrow \quad turn(1)_t,\ coupled_t \\
turn(W)_t &\leftarrow motor(W)_t & \neg turn(1)_t &\leftarrow \neg turn(2)_t,\ coupled_t \\
coupled_t &\leftarrow couple_t & \neg turn(2)_t &\leftarrow \neg turn(1)_t,\ coupled_t \\
\neg coupled_t &\leftarrow uncouple_t
\end{aligned}$$

plus the fhe following inertia axioms:

$$\begin{aligned}
motor(W)_{t+1} &\leftarrow \quad motor(W)_t,\ not\ \neg motor(W)_{t+1} \\
\neg motor(W)_{t+1} &\leftarrow \neg motor(W)_t,\ not\ \quad motor(W)_{t+1} \\
turn(W)_{t+1} &\leftarrow \quad turn(W)_t,\ \ not\ \neg turn(W)_{t+1} \\
\neg turn(W)_{t+1} &\leftarrow \neg turn(W)_t,\ \ not\ \quad turn(W)_{t+1} \\
coupled_{t+1} &\leftarrow \quad coupled_t,\ \ not\ \neg coupled_{t+1} \\
\neg coupled_{t+1} &\leftarrow \neg coupled_t,\ \ not\ \quad coupled_{t+1}
\end{aligned} \tag{2}$$

Suppose that initially both motors are off an the wheels are immobile. This is reflected by the following set of facts $\{\ \neg motor(1)_0,\ \neg motor(2)_0,\ \neg turn(1)_0,\ \neg turn(2)_0,\ \neg cuopled_0\ \}$. Then we perform the following actions $\{\ s(1)_3 : start(1)_3,\ c_6 : couple_6,\ u_9 : uncouple_9\ \}$. Figure 7(a) shows the causal values associated with each fluent in each time interval. Note that we only have labelled the actions avoiding tracing rule application for clearity sake. This example illustrates the behaviour of causal logic programs in the presence of causal cycles. When no action is performed the value of a fluent is just propagate by inertia.

An iteresting variation of this example is incorporating some mechanic device to stop the wheels (Van Belleghem et al. 1998; Lin and Soutchanski 2011). This can be achived by adding the following set of rules:

$$\begin{aligned}
breaked(W)_t &\leftarrow break(W)_t & breaked(W)_{t+1} &\leftarrow \quad breaked(W)_t,\ not\ \neg breaked(W)_{t+1} \\
\neg breaked(W)_t &\leftarrow unbreak(W)_t & \neg breaked(W)_{t+1} &\leftarrow \neg breaked(W)_t,\ not\ \quad breaked(W)_{t+1} \\
\neg turn(W)_t &\leftarrow breaked(W)_t
\end{aligned}$$

If the second wheel is breaked at s_9 we will get insted that $turn(2)_9$ is false an the cause of $\neg turn(2)_t$ in any future situation is $break(2)_9$. Another interesiting situations is when we break the second wheel in the situation 5. In such case we have an incosistence. On the one hand the cause of $turn(1)_6$ and $turn(2)_6$ are respectively s_1 and $s(1)_3 * c(6)$. On the other hand the cause of $\neg turn(1)_6$ and $\neg turn(2)_6$ are respectively $break(2)_5 * c(6)$ and $break(2)_5$. Note that, causal values not only points the existence of an incosistence but also can be explaoited to explain why. In particual, in this case, the motor forces the wheels to spin whereas the break opposed to it.

Another iteresting variation of this example is assuming that wheels are breaked by friction as soon as the motriz force over them is absent. This can be achived by just replacing the inertia axiom (2) for *turn* by the following causal rule:

$$f(W)_t :\ \neg turn(W)_{t+1} \leftarrow not\ \neg turn(W)_t,\ not\ turn(W)_{t+1}$$

Figure 7(b) shows the variation in the causal value of $turn(2)$. Note that $*$ means that the value is the same as Figure 7(a).

t	$0-2$	$3-5$	$6-8$	$9-$		t	$0-8$	$9-$
$motor(1)_t$	0	$s(1)_3$	$s(1)_3$	$s(1)_3$		$motor(1)_t$	$*$	$*$
$\neg motor(1)_t$	1	0	0	0		$\neg motor(1)_t$	$*$	$*$
$coupled_t$	0	0	c_6	0		$coupled_t$	$*$	$*$
$\neg coupled_t$	1	1	0	u_9		$\neg coupled_t$	$*$	$*$
$turn(1)_t$	0	$s(1)_3$	$s(1)_3$	$s(1)_3$		$turn(1)_t$	$*$	$*$
$\neg turn(1)_t$	1	0	0	0		$\neg turn(1)_t$	$*$	$*$
$turn(2)_t$	0	0	$s(1)_3 * c_6$	$s(1)_3 * c_6$		$turn(2)_t$	$*$	0
$\neg turn(2)_t$	1	1	0	0		$\neg turn(2)_t$	$*$	$f(2)_9$
		(a) Innertial turning					(b) Friction brake	

Fig. 7. Temporal projection of a sequence of actions for program P_6.

Appendix C. Example with infinite rules

Example 12

Consider the infinite program P_7 given by the ground instances of the set of rules:

$$l(s(X)) : \quad nat(s(X)) \leftarrow nat(X)$$
$$l(z) \quad : \quad nat(z)$$

defining the natural numbers with a Peano-like representation, where z stands for "zero." For each natural number n, the causal value obtained for $nat(s^n(z))$ in the least model of the program is $l(z) \cdot l(s(z)) \dots l(s^n(z))$. Read from right to left, this value can be seen as the computation steps performed by a top-down Prolog interpreter when solving the query $nat(s^n(z))$. As a further elaboration, assume that we want to check that at least some natural number exists. For that purpose, we add the following rule to the previous program:

$$some \leftarrow nat(X) \tag{3}$$

The interesting feature of this example is that atom *some* collects an infinite number of causes from all atoms $nat(s^n(z))$ with n ranging among all the natural numbers. That is, the value for *some* is $I(some) = \alpha_0 + \alpha_1 + \alpha_2 + \dots + \alpha_n + \dots$ where $\alpha_n \stackrel{\text{def}}{=} l(z) \cdot l(s(z)) \cdot \dots \cdot l(s^n(z))$. However, it is easy to see that the fact $nat(z)$ labelled with $l(z)$ is not only sufficient to prove the existence of some natural number, but, due to the recursive definition of the natural numbers, it is also *necessary* – note how all the proofs α_i actually begin with an application of $l(z)$.

This fact is captured in our semantic by the algebraic equivalences showed in Fig 2. From associativity and identity of '$\cdot$', the following equivalence holds:

$$\alpha_n = 1 \cdot l(z) \cdot \beta_n \qquad \text{with} \qquad \beta_i \stackrel{\text{def}}{=} l(s(z)) \cdot \dots \cdot l(s^i(z))$$

for any $n \geq 1$. Furthermore, from absorption of '$\cdot$' w.r.t the addition, it also holds that

$$\alpha_0 + \alpha_n \quad = \quad l(z) + 1 \cdot l(z) \cdot \beta_n \quad = \quad l(z)$$

As a consequence, the previous infinite sum just collapses to $I(some) = l(z)$, reflecting the fact that, to prove the existence of a natural number, only the fact labelled as $l(z)$ is relevant.

Suppose now that, rather than defining natural numbers in a recursive way, we define them by

directly asserting an infinite set of facts as follows:

$$l(s^n(z)) : \ nat(s^n(z)) \tag{4}$$

for any $n \geq 0$, where $s^0(z)$ stands for z. In this variant, the causal value obtained for $nat(s^n(z))$ is simply $l(s^n(z))$, so that the dependence we had before on lower natural numbers does not exist any more. Adding rule (3) to the set of facts (4) allows us concluding $I(some) = l(z) + l(s(z)) + l(s^2(z)) + \ldots + l(s^n(z)) + \ldots$ and this infinite sum cannot be collapsed into any finite term. This reflects that we have now infinite *independent* ways to prove that some natural number exists.

This last elaboration can be more elegantly captured by replacing the infinite set of facts (4) by an auxiliary recursive predicate *aux* defined as follows:

$$aux(s(X)) \leftarrow aux(X) \qquad\qquad l(X) : \ nat(X) \leftarrow aux(X)$$
$$aux(z)$$

Since rules for *aux* are unlabelled, the value of $aux(s^n(z))$ in the least model is $I(aux(s^n(z))) = 1$ so the effect of this predicate is somehow "transparent" regarding causal justifications. As a result, the value of $nat(s^n(z))$ is just $l(s^n(z))$ as before.

Appendix D. Proofs

In order to improve clarity, for any causal graph $G = \langle V, E \rangle$, vertices v_1 and v_2 and edge (v_1, v_2) we respectively use the notation $v_1 \in G$ and $(v_1, v_2) \in G$ instead of $v_1 \in V$ and $(v_1, v_2) \in E$.

7.1 Causes as graphs

Proposition 1 (*Monotonicity*)
Let G, G' be a pair of causal graph with $G \leq G'$. Then, for any causal graph H:
$$G * H \leq G' * H, \qquad G \cdot H \leq G' \cdot H \qquad \text{and} \qquad H \cdot G \leq H \cdot G'$$

Proof. First we will show that $G * H \leq G' * H$. Suppose that $E(G * H) \not\supseteq E(G' * H)$ and let (l_1, l_2) be an edge in $E(G' * H)$ but not in $E(G * H')$, i.e. $(l_1, l_2) \in E(G' * H) \backslash E(G * H')$.

Thus, since by product definition $E(G' * H) = E(G') \cup E(H)$, it follows that either $(l_1, l_2) \in E(G')$ or $(l_1, l_2) \in E(H)$. It is clear that if $(l_1, l_2) \in E(H)$ then $(l_1, l_2) \in E(G * H) = E(G) \cup E(H)$. Furthermore, since $G \leq G'$ it follows that $E(G) \supseteq E(G')$, if $(l_1, l_2) \in E(G')$ then $(l_1, l_2) \in E(G)$ and consequently $(l_1, l_2) \in E(G * H) = E(G) \cup E(H)$. That is $E(G * H) \supseteq E(G' * H)$ and then $G * H \leq G' * H$. Note that $V(G * H) \supseteq V(G' * H)$ follows directly from $E(G * H) \supseteq E(G' * H)$ and the fact that every vertex has and edge to itself.

To show that $G \cdot H \leq G' \cdot H$ (the case for $H \cdot G \leq H \cdot G'$ is analogous) we have has to show that, in addition to the previous, for every edge $(l_G, l_H) \in E(G' \cdot H)$ with $l_G \in V(G')$ and $l_H \in V(H)$ it holds that $(l_G, l_H) \in E(G \cdot H)$. Simply note that since $G \leq G'$ it follows $V(G) \supseteq V(G)'$ and then $l_G \in V(G)$. Consequently $(l_G, l_H) \in E(G \cdot H)$. $\qquad \square$

Proposition 2 (*Application associativity for causal graphs*)
Let G_1, G_2 and G_3 be three causal graphs. Then $G_1 \cdot (G_2 \cdot G_3) = (G_1 \cdot G_2) \cdot G_3$.

Proof. By definition, it follows that

$$V\big((G_1 \cdot G_2) \cdot G_3\big) = (V(G_1) \cup V(G_2)) \cup V(G_3) = V(G_1) \cup (V(G_2) \cup V(G_3)) \qquad (5)$$
$$E\big((G_1 \cdot G_2) \cdot G_3\big) = \big(E(G_1) \cup E(G_2) \cup E_{12}\big) \cup E(G_3) \cup E_{12,3}$$
$$= E(G_1) \cup E(G_2) \cup E(G_3) \cup E_{12} \cup E_{12,3} \qquad (6)$$
$$E\big(G_1 \cdot (G_2 \cdot G_3)\big) = E(G_1) \cup \big(E(G_2) \cup E(G_3) \cup E_{23}\big) \cup E_{1,23}$$
$$= E(G_1) \cup E(G_2) \cup E(G_3) \cup E_{1,23} \cup E_{23} \qquad (7)$$

where

$$E_{12} \stackrel{\text{def}}{=} \{ (v_1, v_2) \mid v_1 \in V_1 \text{ and } v_2 \in V_2 \} \quad E_{12,3} \stackrel{\text{def}}{=} \{ (v_{12}, v_3) \mid v_{12} \in V_1 \cup V_2 \text{ and } v_3 \in V_3 \}$$
$$E_{23} \stackrel{\text{def}}{=} \{ (v_2, v_3) \mid v_2 \in V_2 \text{ and } v_3 \in V_3 \} \quad E_{1,23} \stackrel{\text{def}}{=} \{ (v_1, v_{23}) \mid v_1 \in V_1 \text{ and } v_{23} \in V_2 \cup V_3 \}$$

From (5) and (7), it follows that

$$(G_1 \cdot G_2) \cdot G_3 = G_1 \cdot (G_2 \cdot G_3) \qquad \text{if and only if } E_{12} \cup E_{12,3} = E_{1,23} \cup E_{23}$$
$$\text{if and only if } E_{12,3} \subseteq E_{1,23} \cup E_{23} \text{ and } E_{1,23} \subseteq E_{12} \cup E_{12,3}$$

Note that $E_{12} \subseteq E_{1,23}$ and $E_{23} \subseteq E_{12,3}$. Then, we will show that $E_{12,3} \subseteq E_{1,23} \cup E_{23}$. Suppose there is an edge $(v_{12}, v_3) \in E_{12,3}$ such that $(v_{12}, v_3) \notin E_{23}$ and $(v_{12}, v_3) \notin E_{1,23}$. Since (v_{12}, v_3), $v_{12} \in V_1 \cup V_2$ and $v_3 \in V_3$. If $v_{12} \in V_1$, then $(v_{12}, v_3) \in E_{12,3}$ which is a contradiction, and if,

otherwise, $v_{12} \in V_2$, then $(v_{12}, v_3) \in E_{23}$ which is also a contradiction. The case $E_{1,23} \subseteq E_{12} \cup E_{12,3}$ is symmetric.

