

LOGIC PROGRAMMING ON A TOPOLOGICAL BILATTICE

Melvin FITTING

*Department of Mathematics and Computer Science:
Herbert H. Lehman College (CUNY), Bronx, NY 10468*

*Department of Computer Science: Department of Philosophy:
The Graduate School and University Center (CUNY),
33 West 42 Street, New York, NY 10036*

Binet MLFLC@CUNYVM

Received October 1987

Accepted December 1987

We investigate the semantics of logic programming using a generalized space of truth values. These truth values may be thought of as evidences for and against – possibly incomplete or contradictory. The truth value spaces we use essentially have the structure of M. Ginsberg's bilattices, and arise from topological spaces. The simplest example is a four-valued logic, previously investigated by N. Belnap. The theory of this special case properly contains that developed in earlier research by the author, on logic programming using Kleene's three-valued logic.

1. INTRODUCTION

Logic programming has generally, though not exclusively, been done in the context of two-valued Classical logic. In [3] (and also [6]) we argued that a three-valued generalization might be better, allowing for *undefined* as well as *true* and *false*. See also [12], [13] and [14]. *Undefined* ought to be taken into account because of the possibility of infinite regress inherent in logic – or any – programming. More recently we began exploring the utility of adding a fourth truth value, *overdefined*, to allow for inconsistencies in logic programs. It turns out the semantic techniques of [3] extend readily to the four-valued setting and we can treat programs which are inconsistent but which still may contain useful information provided we 'stay away from' the inconsistent parts. This four-valued logic was introduced to Computer Science in [2], a paper which we strongly recommend. That it applies to logic programming was noted independently by the author and in [8].

Further, in [7] we showed that one can even make good operational and denotational sense of logic programming with a Heyting algebra as the space of truth values. It is possible to think of the elements of the Heyting algebra as being the 'justifications' for statements, rather than just an indication of their truth or falsity.

It is a reasonable question, then, what is common to all these generalizations of logic programming? M. Ginsberg's notion of *bilattice* ([9] and [10]) provides a framework in which this question can be addressed. Indeed, it provides a nice conceptual setting for many notions connected with default logic, and database theory as well. But we find the version of negation that he uses too restrictive, and some of the other machinery too weak. We leave for another time an abstract formalization of a bilattice generalization that is suitable for our purposes. Instead we work with a concretely defined class of examples, arising from topological spaces, which in turn can be thought of as arising from Kripke Intuitionistic models. These can be treated uniformly, and provide a common generalization of the various logic programming extensions discussed above.

* Research supported in part by NSF grant CCR-8702307 and by PSC-CUNY grant 667295.

2. MOTIVATION

The following is meant to be suggestive only. Suppose we have a Kripke Intuitionistic logic model, \mathbf{K} [11]. Then a topological space can be associated naturally with \mathbf{K} : the domain \mathbf{D} is the set of possible worlds of \mathbf{K} , and a subset O of \mathbf{D} is called open if it is closed under the accessibility relation of \mathbf{K} . This topological space, in turn, gives rise to the Heyting algebra of its open sets, and there are well-known relationships between this algebra and the Kripke model \mathbf{K} with which we began. At any rate, it is the topological space that concerns us now.

For any proposition P , the set of worlds of \mathbf{K} in which P holds is a set closed under the accessibility relation of \mathbf{K} , hence an open set of \mathbf{D} . Dually, the set of worlds of \mathbf{K} in which P fails will be a closed set. We might think of the set of worlds in which a proposition holds as a measure of our belief in that proposition: the larger the set, the stronger our belief. Likewise the set of worlds in which a proposition fails can be taken to reflect our disbelief. Thus, ideally, corresponding to a proposition P we could associate an ordered pair $\langle O, C \rangle$, where O is the set of possible worlds in which P holds (an open set), and C is the set of possible worlds in which P fails (a closed set), and O and C are complementary sets. This pair is an exact depiction of our beliefs for and against the proposition P .

In general, though, our information may be less than perfect. We may not know how P behaves in all worlds. Of course, if we know that P holds in some possible world, we also know it holds in any world accessible from it, so the set of worlds in which we know P holds will be open, though it may be smaller than the set of all worlds in which P holds. Similarly for the worlds in which P fails. Thus, in 'real life' we may have to settle for a pair $\langle O, C \rangle$ where O and C do not, between them, exhaust all possible worlds. Even worse, information we have may be erroneous, leading us to the pair $\langle O, C \rangle$ where O and C overlap! So, adopting a generous viewpoint, we will take as our 'truth-values' all pairs $\langle O, C \rangle$ whatsoever, subject only to the conditions that O be open and C be closed.

Next, following [9] and [10], we define two natural partial orderings on these 'truth-values'. The first is the *knowledge* or *k-ordering*: our knowledge has increased if our degree of belief, or our degree of disbelief, or both, have gone up. The second is the *truth* or *t-ordering*: the 'degree' of truth has increased if our degree of belief has gone up, or our degree of disbelief has gone down, or both. It turns out that these orderings are intimately connected with each other. We consider them formally starting in the next section.