Proposition 3
For every causal graph $G = \langle V, E \rangle$ it holds that $G = \prod \{\, l \cdot l' \mid (l, l') \in E \,\}$.

Proof. Let G' be a causal graphb s.t. $G' = \prod \{\, l \cdot l' \mid (l, l') \in E \,\}$. Then for every edge $(l, l') \in E(G)$ it holds that $(l, l') \in E(l \cdot l')$ and then $(l, l') \in E(G') = \bigcup \{\, E(l \cdot l') \mid (l, l') \in E \,\}$, i.e. $E(G) \subseteq E(G')$. Furthermore for every $(l, l') \in E(G')$ there is $l_i \cdot l_j$ s.t. $(l, l') \in l_i \cdot l_j$ and $(l_i, l_j) \in E(G)$. Then, since $E(l_i \cdot l_j) = \{(l_i, l_j)\}$ it follows that $(l, l') \in E(G)$, i.e. $E(G) \supseteq E(G')$. Consequently $G = G' = \prod \{\, l \cdot l' \mid (l, l') \in E \,\}$. $\square$

Proposition 4 (*Infimum*)
Any set of causal graphs S has a $\leq$-infimum given by their product $\prod S$.

Proof. By definition $\prod S$ is the causal graph whose vertices and edges are respectively the sets $V(\prod S) = \bigcup \{\, V(G) \mid G \in S \,\}$ and $E(\prod S) = \bigcup \{\, E(G) \mid G \in S \,\}$. It is easy to see that $\prod S$ is the supremum of the subgraph relation, so that, since for every pair of causal graphs $G \leq G'$ iff $G \supseteq G'$, it follows that infimum of S w.r.t. $\leq$. $\square$

Proposition 5 (*Application distributivity w.r.t. products over causal graphs*)
For every pair of sets of causal graphs S and S', it holds that

$$\left(\prod S\right) \cdot \left(\prod S'\right) = \prod \{\, G \cdot G' \mid G \in S \text{ and } G' \in S' \,\}.$$

Proof. For readability sake, we define two causal graphs

$$G_R \stackrel{\text{def}}{=} \left(\prod S\right) \cdot \left(\prod S'\right) \qquad \text{and} \qquad G_L \stackrel{\text{def}}{=} \prod \{\, G \cdot G' \mid G \in S \text{ and } G' \in S' \,\}$$

and we assume that both S and S' are not empty sets. Note that $\prod \emptyset = \mathbf{C}_{Lb} = \prod \{G_\emptyset\}$. Then, by product definition, it follows that

$$E(G_L) = \left(\bigcup \{\, E(G) \mid G \in S \,\} \cup \bigcup \{\, E(G') \mid G \in S' \,\} \cup E_L\right)^*$$

$$E(G_R) = \left(\bigcup \{\, E(G) \cup E(G') \cup E_R(G, G') \mid G \in S \text{ and } G' \in S' \,\}\right)^*$$

where
$$E_L = \{\, (l, l') \mid l \in \bigcup \{\, V(G) \mid G \in S \,\} \text{ and } l' \in \bigcup \{\, V(G') \mid G' \in S' \,\} \,\}$$
$$E_R(G, G') = \{\, (l, l') \mid l \in V(G) \text{ and } l' \in V(G') \,\}$$

Furthermore let $E_R = \bigcup \{\, E_R(G, G') \mid G \in S \text{ and } G' \in S' \,\}$. For every edge $(l, l') \in E_L$ there are a pair of c-graphs $G \in S$ and $G' \in S'$ s.t. $l \in V(G)$ and $l' \in V(G')$ and then $(l, l') \in E_R(G, G')$ and so $(l, l') \in E_R$. Moreover, for every edge $(l, l') \in E_R$ there are a pair of c-graphs $G \in S$ and $G' \in S'$ s.t. $(l, l') \in E_R(G, G')$ with $l \in V(G)$ and $l' \in V(G)'$. So that $(l, l') \in E_L$. That is $E_L = E_R$. Then

$$E(G_R) = \left(\bigcup \{\, E(G) \mid G \in S \,\} \cup \bigcup \{\, E(G') \mid G' \in S' \,\} \cup E_R\right)^*$$
$$= \left(E(G_L) \backslash E_L \cup E_R\right)^* = \left(E(G_L)\right)^* = E(G_L)$$

Consequently $G_L = G_R$. $\square$

Proposition 6 (*Transitive application distributivity w.r.t. products over causal graphs*)
For any causal graphs G, $G' \neq \emptyset$ and G'', it holds that

$$G \cdot G' \cdot G'' = G \cdot G' * G' \cdot G''$$

Proof. It is clear that $G \cdot G' \cdot G'' \leq G \cdot G'$ and $G \cdot G' \cdot G'' \leq G' \cdot G''$ and then $G \cdot G' \cdot G'' \leq G \cdot G' * G' \cdot G''$. Let G_1, G_2, G_L and G_R be respectively $G_1 = G \cdot G'$, $G_2 = G' \cdot G''$, $G_L = G_1 \cdot G''$, and $G_R = G_1 * G_2$. Suppose that $G_L < G_R$, i.e. $G_L \supset G_R$ and there is an edge $(v_1, v_2) \in G_L$ but $(v_1, v_2) \notin G_R$. Then $G_1 \subseteq G_R$ and $G'' \subseteq G_2 \subseteq G_R$ and one of the following conditions holds:

1. $(v_1, v_2) \in G_1 \subseteq G_R$ or $(v_1, v_2) \in G'' \subseteq G_R$ which is a contradiction with $(v_1, v_2) \notin G_R$.
2. $v_1 \in G_1$ and $v_2 \in G''$, i.e. $v_1 \in G$ and $v_2 \in G''$ or $v_1 \in G'$ and $v_2 \in G''$. Furthermore, if the last it is clear that $(v_1, v_2) \in G' \cdot G'' = G_2 \subseteq G_R$ which is a contradiction with $(v_1, v_2) \notin G_R$.

Thus it must be that $v_1 \in G$ and $v_2 \in G''$. But then, since $G' \neq \emptyset$ there is some $v' \in G'$ and consequently there are edges $(v_1, v') \in G \cdot G' = G_1 \subseteq G_R$ and $(v', v_2) \in G' \cdot G'' = G_2 \subseteq G_R$. Since G_R is closed transitively, $(v_1, v_2) \in G_R$ which is a contradiction with the assumption that $(v_1, v_2) \notin G_R$. That is, $G_L = G \cdot G' \cdot G'' = G \cdot G' * G' \cdot G'' = G_R$.

Proposition 7 (*Application idempotence w.r.t. singleton causal graphs*)
For any causal graph G whose only edge is (l, l), $G \cdot G = G$.

Proof. By definition $G \cdot G = G \cup G \cup \{ (v_1, v_2) \mid v_1 \in G \text{ and } v_2 \in G \}$. Since the only vertex of G is l, it follows that $G \cdot G = G \cup \{(l, l)\} = G$. $\qquad\square$

Proposition 8 (*Application absorption over causal grapsh*)
Let G_1, G_2 and G_3 be three causal graphs. Then $G_1 \cdot G_2 \cdot G_3 = G_2 * G_1 \cdot G_2 \cdot G_3$.

Proof. By definition, it is clear that $G_1 \cdot G_2 \cdot G_3 \supseteq G_2$ and then

$$G_2 * G_1 \cdot G_2 \cdot G_3 = (G_2 \cup G_1 \cdot G_2 \cdot G_3)^* = (G_1 \cdot G_2 \cdot G_3)^* = G_1 \cdot G_2 \cdot G_3$$

Proposition 9
Let a and b be elements of an algebra holding the identity and absorption equivalences showed in Figure 2. Then

$$a * a \cdot b = a \cdot b \qquad\qquad\qquad a * b \cdot a = b \cdot a$$

Proof. The proof follow from the following equivalences:

$$
\begin{aligned}
a * a \cdot b &= a * 1 \cdot a \cdot b && \text{(identity)} & a * b \cdot a &= a * b \cdot a \cdot 1 && \text{(identity)} \\
&= 1 \cdot a \cdot b && \text{(absorption)} & &= b \cdot a \cdot 1 && \text{(absorption)} \\
&= a \cdot b && \text{(identity)} & &= b \cdot a && \text{(identity)}
\end{aligned}
$$

Proposition 10

Let a, b and c be elements of an algebra holding the identity and absorption equivalences showed in Figure 2 such that b is different from 1. Then

$$a \cdot b * b \cdot c = a \cdot b * b \cdot c * a \cdot c$$

follows from application associative, identity and absorption and distributivity over products.

Proof.

$$
\begin{aligned}
a \cdot b * b \cdot c * a \cdot c &= (a \cdot b * b \cdot c) * a \cdot c && \text{(associative `$*$')} \\
&= (a \cdot b \cdot c) * a \cdot c && \text{(transitivity)} \\
&= a \cdot (b \cdot c) * a \cdot c && \text{(associative `$\cdot$')} \\
&= a \cdot (b \cdot c * c) && \text{(distributivity)} \\
&= a \cdot (b \cdot c) && \text{(absorption der)} \\
&= a \cdot b \cdot c && \text{(associativity)} \\
&= a \cdot b * b \cdot c && \text{(transitivity)}
\end{aligned}
$$

Corollary 1

Given causal graphs G_1, G_2, G_3 and G_l such that (l, l) is the only edge of G_l, the equivalences reflected in the Figures 8 and 9 hold.

Idempotence	Associativity	Product distributivity
$G_l \cdot G_l \quad = \quad G_l$	$G_1 \cdot (G_2 \cdot G_3) \quad = \quad (G_1 \cdot G_2) \cdot G_3$	$G_1 \cdot (G_2 * G_3) \quad = \quad (G_1 \cdot G_2) * (G_1 \cdot G_3)$ $(G_1 * G_2) \cdot G_3 \quad = \quad (G_1 \cdot G_3) * (G_2 \cdot G_3)$

Absorption	Identity	Transitivity
$G_2 * G_1 \cdot G_2 \cdot G_3 \quad = \quad G_1 \cdot G_2 \cdot G_3$	$G_1 \cdot G_\emptyset = G_1$ $G_\emptyset \cdot G_1 = G_1$	$G_1 \cdot G_2 \cdot G_3 = (G_1 \cdot G_2) * (G_2 \cdot G_3)$ with $G_2 \neq G_\emptyset$

Fig. 8. Properties of the `$\cdot$' and `$*$' operators over causal graphs (G_l only contains the edge (l, l)).

Absorption (der)	Transitivity (der)
$G_1 * (G_1 \cdot G_2) \quad = \quad G_1 \cdot G_2$ $G_1 * (G_2 \cdot G_1) \quad = \quad G_2 \cdot G_1$	$(G_1 \cdot G_2) * (G_2 \cdot G_3) = (G_1 \cdot G_2) * (G_2 \cdot G_3) * (G_1 \cdot G_3)$

Fig. 9. Properties following from those in Figure 8.

Theorem 7

Given a set of labels *Lb*, $\langle \mathbf{C}_{Lb}, *, \cdot \rangle$ is the free algebra generated by *Lb* defined by equations in the Figure 8, i.e. the mapping $graph : Lb \longrightarrow \mathbf{C}_{Lb}$ mapping each label *l* to the graph G_l containing the only edge (l, l) is an injective (preserving-idempotence) homomorphism and for any set *F* and idempotence-preserving map $\delta : Lb \longrightarrow F$, there exists a homomorphism $term : \mathbf{C}_{Lb} \longrightarrow F$ defined as $term(G) \mapsto \prod \{ \delta(l_1) \cdot \delta(l_2) \mid (l_1, l_2) \in G \}$ such that $\delta = term \circ graph$.

Proof. For clarity sake we omit the idempotence-preserving mapping δ and we write just *l* instead of $\delta(l)$. Thus the mapping $term : \mathbf{C}_{Lb} \longrightarrow F$ is just $term(G) \mapsto \prod \{ l_1 \cdot l_2 \mid (l_1, l_2) \in G \}$. We start showing that *graph* preservers the '$\cdot$' idempotence equation.

$$graph(l) \cdot graph(l) = G_l \cdot G_l = G_l \cup G_l \cup \{ (l_1, l_2) \mid l_1 \in G_l \text{ and } l_2 \in G_l \}$$
$$= G_l \cup G_l \cup \{ (l, l) \} = G_l \cup G_l \cup G_l = G_l = graph(l)$$

Furthermore, suppose $graph(l_1) = graph(l_2)$, i.e. $\{ (l_1, l_2) \} = \{ (l_2, l_2) \}$. Then $l_1 = l_2$. Hence *graph* is an injective homomorphism. Now we show that *term* preserves the '$*$'. Let $D = \bigcup_{G \in U} G$. Then

$$\prod_{G \in U} term(G) = \prod_{G \in U} \prod_{(l_1, l_2) \in G} l_1 \cdot l_2 = \prod_{(l_1, l_2) \in D} l_1 \cdot l_2$$
$$= \prod_{(l_1, l_2) \in D^*} l_1 \cdot l_2 = term(D^*)$$
$$= term \left(\left(\bigcup_{G \in U} G \right)^* \right) = term \left(\prod_{G \in U} G \right)$$

We will show now that *term* also preserves '$\cdot$'.

$$term(G_1) \cdot term(G_2) = \left(\prod_{(u_1, u_2) \in G_u} u_1 \cdot u_2 \right) \cdot \left(\prod_{(v_1, v_2) \in G_v} v_1 \cdot v_2 \right)$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} (u_1 \cdot u_2) \cdot (v_1 \cdot v_2)$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} u_1 \cdot (u_2 \cdot v_1 \cdot v_2)$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} u_1 \cdot (u_2 \cdot v_1 * v_1 \cdot v_2)$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} u_1 \cdot u_2 \cdot v_1 * u_1 \cdot v_1 \cdot v_2$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} u_1 \cdot u_2 * u_2 \cdot v_1 * u_1 \cdot v_1 * v_1 \cdot v_2 * u_1 \cdot v_1 * u_1 \cdot v_2$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} (u_1 \cdot u_2 * v_1 \cdot v_2) * (u_2 \cdot v_1 * u_1 \cdot v_1 * u_1 \cdot v_1 * u_1 \cdot v_2)$$
$$= \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} (u_1 \cdot u_2 * v_1 \cdot v_2) * \prod_{(u_1, u_2) \in G_u, (v_1, v_2) \in G_v} (u_2 \cdot v_1 * u_1 \cdot v_1 * u_1 \cdot v_1 * u_1 \cdot v_2)$$
$$= \prod_{(l_1, l_2) \in G_u \cup G_v} (l_1 \cdot l_2) * \prod_{u \in G_u, v \in G_v} (u \cdot v)$$
$$= \prod_{(l_1, l_2) \in G_u \cdot G_v} (l_1 \cdot l_2) = term(G_1 \cdot G_2)$$

Finally note that $term(graph(l)) = l \cdot l = l$ (note that *l* stands for the more formally $\delta(l)$).

7.2 Alternative causes

We define an explanation formed by a set of causal graphs S as:

$$\sum S = G \qquad\qquad \text{if } G \in S \text{ and } G * G' = G' \text{ for all } G' \in S$$

Theorem 8

The set of causal values $\mathbf{V}_{Lb}$ with the opperations join '+' and meet '$*$' forms the free, complete distributive lattice with generated by the set of causal graphs $\mathbf{C}_{Lb}$ and further $\downarrow\colon \mathbf{C}_{Lb} \longrightarrow \mathbf{V}_{Lb}$ is an injective homomorphism from the algebra $\langle \mathbf{C}_{Lb}, +, *, \cdot \rangle$ to $\langle \mathbf{V}_{Lb}, +, *, \cdot \rangle$. . $\qquad\square$

Proof of Theorem **??**. Let $\mathbf{F}$ be the set of filters over the lower semilattice $\langle \mathbf{C}_{Lb}, * \rangle$. Stumme was showed in (Stumme 1997) that the concept lattice $\underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{V}_{Lb}, \Delta \rangle$ (with $F \Delta I \Leftrightarrow F \cap I \neq \emptyset$) is isomorphic to the free completely distributive complete lattice generated by the partial lattice $\langle \mathbf{C}_{Lb}, +, * \rangle$ where $+$ and $*$ are two partial functions corresponding with the supremum and infimum. In our particular, case for every set of causal graphs S its infimum is defined as $\prod S$ and the supremum is defined as $G \in S$ such that $G' \leq G$ for all $G' \in S$, when such G exists and undefined otherwise. Thus $\mathbf{V}_{Lb}$ is the set of ideals over the partial lattice $\langle \mathbf{C}_{Lb}, +, * \rangle$, i.e. every $I \in \mathbf{V}_{Lb}$ is also closed under defined suprema. He also show that the elements of such lattice are described as pairs

$$\big\{ \, (\mathbf{F}_t, \mathbf{I}_t) \,\big|\, \mathbf{F}_t \subseteq \mathbf{F}, \ \mathbf{I}_t \subseteq \mathbf{V}_{Lb}, \ \mathbf{F}_t^I = \mathbf{I}_t \text{ and } \mathbf{F}_t = \mathbf{I}_t^I \, \big\}$$

where

$$\mathbf{F}_t^I = \big\{ \, I \in \mathbf{I} \,\big|\, \forall F \in \mathbf{F}_t : F \cap I \neq \emptyset \, \big\}$$
$$\mathbf{I}_t^I = \big\{ \, F \in \mathbf{F} \,\big|\, \forall I \in \mathbf{I}_t : F \cap I \neq \emptyset \, \big\}$$

That is, every element is in the form $\langle \mathbf{I}_t^I, \mathbf{I}_t \rangle$. Furthermore infima and suprema can be described as follows:

$$\bigwedge_{t \in T} (\mathbf{I}_t^I, \mathbf{I}_t) = \left(\bigcap_{t \in T} \mathbf{I}_t^I, \Big(\bigcup_{t \in T} \mathbf{I}_t \Big)^{II} \right)$$
$$\bigvee_{t \in T} (\mathbf{I}_t^I, \mathbf{I}_t) = \left(\Big(\bigcup_{t \in T} \mathbf{I}_t^I \Big)^{II}, \ \bigcap_{t \in T} \mathbf{I}_t \ \right)$$

We will show that $\varepsilon_I : \underline{\mathfrak{B}} \longrightarrow \mathbf{V}_{Lb}$ given by $(\mathbf{I}_t^I, \mathbf{I}_t) \mapsto \bigcap \mathbf{I}_t$ is an isomorphism between $\langle \underline{\mathfrak{B}}, \vee, \wedge \rangle$ and $\langle \mathbf{V}_{Lb}, \cup, \cap \rangle$. Note that, since $\prod \emptyset = G_\emptyset$ it holds that the empty set is not close under defined infimum and then it is not a filter, i.e. $\emptyset \notin \mathbf{F}$, and then for every filter $F \in \mathbf{F}$ it holds that $G_\emptyset \in F$. Thus if $\mathbf{I}_t = \emptyset$ follows that $\mathbf{I}_t^I = \mathbf{F}$ and then $\mathbf{I}_t^{II} = \{ I \in \mathbf{V}_{Lb} \mid G_\emptyset \in I \} = \mathbf{C}_{Lb} \neq \mathbf{I}_t$. That is, $\langle \emptyset^I, \emptyset \rangle \notin \underline{\mathfrak{B}}$.

We will show that for every ideal $I_t \in \mathbf{I}$ and for every set of ideals $\mathbf{I}_t \subseteq \mathbf{I}$ s.t. $I_t = \bigcap \mathbf{I}_t$ it holds that

$$(\mathbf{I}_t^I, \mathbf{I}_t) \in \underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{I}, \Delta \rangle \Longleftrightarrow \mathbf{I}_t = \big\{ \, I \in \mathbf{I} \,\big|\, I_t \subseteq I \, \big\} \tag{8}$$

and consequently ε_I is a bijection between and $\underline{\mathfrak{B}}$ and $\mathbf{V}_{Lb}$.

Suppose that $(\mathbf{I}_t^I, \mathbf{I}_t) \in \underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{I}, \Delta \rangle$. For every $I \in \mathbf{I}_t$ it holds that $I_t \subseteq I$. So suppose there is $I \in \mathbf{I}$ s.t. $I_t \subseteq I$ and $I \notin \mathbf{I}_t$. Then there is $F \in \mathbf{I}_t^I$ s.t. $I \cap F = \emptyset$ and for every element $I' \in \mathbf{I}_t$ it holds that $I' \cap F \neq \emptyset$. Pick a causal graph G s.t. $G = \prod \{ G' \mid G' \in I' \cap F$ and $I' \in \mathbf{I}_t \}$. Since for every G' it holds $G' \in F$ and $G \leq G'$ follows that $G \in F$ (F is close under infimum) and $G \in I'$ (every I' is close under $\leq$). That is, for every $I' \in \mathbf{I}_t$ it holds that $G \in I' \cap F$ and then, since $I_t = \bigcap \mathbf{I}_t$, it also

holds that $G \in I_t \cap F$ and since $I_t \subseteq I$ also $G \in I \cap F$ which contradict that $I \cap F = \emptyset$. So that $I \in \mathbf{I}_t$ and it holds that

$$(\mathbf{I}_t^I, \mathbf{I}_t) \in \underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{I}, \Delta \rangle \Longrightarrow \mathbf{I}_t = \{\, I \in \mathbf{I} \mid I_t \subseteq I \,\}$$

Suppose that $\mathbf{I}_t = \{I \in \mathbf{I} \mid I_t \subseteq I\}$ but $(\mathbf{I}_t^I, \mathbf{I}_t) \in \underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{I}, \Delta \rangle$, i.e. $\mathbf{I}_t \neq \mathbf{I}_t^{II}$. Note that $\mathbf{I}_t \subseteq \mathbf{I}_t^{II}$ because otherwise there are $I \in \mathbf{I}_t$ and $F \in \mathbf{I}_t^I$ s.t. $I \cap F = \emptyset$ which is a contradiction with the fact that for every $F \in \mathbf{I}_t^I$ and $I \in \mathbf{I}_t$ it holds that $F \cap I \neq \emptyset$.

So, there is $I \in \mathbf{I}_t^{II}$ s.t. $I \notin \mathbf{I}_t$, i.e. for every $F \in \mathbf{I}_t^I$ it holds that $F \cap I \neq \emptyset$ but $I_t \not\subseteq I$. Pick $G \in I_t \setminus I$ and $F = \{G' \mid G \leq G'\}$. It is clear that $F \in \mathbf{F}$ and $F \cap I_t \neq \emptyset$ because $G \in I_t$, so that $F \in \mathbf{I}_t^I$. Furthermore $F \cap I = \emptyset$, because $G \notin I$, which is a contradiction with the assumption. Thus

$$(\mathbf{I}_t^I, \mathbf{I}_t) \in \underline{\mathfrak{B}}\langle \mathbf{F}, \mathbf{I}, \Delta \rangle \Longleftarrow \mathbf{I}_t = \{\, I \in \mathbf{I} \mid I_t \subseteq I \,\}$$

Now, we will show that $(\mathbf{I}_1^I, \mathbf{I}_1) \vee (\mathbf{I}_2^I, \mathbf{I}_2) = (\mathbf{I}_3^I, \mathbf{I}_3)$ iff $I_1 \cup I_2 = I_3$. From the above statement follows that

$$\mathbf{I}_1 \cap \mathbf{I}_2 = \{I \in \mathbf{I} \mid I_1 \subseteq I \text{ and } I_2 \subseteq I\} =$$
$$= \{I \in \mathbf{I} \mid I_1 \cup I_2 \subseteq I\}$$
$$\mathbf{I}_3 = \{I \in \mathbf{I} \mid I_3 \subseteq I\}$$

That is, $\mathbf{I}_1 \cap \mathbf{I}_2 = \mathbf{I}_3$ iff $I_1 \cup I_2 = I_3$ and by definition of $\vee$ the first is equivalent to $(\mathbf{I}_1^I, \mathbf{I}_1) \vee (\mathbf{I}_2^I, \mathbf{I}_2) = (\mathbf{I}_3^I, \mathbf{I}_3)$.

Finally we will show that $(\mathbf{I}_1^I, \mathbf{I}_1) \wedge (\mathbf{I}_2^I, \mathbf{I}_2) = (\mathbf{I}_3^I, \mathbf{I}_3)$ iff $I_1 \cap I_2 = I_3$. It holds that

$$(\mathbf{I}_1 \cup \mathbf{I}_2)^{II} = \left(\{\, I \in \mathbf{I} \mid I_1 \subseteq I \text{ or } I_2 \subseteq I \,\}\right)^{II} =$$
$$= \left(\{\, I \in \mathbf{I} \mid I_1 \cap I_2 \subseteq I \,\}\right)^{II}$$
$$\mathbf{I}_3 = \{\, I \in \mathbf{I} \mid I_3 \subseteq I \,\}$$

Since ε_I is a bijection, it holds that $(\mathbf{I}_1 \cup \mathbf{I}_2)^{II} = \mathbf{I}_3$ iff $I_1 \cap I_2 = I_3$.

Thus $\varepsilon_I : \underline{\mathfrak{B}} \longrightarrow \mathbf{V}_{Lb}$ is an isomorphism between $\langle \underline{\mathfrak{B}}, \vee, \wedge \rangle$ and $\langle \mathbf{V}_{Lb}, \cup, \cap \rangle$, i.e. $\langle \mathbf{V}_{Lb}, \cup, \cap \rangle$ is isomorphic to the free completely distributive lattice generated by $\langle \mathbf{C}_{Lb}, +, * \rangle$.

Let's check now that $\downarrow : \mathbf{C}_{Lb} \longrightarrow \mathbf{V}_{Lb}$ is an injective homomorphism. Stumme has already showed that $\varepsilon_p : \mathbf{C}_{Lb} \longrightarrow \underline{\mathfrak{B}}$ given by

$$\varepsilon_p(G) \mapsto \left(\{\, F \in \mathbf{F} \mid G \in F \,\}, \{\, I \in \mathbf{I} \mid G \in I \,\}\right)$$

is an injective homomorphism between the partial lattice $\langle \mathbf{C}_{Lb}, +, * \rangle$ and $\langle \underline{\mathfrak{B}}, \vee, \wedge \rangle$. So that $\varepsilon_I \circ \varepsilon_p$ is an injective homomorphism between $\langle \mathbf{C}_{Lb}, +, * \rangle$ and $\langle \mathbf{V}_{Lb}, \cup, \cap \rangle$ given by

$$\varepsilon_I \circ \varepsilon_p(G) \mapsto \bigcap \{\, I \in \mathbf{V}_{Lb} \mid G \in I \,\} = {\downarrow}G$$

Note that for any causal graph G and $G' \in \mathbf{C}_{Lb}$ s.t. $G' \leq G$ it holds that $G' \in \varepsilon_I \circ \varepsilon_p(G)$, that is ${\downarrow}G \subseteq \varepsilon_I \circ \varepsilon_p(G)$. Furthermore for every causal graph G it holds that $\varepsilon(G)$ is an ideal, i.e. ${\downarrow}G \in \mathbf{V}_{Lb}$ and it is clear that $G \in {\downarrow}G$ so that, $\varepsilon_I \circ \varepsilon_p$ is an intersection with ${\downarrow}G$ as one of its operands, thus $\varepsilon_I \circ \varepsilon_p(G) \subseteq {\downarrow}G$. That is ${\downarrow}G = \varepsilon_I \circ \varepsilon_p(G)$ and consequently it is an injective homomorphism between $\langle \mathbf{C}_{Lb}, +, * \rangle$ $\langle \mathbf{V}_{Lb}, \cup, \cap \rangle$.

Let us show now that the mapping $\downarrow$ also preserves the '$\cdot$' operation. Take any causal graphs G_1 and G_2, then

$$\downarrow G_1 \cdot \downarrow G_2 = \downarrow \{ (G_1' \cdot G_2') \mid G_1' \in \downarrow G_1 \text{ and } G_2' \in \downarrow G_2 \}$$
$$= \downarrow \{ (G_1' \cdot G_2') \mid G_1' \le G_1 \text{ and } G_2' \le G_2 \} = \downarrow (G_1 \cdot G_2)$$

Note that, from Proposition 1, it follows that $G_1' \cdot G_2' \le G_1 \cdot G_2$. That is, $\downarrow$ in an homomorphism between $\langle \mathbf{C}_{Lb}, +, *\cdot \rangle$ and $\langle \mathbf{V}_{Lb}, +, *\cdot \rangle$.

Corollary 2
Every equivalence showed in Figure 4 hold. Futheremore equivalences showed in Figures 8 and 9 also hold if principal ideals $\downarrow G_1$, $\downarrow G_2$ and $\downarrow G_3$ are considered instead of causal graphs G_1, G_2 and G_3.

Proposition 11
For every causal term without addition sums t there exists a causal graph G such that $t = \downarrow G$.

Proof. In case that t is a label, by definition is the principal ideal $\downarrow t$. We proceed by structural induction. In case that $t = \prod S$, by induction hypothesis, for every $t_i \in S$ there is some causal graph G_i such that $t_i = \downarrow G_i$ and from Theorem 1 it follows that

$$t = \prod S = \bigcap \{ \downarrow G_i \mid t_i \in S \} = \downarrow \prod \{ G_i \mid t_i \in S \}$$

which is a principal ideal. In case that $t = t_1 \cdot t_2$, by induction hypothesis, $t_1 = \downarrow G_1$ and $t_2 = \downarrow G_2$ and from Proposition 16, it follows that

$$t = t_1 \cdot t_2 = \downarrow G_1 \cdot \downarrow G_2 = \downarrow (G_1 \cdot G_2)$$

which is a principal ideal.

Corollary 3
Given causal terms without sums c, d, e and a label l, the equivalences reflected in the Figures 10 and 11 hold.

Idempotence	Associativity	Product distributivity	
$l \cdot l \;=\; l$	$c \cdot (d \cdot e) \;=\; (c \cdot d) \cdot e$	$c \cdot (d * e) \;=\; (c \cdot d) * (c \cdot e)$	
		$(c * d) \cdot e \;=\; (c \cdot e) * (d \cdot e)$	
Identity	Absorption	Transitivity	
$c \cdot 1 \;=\; c$	$d * c \cdot d \cdot e \;=\; c \cdot d \cdot e$	$c \cdot d \cdot e \;=\; (c \cdot d) * (d \cdot e)$ with $d \ne 1$	
$1 \cdot c \;=\; c$			

Fig. 10. Properties of the '$\cdot$' and '$*$' operators over causal terms without '$+$'.

Absorption (der)	Transitivity (der)
$d * c \cdot d \;=\; c \cdot d$	$c \cdot d * d \cdot e \;=\; c \cdot d * d \cdot e * c \cdot e$
$d * d \cdot e \;=\; d \cdot e$	

Fig. 11. Additionally properties of the '$\cdot$' and '$*$' operators over causal terms without '$+$'.

Proposition 12 (*Application associativity*)

Let T, U and W be three causal values. Then $T \cdot (U \cdot W) = (T \cdot U) \cdot W$.

Proof. By definition, it follows that

$$(T \cdot U) \cdot W = \downarrow\{\ G_t \cdot G_u \mid G_t \in T \text{ and } G_u \in U \ \} \cdot W$$
$$= \downarrow\{\ G' \cdot G_w \mid G' \leq G_t \cdot G_u,\ G_t \in T,\ G_u \in U \text{ and } G_w \in W \ \}$$

It is clear that $G_t \cdot G_u \leq G_t \cdot G_u$ and then

$$\downarrow\{\ (G_t \cdot G_u) \cdot G_w \mid G_t \in T,\ G_u \in U \text{ and } G_w \in W \ \} \subseteq \downarrow\{\ G' \cdot G_w \mid G' \leq G_t \cdot G_u,\ G_t \in T, \dots \ \}$$

Furthermore since '·' is monotonic, for every $G' \leq G_t \cdot G_u$, it holds that $G' \cdot G_w \leq (G_t \cdot G_u) \cdot G_w$ and then

$$(T \cdot U) \cdot W = \downarrow\{\ (G_t \cdot G_u) \cdot G_w \mid G_t \in T,\ G_u \in U \text{ and } G_w \in W \ \}$$

Applying the same reasoning we can also conclude that

$$T \cdot (U \cdot W) = \downarrow\{\ G_t \cdot (G_u \cdot G_w) \mid G_t \in T,\ G_u \in U \text{ and } G_w \in W \ \}$$

That is, $T \cdot (U \cdot W) = (T \cdot U) \cdot W$ holds whether, for every causal graph G_t, G_u and G_w, it holds that $(G_t \cdot G_u) \cdot G_w = G_t \cdot (G_u \cdot G_w)$. This holds due to Proposition 2.

Proposition 13 (*Application distributivity w.r.t. additions*)

Let T, U and W be three causal values. Then, it holds that $U \cdot (T + W) = (U \cdot T) + (U \cdot W)$ and $(U + T) \cdot W = (U \cdot W) + (T \cdot W)$.

Proof. By definition, it follows that

$$(U \cdot T) + (U \cdot W) = (U \cdot T) \cup (U \cdot W)$$
$$= \downarrow\{\ G_U \cdot G_T \mid G_U \in U \text{ and } G_T \in T \ \} \cup \downarrow\{\ G_U \cdot G_T \mid G_U \in U \text{ and } G_W \in W \ \}$$
$$= \downarrow\{\ G_U \cdot G' \mid G_U \in U \text{ and } G' \in T \cup W \ \} = U \cdot (T \cup W) = U \cdot (T + W)$$

Furthermore $(U + T) \cdot W = (U \cdot W) + (T \cdot W)$ holds symmetrically.

Proposition 14 (*Application absorption*)

Let T, U and W be three causal values. Then $T = T + U \cdot T \cdot W$ and $U \cdot T \cdot W = T * U \cdot T \cdot W$

Proof. From Proposition 12 it follows that

$$U \cdot T \cdot W = \downarrow\{\ G_U \cdot G_T \cdot G_W \mid G_U \in U,\ G_T \in T \text{ and } G_W \in W \ \}$$

Furthermore, for every c-graph $G_T \in T$, it holds that $G_U \cdot G_T \cdot T_W \leq G_T$. Then, since T is an ideal, it follows that $G_U \cdot G_T \cdot T_W \in T$ and consequently $U \cdot T \cdot W \subseteq T$. Thus $U \cdot T \cdot W \cup T = T$ and $U \cdot T \cdot W \cap T = U \cdot T \cdot W$ and, by definition, these equalities can be rewritten as $U \cdot T \cdot W + T = T$ and $U \cdot T \cdot W * T = U \cdot T \cdot W$.

Proposition 15 (*Application identity and annihilator*)

Given a causal value T, it holds that $T = 1 \cdot T$, $T = T \cdot 1$, $0 = T \cdot 0$ and $0 = 0 \cdot T$.

Proof. Note that 1 and 0 respectively correspond to $\mathbf{C}_{Lb}$ and $\emptyset$ and by definition it follows that

$$1 \cdot T = \,\downarrow\!\big\{\; G \cdot G_T \;\big|\; G \in \mathbf{C}_{Lb} \text{ and } G_T \in T \;\big\} = T$$
$$0 \cdot T = \,\downarrow\!\big\{\; G \cdot G_T \;\big|\; G \in \emptyset \text{ and } G_T \in T \;\big\} = \emptyset = 0$$

The other cases are symmetric.

Corollary 4

The equivalences reflected in the Figure 2 hold.

Theorem 9

Given a set of labels *Lb*, $\langle \mathbf{V}_{Lb}, +, *, \cdot \rangle$ is the free algebra generated by *Lb* defined by equations in the Figure 4 and 2, i.e. the mapping $value : Lb \longrightarrow \mathbf{V}_{Lb}$ mapping each label l to the causal value $\downarrow\! G_l$ with G_l being the causal graph containing the only edge (l, l) is an injective (preserving-idempotence) homomorphism and for any set F and idempotence-preserving map $\delta : Lb \longrightarrow F$, there exists a homomorphism $term : \mathbf{V}_{Lb} \longrightarrow F$ defined as $term(U) \mapsto \sum \{\, term(G) \mid G \in U \,\}$ such that $\delta = term \circ graph$.

Proof. Note, from Theorems 7 and 8, that $graph : Lb \longrightarrow \mathbf{C}_{Lb}$ and $\downarrow : \mathbf{C}_{Lb} \longrightarrow \mathbf{V}_{Lb}$ are injective homomorphisms respectively from *Lb* to $\mathbf{C}_{Lb}$ and from $\mathbf{C}_{Lb}$ to $\mathbf{V}_{Lb}$. Furthermore the composition of injective homomorphisms is injective too. So that $value(l) \mapsto\, \downarrow\! graph(l)$ is an injective homomorphism between *Lb* and $\mathbf{V}_{Lb}$. We will show now that $term : \mathbf{V}_{Lb} \longrightarrow F$ preserves the '+':

$$term\Big(\sum_{U \in \mathscr{U}} U \Big) = term\Big(\bigcup \mathscr{U} \Big) = \sum_{G \subseteq \bigcup \mathscr{U}} term(G) = \sum_{U \in \mathscr{U}} \sum_{G \in U} term(G) = \sum_{U \in \mathscr{U}} term(U)$$

that *term* also preserves '$*$':

$$\prod_{U \in \mathscr{U}} term(U) \;=\; \prod_{U \in \mathscr{U}} \sum_{G \in U} term(G) = \sum_{\varphi \in \Phi} \prod_{U \in \mathscr{U}} term(\varphi(U)) = \sum_{\varphi \in \Phi} term\Big(\prod_{U \in \mathscr{U}} \varphi(U) \Big) = \sum_{t \in T_1} t$$
$$term\Big(\prod_{U \in \mathscr{U}} U \Big) = \;\; term\Big(\bigcap \mathscr{U} \Big) \;\;=\;\; \sum_{G \in \bigcap \mathscr{U}} term(G) \qquad\qquad\qquad\qquad = \sum_{t \in T_2} t$$

where $\Phi \stackrel{\text{def}}{=} \{\, \varphi \mid \varphi(U) \in U \text{ with } U \in \mathscr{U} \,\}$. Note that $G \in \bigcap \mathscr{U}$ implies that $G \in U$ for all $U \in \mathscr{U}$ and consequently there is $\varphi \in \Phi$ s.t. $\varphi(U) = G$ for all $U \in \mathscr{U}$. Thus $term\big(\prod_{U \in \mathscr{U}} \varphi(U) \big) = term\big(\prod_{U \in \mathscr{U}} G \big) = term(G)$. That is $T_2 \subseteq T_1$. Note also that $\prod_{U \in \mathscr{U}} \varphi(U) = \big(\bigcup_{U \in \mathscr{U}} U \big)^* \supseteq \varphi(U)$ for all $U \in \mathscr{U}$ and consequently $\prod_{U \in \mathscr{U}} U \in U$ for all $U \in \mathscr{U}$ and so that $\prod_{U \in \mathscr{U}} U \in \bigcap \mathscr{U}$ and

$T_2 \supseteq T_1$. And that *term* preserves '$\cdot$' too:

$$
\begin{aligned}
term(U) \cdot term(W) &= \Big(\sum_{G_u \in U} term(G_u) \Big) \cdot \Big(\sum_{G_w \in W} term(G_w) \Big) \\
&= \sum_{G_u \in U,\ G_w \in W} term(G_u) \cdot term(G_w) \\
&= \sum_{G_u \in U,\ G_w \in W} term(G_u \cdot G_w) \\
&= term\Big(\sum_{G_u \in U,\ G_w \in W} G_u \cdot G_w \Big) \\
&= term\Big(\sum_{G_u \in U} G_u \cdot \sum_{G_w \in W} G_w \Big) \\
&= term(U \cdot W)
\end{aligned}
$$

Finally $term(value(l)) = \sum_{G \in value(l)} term(G) = term(G_l)$ and, from Theorem 7, $term(G_l) = l$.

7.3 Positive programs

Lemma 7.1
Let P be a positive (and possible infinite) logic program over signature $\langle At, Lb \rangle$. Then, (*i*) the least fix point of T_P, $lfp(T_P)$ is the least model of P, and (*ii*) $lfp(T_P) = T_P\uparrow^{\omega}(\mathbf{0})$. □

Proof. Since the set of causal values forms a lattice causal logic programs can be translated to *Generalized Annotated Logic Programming* (GAP). GAP is a general a framework for multivalued logic programming where the set of truth values must to form an upper semilattice and rules (*annotated clauses*) have the following form:

$$H : \rho \leftarrow B_1 : \mu_1 \ \& \ \ldots \ \& \ B_n : \mu_n \tag{9}$$

where $L_0, \ldots, L_m$ are literals, ρ is an *annotation* (may be just a truth value, an *annotation variable* or a *complex annotation*) and $\mu_1, \ldots, \mu_n$ are values or annotation variables. A complex annotation is the result to apply a total continuous function to a tuple of annotations. Thus a positive program P is encoded in a GAP program, $GAP(P)$ rewriting each rule $R \in \Pi$ of the form

$$t : H \leftarrow B_1 \wedge \ldots \wedge B_n \tag{10}$$

as a rule $GAP(R)$ in the form (9) where $\mu_1, \ldots, \mu_n$ are annotation that capture the causal values of each body literal and ρ is a complex annotation defined as $\rho = (\mu_1 * \ldots * \mu_n) \cdot t$.

Thus we will show that a causal interpretation $I \models \Pi$ if and only if $I \models^r GAP(P)$ where $\models^r$ refers to the GAP restricted semantics.

For any program P and interpretation I, by definition, $I \models P$ (resp. $I \models^r GAP(P)$) iff $I \models R$ (resp. $I \models^r GAP(R)$) for every rule $R \in P$. Thus it is enough to show that for every rule R it holds that $I \models R$ iff $I \models^r GAP(R)$.

By definition, for any rule R of the form of (10) and an interpretation I, $I \models R$ if and only if $\big(I(B_1) * \ldots * I(B_n)\big) \cdot t \leq I(H)$ whereas for any rule $GAP(R)$ in the form of (9), $I \models^r GAP(R)$ iff for all $\mu_i \leq I(B_i)$ implies that $\rho = (\mu_1 * \ldots * \mu_n) \cdot t \leq I(H)$.

For the only if direction, take $\mu_i = I(B_i)$, then $\rho = (\mu_1 * \ldots * \mu_n) \cdot t = (I(B_1) * \ldots * I(B_n)) \cdot t$ and then $\rho \leq I(H)$ implies $\big(I(B_1) * \ldots * I(B_n)\big) \cdot t \leq I(H)$, i.e. $I \models^r GAP(R)$ implies $I \models R$. For the if direction, take $\mu_i \leq I(B_i)$ then, since product an applications are monotonic operations, it follows that $(\mu_1 * \ldots * \mu_n) \cdot t \leq (I(B_1) * \ldots * I(B_n)) \cdot t \leq I(H)$, That is, $I \models R$ also implies $I \models^r GAP(R)$. Consequently $I \models R$ iff $I \models^r GAP(R)$.

Thus, from Theorem 1 in (Kifer and Subrahmanian 1992), it follows that the operator T_P is monotonic.

To show that the operator T_P is also continuous we need to show that for every causal program P the translation $GAP(P)$ is an *acceptable* program. Indeed since in a program $GAP(P)$ all body atoms are v-annotated it is *acceptable*. Thus from Theorem 3 in (Kifer and Subrahmanian 1992), it follows that $T_P\uparrow^{\omega}(\mathbf{0}) = lfp(T_P)$ and this is the least model of P.

Lemma 7.2
Given a positive and completely labelled program P, for every atom p and integer $k \geq 1$,

$$T_P\uparrow^{k}(\mathbf{0})(p) = \sum_{R \in \Psi} \sum_{f \in R} \prod \big\{ f\big(T_P\uparrow^{k-1}(\mathbf{0})(q)\big) \mid q \in body(R) \big\} \cdot label(R)$$

where Ψ is the set of rules $\Psi = \{\, R \in \Pi \mid head(R) = p \,\}$ and R is the set of choice functions $R = \{\, f \mid f(S) \in S \,\}$.

Proof. By definition of $T_P \uparrow^k (\mathbf{0})(p)$ it follows that

$$T_P \uparrow^k (\mathbf{0})(p) = \sum \{\, \big(T_P \uparrow^{k-1} (\mathbf{0})(q_1) * \ldots * T_P \uparrow^{k-1} (\mathbf{0})(q_1)\big) \cdot label(R) \mid R \in P \text{ with } head(R) = p \,\}$$

then, applying distributive of application w.r.t. to the sum and and rewriting the sum and the product aggregating properly, it follows that

$$T_P \uparrow^k (\mathbf{0})(p) = \sum_{R \in \Psi} \prod \{\, T_P \uparrow^{k-1} (\mathbf{0})(q) \mid q \in body(R) \,\} \cdot label(R)$$

Furthermore for any atom q the causal value $T_P \uparrow^{k-1} (\mathbf{0})(q)$ can be expressed as the sum of all c-graphs in it and then

$$T_P \uparrow^k (\mathbf{0})(p) = \sum_{R \in \Psi} \prod \{\, \sum_{f \in R} f\big(T_P \uparrow^{k-1} (\mathbf{0})(q)\big) \mid q \in body(R) \,\} \cdot abel(R)$$

and applying distributivity of products over sums it follows that

$$T_P \uparrow^k (\mathbf{0})(p) = \sum_{R \in \Psi} \sum_{f \in R} \prod \{\, f\big(T_P \uparrow^{k-1} (\mathbf{0})(q)\big) \mid q \in body(R) \,\} \cdot l_R \qquad \square$$

Lemma 7.3
Given a positive and completely labelled program P and a causal graph G, for every atom p and integer $k \geq 1$, it holds that $G \in T_P \uparrow^k (\mathbf{0})(p)$ iff there is a rule $l : p \leftarrow q_1, \ldots, q_m$ and causal graphs $G_{q_1}, \ldots, G_{q_m}$ respectively in $T_P \uparrow^{k-1} (\mathbf{0})(q_i)$ and $G \leq \big(G_{q_1} * \ldots * G_{q_m}\big) \cdot l$.

Proof. From Lemma 7.2 it follows that $G \in T_P \uparrow^k (\mathbf{0})(p)$ iff

$$G \in value\Big(\sum_{R \in \Psi} \sum_{f \in R} \prod \{\, f\big(T_P \uparrow^{k-1} (\mathbf{0})(q)\big) \mid q \in body(R) \,\} \cdot label(R) \Big)$$

iff
$$G \in \bigcup_{R \in \Psi} \bigcup_{f \in R} value\Big(\prod \{\, \downarrow f\big(T_P \uparrow^{k-1} (\mathbf{0})(q)\big)\big) \mid q \in body(R) \,\} \cdot label(R)$$

iff there is $R \in \Phi$, with $head(R) = p$ and a choice function $f \in \Psi$ s.t.

$$G \in value\Big(\prod \{\, f\big(T_P \uparrow^{k-1} (\mathbf{0})(q)\big) \mid q \in body(R) \,\} \cdot label(R) \Big)$$

Let $R = l : p \leftarrow q_1, \ldots, q_m$ and $f\big(T_P \uparrow^{k-1} (\mathbf{0})(q_i)\big) = G_{q_i}$. Then the above can be rewritten as $G \leq \big(G_{q_1} * \ldots * G_{q_m}\big) \cdot l$. $\qquad \square$

Definition 10
Given a causal graph $G = \langle V, E \rangle$, we define the *restriction* of G to a set of vertex $V' \subseteq V$ as the casual graph $G' = \langle V', E' \rangle$ where $E' = \{\, (l_1, l_2) \in E \mid l_1 \in V' \text{ and } l_2 \in V' \,\}$, and we define the *reachable restriction* of G to a set of vertex $V' \subseteq V$, in symbols $G^{V'}$, as the restriction of G to the set of vertex V'' from where some vertex $l \in V'$ is reachable $V'' = \{\, l' \in V \mid (l', l) \in E^* \text{ for some } l \in V' \,\}$. When $V' = l$ is a singleton we write G^l.

Lemma 7.4
Let P be a positive, completely labelled program, p and q be atoms, G be a causal graph, R be a causal rule s.t. $head(R) = q$ and $label(R) = l$ is a vertex in G and $k \in \{1, \ldots, \omega\}$ be an ordinal. If $G \in T_P \uparrow^k (\mathbf{0})(p)$, then $G^l \in T_P \uparrow^k (\mathbf{0})(q)$. □

Proof. In case that $k = 0$ the lemma statement holds vacuous. Otherwise assume as induction hypothesis that the lemma statement holds for $k - 1$. From Lemma 7.3, since $G \in T_P \uparrow^k (\mathbf{0})(p)$, there is a rule $R_p = (l_p : p \leftarrow p_1, \ldots, p_m)$ and c-graph $G_{p_1}, \ldots, G_{p_m}$ s.t. each $G_{p_i} \in T_P \uparrow^{k-1} (\mathbf{0})(p_i)$ and $G \leq (G_{p_1} * \ldots * G_{p_m}) \cdot l_p$.

If $l = l_p$ then, since P is uniquely labelled, $R = R_p$, $G^l = G$ and by assumption $G \in T_P \uparrow^k (\mathbf{0})(p)$. Otherwise $l \in G_{p_i}$ for some G_{p_i} and in its turn $G_{p_i} \in T_P \uparrow^{k-1} (\mathbf{0})(p_i)$. By induction hypothesis $G^l \in T_P \uparrow^{k-1} (\mathbf{0})(q)$ and since $T_P \uparrow^{k-1} (\mathbf{0})(q) \subseteq T_P \uparrow^k (\mathbf{0})(q)$ it follows that $G^l \in T_P \uparrow^k (\mathbf{0})(q)$.

In case that $k = \omega$, by definition $T_P \uparrow^\omega (\mathbf{0})(p) = \sum_{i < \omega} T_P \uparrow^i (\mathbf{0})(p)$ and the same for atom q. Thus, if $G \in T_P \uparrow^\omega (\mathbf{0})(p)$ there is some $i < \omega$ s.t. $G \in T_P \uparrow^i (\mathbf{0})(p)$, and as we already show, $G^l \in T_P \uparrow^i (\mathbf{0})(q)$ and consequently $G^l \in T_P \uparrow^\omega (\mathbf{0})(q)$. □

Lemma 7.5
Let P be a positive, completely labelled program, p be an atom and G be a causal graph and $k \geq 1$ be an integer. If G is maximal in $T_P \uparrow^k (\mathbf{0})(p)$ then

1. there is a causal rule $R = (l : p \leftarrow p_1, \ldots, p_m)$ and there are causal graphs $G_{p_1}, \ldots, G_{p_m}$ s.t. each $G_{p_i} \in \max T_P \uparrow^{k-1} (\mathbf{0})(p_i)$ and $G_p = (G_{p_1} * \ldots * G_{p_m}) \cdot l$ and
2. l is not a vertex of any G_{p_i}. □

Proof. From Lemma 7.3 it follows that $G \in T_P \uparrow^k (\mathbf{0})(p)$ iff there is a rule $R = (l : p \leftarrow q_1, \ldots, q_m)$ and causal graphs $G'_{q_1}, \ldots, G'_{q_m}$ s.t. each $G_{p_i} \in \max T_P \uparrow^{k-1} (\mathbf{0})(p_i)$ and $G = (G'_{q_1} * \ldots * G'_{q_m}) \cdot l$. Let $G_{q_1}, \ldots, G_{q_m}$ be causes such that each $G_{q_i} \in \max T_P \uparrow^{k-1} (\mathbf{0})(q_i)$ and $G'_{q_i} \leq G_{q_i}$ and let G' be the c-graph $G' = (G_{q_1} * \ldots * G_{q_m}) \cdot l$. By product and application monotonicity it holds that $G \leq G'$ and, again from Lemma 7.3, it follows that $G' \in T_P \uparrow^k (\mathbf{0})(p)$. Thus, since G is maximal, it must to be that $G = G'$ and consequently $G = (G_{q_1} * \ldots * G_{q_m}) \cdot l$ where each G_{q_i} is maximal.

Suppose that l is a vertex of G_{p_i} for some G_{p_i}. From Lemma 7.4, if follows that $G_{p_i}^l \in T_P \uparrow^k (\mathbf{0})(p)$. Furthermore, since $G_{p_i} \supseteq G_{p_i}^l$, it follows that $G_{p_i} \leq G_{p_i}^l$ and, since l is a label ($l \neq 1$), it follows that $G < G_{p_i}$ and so that $G < G_{p_i}^l$ which contradicts the assumption that $G \in \max T_P \uparrow^k (\mathbf{0})(p)$. □

Definition 11
Given a causal graph G we define *height*(G) as the length of the maximal simple (no repeated vertices) path in G.

Lemma 7.6
Let P be a positive, completely labelled program, p be an atom, $k \in \{1, \ldots, \omega\}$ be an ordinal and G be a causal graph. If $G \in \max T_P \uparrow^k (\mathbf{0})(p)$ and *height*$(G) = h \leq k$ then $G \in T_P \uparrow^h (\mathbf{0})(p)$.

Proof. In case that $h = 0$, from Lemma 7.5, it follows that if $G \in \max T_P \uparrow^k (\mathbf{0})(p)$ there is a causal rule $R = (l : p \leftarrow p_1, \ldots, p_m)$ and c-graphs.... Furthermore, since P is completely labelled, it follows that $l \neq 1$ and then $G < l < 1$. Since 1 is the only c-graph whose *height* is 0 the lemma statement holds vacuous.

In case that $h > 0$, we proceed by induction assuming as hypothesis that the lemma statement holds for any $h' < h$. From Lemma 7.5, there is a causal rule $l : p \leftarrow p_1, \ldots, p_m$, and there are causal graphs $G_{p_1}, \ldots, G_{p_m}$ s.t. each $G_{p_i} \in \max T_P \uparrow^{k-1} (\mathbf{0})(p_i)$, $G = G_R \cdot l$ and $l \notin V(G_{p_i})$ for any G_{p_i} where $G_R = G_{p_1} * \ldots * G_{p_m}$.

If $m = 0$ then $G = 1 \cdot l = l$, $height(l) = 1$ and $l \in \max T_P \uparrow^k (\mathbf{0})(p)$ for any $k \geq 1$. Otherwise, since any path in G_{p_i} is also a path G_p, it is clear that $height(G_{p_i}) = h'_{p_i} \leq h$ for any G_{p_i}. Suppose that $h'_{p_i} = h$ for some G_{p_i}. Then there is a simple path $l_1, \ldots, l_h$ of length h in G_{p_i} and, since $G = G_R \cdot l$, there is an edge $(l_h, l) \in E(G)$. That is $l_1, \ldots, l_h, l$ is a walk of length $h + 1$ in G and, since $l \notin V(G_{p_i})$, it follows that $l_i \neq l$ with $1 \leq i \leq h$. So that $l_1, \ldots, l_h, l$ is a simple path of length $h + 1$ which contradicts the assumption that $height(G) = h$. Thus $height(G_{p_i}) = h'_{p_i} < h$ for any G_{p_i} and then, by induction hypothesis, $G_{p_i} \in \max T_P \uparrow^{h'_{p_i}} (\mathbf{0})(p_i)$.

Let $h' = \max\{ h'_{p_i} \mid 1 \leq i \leq m \} < h$. Since the T_P operator is monotonic and $h'_{p_i} \leq h'$ for any p_i, it follows that $T_P \uparrow^{h'_{p_i}} (\mathbf{0})(p_i) \leq T_P \uparrow^{h'} (\mathbf{0})(p_i)$ and then there are casual graphs $G'_{p_1}, \ldots, G'_{p_m}$ such that each $G'_{p_i} \in \max T_P \uparrow^{h'} (\mathbf{0})(p_i)$, $G_{p_i} \leq G'_{p_i}$ and $G' = G_R \cdot l$ where $G'_R = G'_{p_1} * \ldots * G'_{p_m}$. By product and application monotonicity, it follows that $G \leq G'$, and, from Lemma 7.3, it follows that $G' \in T_P \uparrow^{h'+1} (\mathbf{0})(p)$. Since $h' + 1 \leq h$ it follows that $G' \in T_P \uparrow^h (\mathbf{0})(p)$ and since $G \leq G'$ it follows that $G \in T_P \uparrow^h (\mathbf{0})(p)$.

Suppose that $G \notin \max T_P \uparrow^h (\mathbf{0})(p)$. Then there is $G'' \in \max T_P \uparrow^h (\mathbf{0})(p)$ s.t. $G < G''$ and then, since $h \leq k$, it follows that $G'' \in T_P \uparrow^k (\mathbf{0})(p)$ which, since $G < G''$, contradicts the assumption that $G \in \max T_P \uparrow^k (\mathbf{0})(p)$. Thus, if $G \in \max T_P \uparrow^k (\mathbf{0})(p)$ and $height(G) = h \leq k$ it follows that $G \in T_P \uparrow^h (\mathbf{0})(p)$.

In case that $k = \omega$, by definition $T_P \uparrow^\omega (\mathbf{0})(p) = \sum_{i < \omega} T_P \uparrow^i (\mathbf{0})(p)$. Thus, if $G \in \max T_P \uparrow^\omega (\mathbf{0})(p)$ and $height(G) = h$ then there is some $i < \omega$ s.t. $G \in \max T_P \uparrow^i (\mathbf{0})(p)$ and $h \leq i$, and as we already show, then $G \in T_P \uparrow^h (\mathbf{0})(p)$. $\square$

Lemma 7.7

Let P, Q two positive causal logic programs such that Q is the result of replacing label l in P by some u (a label or 1) then $T_Q \uparrow^k (\mathbf{0})(p) = T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u]$ for any atom p and $k \in \{1, \ldots, \omega\}$.

Proof. In case that $n = 0$, $T_Q \uparrow^k (\mathbf{0})(p) = 0$ and $T_P \uparrow^k (\mathbf{0})(p) = 0$ and $0 = 0[l \mapsto u]$. That is $T_Q \uparrow^k (\mathbf{0})(p) = T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u]$.

We proceed by induction on k assuming that $T_Q \uparrow^{k-1} (\mathbf{0})(p) = T_P \uparrow^{k-1} (\mathbf{0})(p)[l \mapsto u]$ for any atom p and we will show that $T_Q \uparrow (\mathbf{0})(p) = T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u]$.

Pick $G \in T_P \uparrow^k (\mathbf{0})(p)$ then, from Lemma 7.3, there is a rule $l' : p \leftarrow q_1, \ldots, q_m$ and causal graphs $G_{q_1}, \ldots, G_{q_m}$ each one respectively in $T_P \uparrow^{k-1} (\mathbf{0})(q_i)$ s.t. $G \leq G_R = (G_{q_1} * \ldots * G_{q_m}) \cdot l'$. Thus, by induction hypothesis, for every atom q_i and c-graph $G_{q_i} \in T_P \uparrow^{k-1} (\mathbf{0})(q)$ it holds that $G_{q_i}[l \mapsto u] \in T_Q \uparrow^{k-1} (\mathbf{0})(q_i)$.

Let $G_R[l \mapsto u]$ be a c-graph defined as $G_R[l \mapsto u] = \big(G_{q_1}[l \mapsto u] * \ldots * G_{q_m}[l \mapsto u]\big) \cdot l'[l \mapsto u]$. Then, since $G \leq G_R$, it follows that $G[l \mapsto u] \leq G_R[l \mapsto u]$ and then, again from Lemma 7.3, it follows that $G[l \mapsto u] \in T_Q \uparrow^n (\mathbf{0})(p)$. That is $T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u] \subseteq T_Q \uparrow^k (\mathbf{0})(p)$.

Pick $G \in T_Q \uparrow^k (\mathbf{0})(p)$ then, from Lemma 7.3, there is a rule there is a rule $l' : p \leftarrow q_1, \ldots, q_m$ and c-graphs $G_{q_1}, \ldots, G_{q_m}$ respectively in $T_P \uparrow^{k-1} (\mathbf{0})(q_i)$ s.t. $G \leq G_R$ where $G_R = (G_{q_1} * \ldots * G_{q_m}) \cdot l'$.

By induction hypothesis, for every atom q_i and graph G_{q_i} it holds that if $G_{q_i} \in T_Q \uparrow^{k-1} (\mathbf{0})(q_i)$ then $G_{q_i} \in T_P \uparrow^{k-1} (\mathbf{0})(q_i)[l \mapsto u]$. Thus, it follows that there is a graph $G'_{q_i} \in T_P \uparrow^{k-1} (\mathbf{0})(q_i)$ such that $G'_{q_i}[l \mapsto u] = G_{q_i}$. Let G'_R be a graph s.t. $G'_R = (G'_{q_1} * \ldots * G'_{q_m}) \cdot l'$. From Lemma 7.3 for every causal graph $G' \leq G'_R$ it holds that $G' \in T_P \uparrow^k (\mathbf{0})(p)$. Since $G'_R[l \mapsto u] = G_R$ and $G \leq G_R$ it follows that $G \leq G_R[l \mapsto u]$ and, since $G_R \in T_P \uparrow^k (\mathbf{0})(p)$, it follows that $G \in T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u]$. Consequently $T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u] \supseteq T_Q \uparrow^n (\mathbf{0})(p)$ and then $T_P \uparrow^k (\mathbf{0})(p)[l \mapsto u] = T_Q \uparrow^n (\mathbf{0})(p)$.

In case that $k = \omega$, by definition $T_P \uparrow^\omega (\mathbf{0})(p)[l \mapsto u] = \sum_{i < \omega} T_P \uparrow (\mathbf{0})(i)p[l \mapsto u]$ and as we aleready show $T_P \uparrow (\mathbf{0})(i)p[l \mapsto u] = T_Q \uparrow^i (\mathbf{0})(p)$ for all integer $i < \omega$, so that, their sum is also equal and consequently $T_P \uparrow^\omega (\mathbf{0})(p)[l \mapsto u] = T_Q \uparrow^\omega (\mathbf{0})(p)$.

Proof of Theorem 2. Let P' be a positive, completely labelled causal program with the same rules as P. From Lemma 7.1 it follows that (*i*) $lfp(T_P)$ and $lfp(T_{P'})$ are respectively the least model of the programs P and P', and (*ii*) $lfp(T_P) = T_P \uparrow^\omega (\mathbf{0})$ and $lfp(T_{P'}) = T_{P'} \uparrow^\omega (\mathbf{0})$.

Futhermore, it is clear that if P is an infinite program, i.e. $n = \omega$, then $T_{P'} \uparrow^n (\mathbf{0}) = T_{P'} \uparrow^\omega (\mathbf{0})$. Otherwise, by definition it holds that $T_{P'} \uparrow^n (\mathbf{0}) \leq T_{P'} \uparrow^\omega (\mathbf{0})$. Suppose $T_{P'} \uparrow^n (\mathbf{0}) < T_{P'} \uparrow^\omega (\mathbf{0})$. Then there is some atom p and c-graph $G \in T_{P'} \uparrow^\omega (\mathbf{0})(p)$ such that $G \notin T_{P'} \uparrow^n (\mathbf{0})(p)$. The longest simple path in G must be smaller than the number of its vertices and this must be smaller than the number of labels of the program which in its turn is equal to the number of rules n, i.e. $height(G) = h \leq n$. From Lemma 7.6 it follows that $G \in T_{P'} \uparrow^h (\mathbf{0})(p)$ and since $h \leq n$ it follows that $T_{P'} \uparrow^h (\mathbf{0})(p) \subseteq T_{P'} \uparrow^n (\mathbf{0})(p)$ and so that $G \in T_{P'} \uparrow^n (\mathbf{0})(p)$ which is a contradiction with the assumption that $G \in T_{P'} \uparrow^\omega (\mathbf{0})(p)$ but $G \notin T_{P'} \uparrow^n (\mathbf{0})(p)$. Thus $T_{P'} \uparrow^n (\mathbf{0}) = T_{P'} \uparrow^\omega (\mathbf{0})$.

Furthermore, from Lemma 7.7, $T_P \uparrow^k (\mathbf{0})(p) = T_{P'} \uparrow^k (\mathbf{0})(p)[l'_1 \mapsto l_1] \ldots [l'_n \mapsto l_n]$ for $k \in \{n, \omega\}$ and where $l'_1, \ldots, l'_n$ are the labels of rules of P' and $l_1, \ldots, l_n$ are the correspondent labels of such rules in P. Thus, since $T_{P'} \uparrow^n (\mathbf{0}) = T_{P'} \uparrow^\omega (\mathbf{0})$, it follows that $T_P \uparrow^n (\mathbf{0}) = T_P \uparrow^\omega (\mathbf{0})$.

Lemma 7.8
For any proof $\pi(p)$ it holds that

$$graph\left(\frac{\pi(q_1), \ldots, \pi(q_m)}{p} (l)\right) = \left(graph(\pi(q_1)) * \ldots * graph(\pi(q_1))\right) \cdot l$$

Proof. We proceed by structural induction assuming that for every proof in the antecedent $\pi(q_i)$ and every label $l' \in V(graph(\pi(q_i)))$ there is an edge $(l', label(\pi(q_i))) \in E(graph(\pi(q_i)))$.

By definition $graph(\pi(p)) = G^*_{\pi(p)}$ is the reflexive and transitive closure of $G_{\pi(p)}$ and then

$$graph(\pi(p)) = \left(\bigcup \{ graph(\pi(q_i)) \mid 1 \leq i \leq m \} \cup \{ (label(\pi(q_i), l) \mid 1 \leq i \leq m \}\right)^*$$

Thus, $graph(\pi(p)) \geq \prod \{ graph(\pi(q_i)) \mid 1 \leq i \leq m \} \cdot l$ and remain to show that for every atom q_i and label $l' \in V(graph(\pi(q_i)))$ the edge $(l', l) \in E(graph(\pi(p)))$. Indeed, since by induction hypothesis there is an edge $(l', label(\pi(q_i))) \in E(graph(\pi(q_i))) \subseteq E(graph(\pi(p)))$, the fact that the edge $(label(\pi(q_i), l) \in E(graph(\pi(p)))$ and since $graph(\pi(p))$ is closed transitively, it follows that $(l', l) \in E(graph(\pi(p)))$. $\qquad\square$

Lemma 7.9

Let P be a positive, completely labelled program and $\pi(p)$ be a proof for p w.r.t. P. Then it holds that $graph(\pi_p) \in T_P \uparrow^h (\mathbf{0})(p)$ where h is the height of $\pi(p)$ which is recursively defined as

$$height(\pi) = 1 + \max\{\, height(\pi') \mid \pi' \text{ is a sub-proof of } \pi \,\}$$

Proof. In case that $h = 1$ the antecedent of $\pi(p)$ is empty, i.e.

$$\pi(p) = \frac{\top}{p}\,(l)$$

where l is the label of the fact $(l : p)$. Then $graph(\pi(p)) = l$. Furthermore, since the fact $(l : p)$ is in the program P, it follows that $l \in T_P \uparrow^1 (\mathbf{0})(p)$.

In the remain cases, we proceed by structural induction assuming that for every natural number $h \leq n - 1$, atom p and proof $\pi(p)$ of p w.r.t. P whose $height(\pi(p)) = h$ it holds that $graph(\pi(p)) \in T_P \uparrow^h (\mathbf{0})(p)$ and we will show it in case that $h = n$.

Since $height(\pi_p) > 1$ it has a non empty antecedent, i.e.

$$\pi(p) = \frac{\pi(q_1), \ldots, \pi(q_m)}{p}\,(l)$$

where l is the label of the rule $l : p \leftarrow q_1, \ldots, q_m$. By *height* definition, for each q_i it holds that $height(\pi(q_i)) \leq n - 1$ and so that, by induction hypothesis, $graph(\pi(q_i)) \in T_P \uparrow^{h-1} (\mathbf{0})(q_i)$. Thus, from Lemmas 7.3 and 7.8, it follows respectively that

$$\prod\{\, graph(\pi(q_i)) \mid 1 \leq i \leq m \,\} \cdot l \in T_P \uparrow^h (\mathbf{0})(p)$$
$$graph(\pi(p)) = \prod\{\, graph(\pi(q_i)) \mid 1 \leq i \leq m \,\} \cdot l$$

That is, $graph(\pi(p)) \in T_P \uparrow^h (\mathbf{0})(p)$. $\qquad\square$

Lemma 7.10

Let P be a positive, completely labelled program and $\pi(p)$ be a proof of p w.r.t. P. For every atom p and maximal causal graph $G \in T_P \uparrow^\omega (\mathbf{0})(p)$ there is a non-redundant proof $\pi(p)$ for p w.r.t. P s.t. $graph(\pi(p)) = G$.

Proof. From Lemma 7.5 for any maximal graph $G \in T_P \uparrow^k (\mathbf{0})(p)$, there is a rule $l : p \leftarrow q_1, \ldots, q_m$ and maximal graphs $G_{q_1} \in T_P \uparrow^{h-1} (\mathbf{0})(q_1), \ldots, G_{q_m} \in T_P \uparrow^{k-1} (\mathbf{0})(q_m)$ s.t.

$$G = (G_{q_1} * \ldots * G_{q_m}) \cdot l$$

Furthermore, we assume as induction hypothesis that for every atom q_i there is a non redundant proof $\pi(q_i)$ for q_i w.r.t. P s.t. $graph(\pi(q_i)) = G_{q_i}$. Then $\pi(p)$ defined as

$$\pi(p) = \frac{\pi(q_1), \ldots, \pi(q_m)}{p}\,(l)$$

is a proof for p w.r.t. P which holds $graph(\pi(p)) = G_p$ (from Lemma 7.8) and $height(\pi(p) \leq h$. Furthermore, suppose that $\pi(p)$ is redundant, i.e. there is a proog π' for p w.r.t P such that $graph(\pi(p)) < graph(\pi')$. Let $h = height(\pi')$. Then, from Lemma 7.9, it follows that $graph(\pi') \in T_P \uparrow^h (\mathbf{0})(p)$ and then $graph(\pi') \in T_P \uparrow^\omega (\mathbf{0})(p)$ which contradicts the hypothesis that G is maximal in $T_P \uparrow^\omega (\mathbf{0})(p)$. $\qquad\square$

Proof of Theorem 3. From Theorem 2 it follows that the least model I is equal to $T_P \uparrow^{\omega} (\mathbf{0})$. For the only if direction, from Lemma 7.10, it follows that for every maximal c-graph $G \in I(p) = T_P \uparrow^{\omega} (\mathbf{0})(p)$ there is a non-redundant proof $\pi(p)$ for p w.r.t P s.t. $G = graph(\pi(p))$. That is, $\pi(p) \in \Pi_p$ and then $G = graph(\pi(p)) \in graph(\Pi_p)$. For the if direction, from Lemma 7.9, for every $G \in graph(\Pi_p)$, i.e. $G = graph(\pi(p))$ for some non-redundant proof $\pi(p)$ for p w.r.t. P, it holds that $G \in T_P \uparrow^{\omega} (\mathbf{0})(p)$ and so that $G \in I(p)$. Furthermore, suppose that G is not maximal, i.e. there is a maximal c-graph $G' \in I(p)$ s.t. $G < G'$ and a proof π' for p w.r.t. P s.t. $graph(\pi') = G'$ which contradicts that $\pi(p)$ is non-redundant. □

Lemma 7.11
Let t be a causal term. Then $value(t[l \mapsto u]) = value(t)[l \mapsto u]$.

Proof. We proceed by structural induction. In case that t is a label. If $t = l$ then $value(l[l \mapsto u]) = value(u) = \downarrow u = value(l)[l \mapsto u]$. If $t = l' \neq l$ then $value(l'[l \mapsto u] = value(l') = \downarrow l' = value(l')[l \mapsto u]$. In case that $t = \prod T$ it follows that $value(\prod T[l \mapsto u]) = \bigcap\{ value(t'[l \mapsto u]) \mid t' \in T \})$ and by induction hypothesis $value(t'[l \mapsto u]) = value(t')[l \mapsto u]$. Then $value(\prod T[l \mapsto u]) = \bigcap\{ value(t')[l \mapsto u] \mid t' \in T \}) = value(\prod T)[l \mapsto u]$. The cases for $t = \sum T$ is analogous. In case that $t = t_1 \cdot t_2$ it follows that $value(t[l \mapsto u]) = value(t_1[l \mapsto u]) \cdot value(t_2[l \mapsto u]) = value(t_1)[l \mapsto u] \cdot value(t_2)[l \mapsto u] = value(t)[l \mapsto u]$

Proof of Theorem 4. From Theorem 2, models I and I' are respectively equal to $T_P \uparrow^{\omega} (\mathbf{0})$ and $T_{P'} \uparrow^{\omega} (\mathbf{0})$. Furthermore, from Lemma 7.7, it follows that $T_{P'} \uparrow^{\omega} (\mathbf{0})(p) = T_P \uparrow^{\omega} (\mathbf{0})(p)[l \mapsto u]$ for any atom p. Lemma 7.11 shows that the replacing can be done in any causal term without operate it.

Proof of Theorem 5. It is clear that if every rule in P is unlabelled, i.e. $P = P'$, then their least model assigns 0 to every *false* atom and 1 to every *true* atom, so that their least models coincide with the classical one, i.e. $I = I'$ and then $I^{cl} = I = I'$. Otherwise, let P_n be a program where n rules are labelled. We build a program P_{n-1} removing one label l and, from Theorem 4, it follows that $I_{n-1} = I_n[l \to 1]$. By induction hypothesis the corresponding classical interpretation of least model of P_{n-1} coincides with the least model of the unlabelled program, i.e. $I_{n-1}^{cl} = I'$, and then $I_n[l \mapsto 1]^{cl} = I_{n-1}^{cl} = I'$. Furthermore, for every atom p and c-graph G it holds that $G \in I_n(p)$ iff $G[l \mapsto 1] \in I_n[l \mapsto 1](p)$. Simple remain to note that $value(z) = \emptyset$, so that $I_n(p) = 0$ iff $I_n[l \mapsto 1](p) = 0$ and consequently $I_n^{cl} = I_n[l \mapsto 1]^{cl} = I'$. □

Proof of Theorem 6. By definition I and I^{cl} assigns 0 to the same atoms, so that $P^I = P^{I^{cl}}$. Furthermore let Q (instead of P' for clarity) be the unlabelled version of P. Then $Q^{I^{cl}}$ is the unlabelled version of P^I. (1) Let I be a stable model of P and J be the least model of $Q^{I^{cl}}$. Then, I is the least model of P^I and, from Theorem 5, it follows that $I^{cl} = J$, i.e. I^{cl} is a stable model of Q. (2) Let I' is a stable model of Q and I be the least model of $P^{I'}$. Since I' is a stable model of Q, by definition it is the least model of $Q^{I'}$, furthermore, since $Q^{I'}$ is the unlabelled version of $P^{I'}$ it follows, from Theorem 5, that $I^{cl} = I'$. Note that $P^I = P^{I^{cl}} = P^{I'}$. Thus I is a stable model of P. □

8 Algebraic completeness

We will show that causal terms with the algebraic properties reflected in Figures 12 and 13 are correct and complete with respect to the algebra of causal values.

Associativity			*Commutativity*			*Absorption*		
$t + (u{+}w)$	$=$	$(t{+}u) + w$	$t + u$	$=$	$u + t$	t	$=$	$t + (t * u)$
$t * (u * w)$	$=$	$(t * u) * w$	$t * u$	$=$	$u * t$	t	$=$	$t * (t{+}u)$

Distributive			*Identity*			*Idempotence*			*Annihilator*		
$t + (u * w)$	$=$	$(t{+}u) * (t{+}w)$	t	$=$	$t + 0$	t	$=$	$t + t$	1	$=$	$1 + t$
$t * (u{+}w)$	$=$	$(t * u) + (t * w)$	t	$=$	$t * 1$	t	$=$	$t * t$	0	$=$	$0 * t$

Fig. 12. Sum and product satisfy the properties of a completely distributive lattice.

Absorption			*Associativity*			*Identity*			*Annihilator*		
t	$=$	$t + u \cdot t \cdot w$	$t \cdot (u \cdot w)$	$=$	$(t \cdot u) \cdot w$	t	$=$	$1 \cdot t$	0	$=$	$t \cdot 0$
$u \cdot t \cdot w$	$=$	$t * u \cdot t \cdot w$				t	$=$	$t \cdot 1$	0	$=$	$0 \cdot t$

Indempotence	*Addition distributivity*			*Product distributivity*		
$l \cdot l = l$	$t \cdot (u{+}w)$	$=$	$(t \cdot u) + (t \cdot w)$	$c \cdot d \cdot e$	$=$	$(c \cdot d) * (d \cdot e)$ with $d \neq 1$
	$(t + u) \cdot w$	$=$	$(t \cdot w) + (u \cdot w)$	$c \cdot (d * e)$	$=$	$(c \cdot d) * (c \cdot e)$
				$(c * d) \cdot e$	$=$	$(c \cdot e) * (d \cdot e)$

Fig. 13. Properties of the '$\cdot$' operator (c, d, e are terms without '$+$' and l is a label).

First note that the algebraic properties for "$*$" and "$+$" reflected Figure 12 are the common algebraic properties satisfy by distributive lattices. Then their correctness follows directly from Theorem 1.

Proposition 16 (*Application homomorphism*)
The mapping $\downarrow\colon \mathbf{C}_{Lb}, \longrightarrow \mathbf{V}_{Lb}$ is an homomorphism between $\langle \mathbf{C}_{Lb}, \cdot \rangle$ and $\langle \mathbf{V}_{Lb}, \cdot \rangle$,

Proof. Take any c-graphs G_1 and G_2, then

$$\downarrow G_1 \cdot \downarrow G_2 = \downarrow \{ (G_1' \cdot G_2') \mid G_1' \in \downarrow G_1 \text{ and } G_2' \in \downarrow G_2 \}$$
$$= \downarrow \{ (G_1' \cdot G_2') \mid G_1' \leq G_1 \text{ and } G_2' \leq G_2 \} = \downarrow (G_1 \cdot G_2)$$

Note that, from Proposition 1,it follows that $G_1' \cdot G_2' \leq G_1 \cdot G_2$. That is, $\downarrow$ in an homomorphism between $\langle \mathbf{C}_{Lb}, \cdot \rangle$ and $\langle \mathbf{V}_{Lb}, \cdot \rangle$.

Theorem 10 (*Homomorphism between causal graphs and values*)
The mapping $\downarrow\colon \mathbf{C}_{Lb}, \longrightarrow \mathbf{V}_{Lb}$ is an injective homomorphism between $\langle \mathbf{C}_{Lb}, +, * \cdot \rangle$ and $\langle \mathbf{V}_{Lb}, +, *, \cdot \rangle$,

Proof. This follows directly from the Theorem 1 and Proposition 16

Proposition 17

For every causal term t without addition "+" there exists a causal graph G such that $t = \downarrow G$.

Proof. In case that t is a label, by definition is the principal ideal $\downarrow t$. We proceed by structural induction. In case that $t = \prod S$, by induction hypothesis, for every $t_i \in S$ there is some causal graph G_i such that $t_i = \downarrow G_i$ and from Theorem 1 it follows that

$$t = \prod S = \prod \{ \downarrow G_i \mid t_i \in S \} = \downarrow \{ G_i \mid t_i \in S \}$$

which is a principal ideal. In case that $t = t_1 \cdot t_2$, by induction hypothesis, $t_1 = \downarrow G_1$ and $t_2 = \downarrow G_2$ and from Proposition 16, it follows that

$$t = t_1 \cdot t_2 = \downarrow G_1 \cdot \downarrow G_2 = \downarrow (G_1 \cdot G_2)$$

which is a principal idea.

Lemma 8.1

Let G_u and G_v be two causal graphs. If $G_u \supseteq G_v$, then $term(G_u) \leq term(G_v)$ follows from the equivalence laws reflected in Figures 12 and 13.

Proof. By definition $term(G_u) * term(G_v)$ is equal to

$$\prod \{ u_1 \cdot u_2 \mid (u_1, u_2) \in G_u \} * \prod \{ v_1 \cdot v_2 \mid (v_1, v_2) \in G_v \} \tag{11}$$

and since, $G_u \supseteq G_v$, it follows that every edge $(v_1, v_2) \in G_v$ also belong to G_u. Thus applying associative and conmutative laws we can rewrite (11) as

$$\prod \{ u_1 \cdot u_2 \mid (u_1, u_2) \in G_u \backslash G_v \} * \prod \{ (v_1 \cdot v_2) * (v_1 \cdot v_2) \mid (v_1, v_2) \in G_v \}$$

that, by product idempotence, is clearly equivalent to

$$\prod \{ u_1 \cdot u_2 \mid (u_1, u_2) \in G_u \backslash G_v \} * \prod \{ (v_1 \cdot v_2) \mid (v_1, v_2) \in G_v \}$$

That is, $term(G_u) * term(G_v) = term(G_u)$ which implies $term(G_u) \leq term(G_v)$.

Lemma 8.2

Let U and V be two causal values. If $U \subseteq V$, then $term(U) \leq term(V)$ follows from the equivalence laws reflected in Figures 12 and 13.

Proof. By definition $term(U) + term(V)$ is equal to

$$\sum \{ term(G_u) \mid G_u \in U \} + \sum \{ term(G_v) \mid G_v \in V \} \tag{12}$$

and since, $U \subseteq V$, it follows that every c-graph $G_u \in U$ also belong to V. Thus applying associative and conmutative laws we can rewrite (12) as

$$\sum \{ G_u \mid (u_1, u_2) \in U \} + \sum \{ G_v + G_v \mid G_v \in V \backslash U \}$$

that, by summ idempotence, is clearly equivalent to

$$\sum \{ G_u \mid (u_1, u_2) \in U \} + \sum \{ G_v \mid G_v \in V \backslash U \}$$

That is, $term(U) + term(V) = term(V)$ which implies $term(U) \leq term(V)$.

Proposition 18 (*Application associativity*)
Let T, U and W be three causal values. Then, it holds that $U \cdot (T \cdot W) = (U \cdot T) \cdot W$ and further
$U \cdot T \cdot W = \downarrow\{ \ G_U \cdot G_T \cdot G_W \ \big| \ G_U \in U, \ G_T \in T \text{ and } G_W \in W \ \}$.

Proof. By definition it follows that

$$(U \cdot T) \cdot W = \downarrow\{ \ G_U \cdot G_T \ \big| \ G_U \in U \text{ and } G_T \in T \ \} \cdot W$$
$$= \downarrow\{ \ G' \cdot G_W \ \big| \ G_U \in U, \ G_T \in T, \ G' \le G_U \cdot G_T \text{ and } G_W \in W \ \}$$
$$= \downarrow\{ \ (G_U \cdot G_T) \cdot G_W \ \big| \ G_U \in U, \ G_T \in T \text{ and } G_W \in W \ \}$$

In the same way, it also follows that

$$U \cdot (T \cdot W) = \downarrow\{ \ G_U \cdot (G_T \cdot G_W) \ \big| \ G_U \in U, \ G_T \in T \text{ and } G_W \in W \ \}$$

Then it is enough to show that $(G_U \cdot G_T) \cdot G_W = G_U \cdot (G_T \cdot G_w) = G_U \cdot G_T \cdot G_W$ which holds due to Proposition 2.

Proposition 19 (*Application absorption*)
Let T, U and W be three causal values. Then $T = T + U \cdot T \cdot W$ and $U \cdot T \cdot W = T * U \cdot T \cdot W$

Proof. From Proposition 18 it follows that

$$U \cdot T \cdot W = \downarrow\{ \ G_U \cdot G_T \cdot G_W \ \big| \ G_U \in U, \ G_T \in T \text{ and } G_W \in W \ \}$$

Furthermore, for every c-graph $G_T \in T$, it holds that $G_U \cdot G_T \cdot T_W \le G_T$. Then, since T is an ideal, it follows that $G_U \cdot G_T \cdot T_W \in T$ and consequently $U \cdot T \cdot W \subseteq T$. Thus $U \cdot T \cdot W \cup T = T$ and $U \cdot T \cdot W \cap T = U \cdot T \cdot W$ and, by definition, these equalities can be rewritten as $U \cdot T \cdot W + T = T$ and $U \cdot T \cdot W * T = U \cdot T \cdot W$.

Proposition 20 (*Application distributivity w.r.t. additions*)
Let T, U and W be three causal values. Then, it holds that $U \cdot (T + W) = (U \cdot T) + (U \cdot W$ and $(U + T) \cdot W = (U \cdot W) + (T \cdot W)$.

Proof. By definition, it follows that

$$(U \cdot T) + (U \cdot W) = (U \cdot T) \cup (U \cdot W)$$
$$= \downarrow\{ \ G_U \cdot G_T \ \big| \ G_U \in U \text{ and } G_T \in T \ \} \cup \downarrow\{ \ G_U \cdot G_T \ \big| \ G_U \in U \text{ and } G_W \in W \ \}$$
$$= \downarrow\{ \ G_U \cdot G' \ \big| \ G_U \in U \text{ and } G' \in T \cup W \ \} = U \cdot (T \cup W) = U \cdot (T + W)$$

Furthermore $(U + T) \cdot W = (U \cdot W) + (T \cdot W)$ holds symmetrically.

Product distributivity laws follows from Propositions 5 and 6.

Corollary 5
The applications distributivity properties reflected in Figure 13 hold.

Proof. From proposition 17, it follows that c, d and e are the principal ideals $\downarrow G_c$, $\downarrow G_d$ and $\downarrow G_e$ for some c-graphs G_c, G_d and G_e. Furthermore, from Proposition 5 distributivity holds for causal graphs, i.e. $G_c \cdot (G_d * G_e) = (G_c \cdot G_d) * (G_c \cdot G_e)$ and $(G_c * G_d) \cdot G_e = (G_c \cdot G_e) * (G_d \cdot G_e)$ and from Theorem 10 it follows that this also holds for they principal ideals $\downarrow G_c$, $\downarrow G_d$ and $\downarrow G_e$. In the same way, from Proposition 6, it follows $c \cdot d \cdot e = (c \cdot d) * (d \cdot e)$.

Proposition 21 (*Application identity and annihilator*)
Given a causal value T, it holds that $T = 1 \cdot T$, $T = T \cdot 1$, $0 = T \cdot 0$ and $0 = 0 \cdot T$.

Proof. Note that 1 and 0 respectively correspond to $\mathbf{C}_{Lb}$ and $\emptyset$ and by definition it follows that

$$1 \cdot T = \downarrow \{ \ G \cdot G_T \ \big| \ G \in \mathbf{C}_{Lb} \text{ and } G_T \in T \ \} = T$$
$$0 \cdot T = \downarrow \{ \ G \cdot G_T \ \big| \ G \in \emptyset \text{ and } G_T \in T \ \} = \emptyset = 0$$

The other cases are symmetric.

Proposition 22 (*Application idempotentce*)
Given a label l, it holds that $l \cdot l = l$

Proof. By definition l corresponds to the causal value $\downarrow G_l$ where G_l is the causal graphs which contains the only edge (l,l). Furthermore $G_l \cdot G_l = G_l$ and from Theorem 10 it follows that $\downarrow G_l = \downarrow G_l \cdot \downarrow G_l$ and consequently $l = l \cdot l$.

Proposition 23 (*Causal term representation*)
For every causal value T is equal to $\sum \{ \ term(G) \ \big| \ G \in T \ \}$.

Proof. For every causal term T, it holds that $T = \sum \{ \ \downarrow G \ \big| \ G \in T \ \}$. Furthermore, from Proposition 3, it follows that every causal graph G is equal to $\prod \{ \ l_1 \cdot l_2 \ \big| \ (l_1, l_2) \text{ is an edge of } G \ \}$ and then, from Theorem 10, it holds that $\downarrow G = \prod \{ \ l_1 \cdot l_2 \ \big| \ (l_1, l_2) \text{ is an edge of } G \ \}$. Consequently $T = \sum \{ \ term(G) \ \big| \ G \in T \ \}$.

Proposition 24
Let c, d and e be causal terms without "$+$" and different from 1. Then from the algebraic equivalences in Figures 12 and 13 it follows that $c \cdot d * d \cdot e = c \cdot d * d \cdot e * c \cdot e$.

Proof.

$$
\begin{aligned}
c \cdot d * d \cdot e * c \cdot e &= c \cdot d \cdot e * c \cdot e \\
&= c \cdot d \cdot e * c \cdot e \\
&= c \cdot (d \cdot e * e) \\
&= c \cdot (d \cdot e \cdot 1 * e) \\
&= c \cdot (d \cdot e \cdot 1) \\
&= c \cdot d \cdot e \\
&= c \cdot d * d \cdot e
\end{aligned}
$$

Proposition 25
Every causal term without sums c can be rewritten as $normal(c) \stackrel{\text{def}}{=} \prod \{ \ l_1 \cdot l_2) \ \big| \ (l_1, l_2) \text{ is an edge of } G \ \}$ for some casual graph G using the algebraic equivalences in Figures 12 and 13.

Proof. We star rewritten c as $\prod C$ where every causal term $x \in C$ is in the form of $l_1 \cdot \ldots \cdot l_n$ by by applying the distributive law until no product is in the scope of "$\cdot$". Then we remove each occurrence of 1 applying the identity law and we replace term x by $x \cdot x$ when x is a label. That is, we have C' s.t. every $x' \in C$ is equal to $x' = l_1 \cdot \ldots \cdot l_n$ with $n > 1$ and $l_i \neq 1$. Then we rewrite each $x' \in C'$ as $l_1 \cdot l_2 * l_2 \cdot l_3 * \ldots * l_{n-1} \cdot l_n$ by successively application of the equivalence $c \cdot d \cdot e = c \cdot d * d \cdot e$. Finally we add every transitive edge applying the equivalence $c \cdot d * d \cdot e = c \cdot d * d \cdot e * c \cdot e$ (Proposition 24). $\quad\blacksquare$

Proposition 26
Every causal term t can be rewritten as $normal(t) \overset{\text{def}}{=} \{\, normal(c) \mid c \in S \,\}$ for some set of terms without sums S using the algebraic equivalences in Figures 12 and 13.

Proof. We only have to rewrite t as a causal term where sums are not in the scope of "$*$" and "$\cdot$" by successively application of distributivity over sums. Then the statement holds from Proposition 25 $\quad\blacksquare$

Proposition 27
Let c and d be causal terms without "$+$". Then $c \leq d$ iff $c = c * d$ follows from the algebraic equivalences in Figures 12 and 13.

Proof. From Proposition 25, c and d can be rewrite as $c' = \prod\{\, l_1 \cdot l_2) \mid (l_1, l_2)$ is an edge of $G_c \,\}$ and $d' = \prod\{\, l_1 \cdot l_2) \mid (l_1, l_2)$ is an edge of $G_d \,\}$ for some causal graphs G_c and G_d. Furthermore, from Proposition 3 and Theorem 10, it holds that $\downarrow G_c = c'$ and $\downarrow G_d = d'$. Thus, by definition $c \leq d$ iff $\downarrow G_c \leq \downarrow G_d$ iff $\downarrow G_c = \downarrow G_c \cap \downarrow G_d$ iff $\downarrow G_c = \downarrow G_c * \downarrow G_d$ iff $c' = c' * d'$. Let us see that $c' = c' * d'$ can be follows from the algebraic equivalences.