3. TOPOLOGICAL BILATTICES

Let \mathbf{D} be a topological space, fixed for this section. All our definitions are relative to it. We introduce several constructs in this section, based on \mathbf{D} . The resulting collection of sets and relations constitutes what we loosely term a *topological bilattice*. An exact definition will not be needed.

Definition. A \mathbf{D} truth value is a pair $\langle O, C \rangle$ where O is open and C is closed. We write $\mathbf{T}(\mathbf{D})$ for the space of \mathbf{D} truth values. A \mathbf{D} truth value is:

- 1) overdefined if $O \cap C \neq \emptyset$,
- 2) consistent if $O \cap C = \emptyset$,
- 3) exact if $O \cap C = \emptyset$ and $O \cup C = \mathbf{D}$.

Definition. We define two partial orderings on the family $\mathbf{T}(\mathbf{D})$:

- 1) the knowledge order: $\langle O_1, C_1 \rangle \leq_k \langle O_2, C_2 \rangle$ if $O_1 \subseteq O_2$ and $C_1 \subseteq C_2$.
- 2) the truth order: $\langle O_1, C_1 \rangle \leq_t \langle O_2, C_2 \rangle$ if $O_1 \subseteq O_2$ and $C_2 \subseteq C_1$.

We give diagrams of the two simplest topological bilattices. The first, Figure 1, arises from the one-world model whose only world is a . It is the bilattice of Classical logic, since the one-world Kripke models are essentially the Classical models. It is also fundamental in the sense that an isomorphic copy of it occurs as part of every bilattice. The logic arising from this bilattice was extensively investigated in [2].

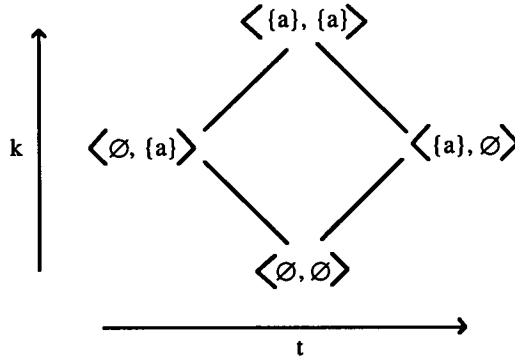


Figure 1.

Figure 1 is essentially a double Hasse diagram, and is intended to be read in the following way. A path uphill from x to y indicates that $x \leq_k y$; a path to the right from x to y indicates that $x \leq_t y$. This convention applies throughout the paper.

The second example, Figure 2, is the topological bilattice deriving from a two-world Kripke model, with worlds a and b , such that b is accessible from a but not conversely. By changing the accessibility relation but not the set of worlds, other topological bilattices based on the same set of worlds can result.

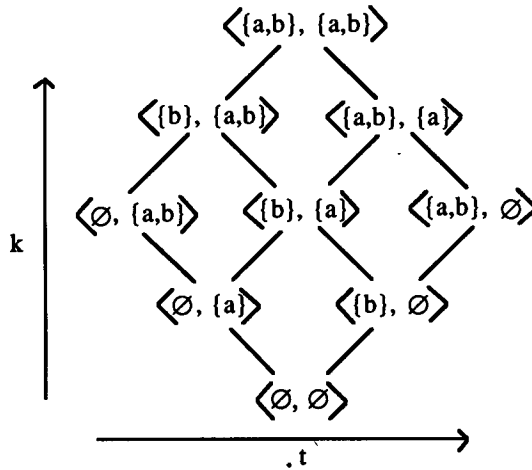


Figure 2.

Now we look at the properties of the space $T(D)$ under the knowledge and the truth orderings separately, then together, and finally we introduce a weak notion of negation.

Under the truth order:

$T(D)$ is a complete lattice. The smallest element is $\langle \emptyset, D \rangle$, which we denote *false*; the largest element is $\langle D, \emptyset \rangle$, which we denote *true*. Note that *false* indicates no belief, but total disbelief, while *true* is the dual. We denote the least upper bound of a non-empty set S in this ordering by $\bigvee S$, and the greatest lower bound by $\bigwedge S$. It is easy to see that $\bigvee S = \langle O, C \rangle$, where $O = \bigcup \{O \mid \langle O, C \rangle \in S\}$ and $C = \bigcap \{C \mid \langle O, C \rangle \in S\}$. Also, $\bigwedge S =$

$\langle O, C \rangle$, where $O = \text{interior} \cap \{O \mid \langle O, C \rangle \in S\}$ and $C = \text{closure} \cup \{C \mid \langle O, C \rangle \in S\}$. Note that if S is finite, the interior and closure operations are not needed. We also use the notation $a \vee b$ for $\bigvee \{a, b\}$, and $a \wedge b$ for $\bigwedge \{a, b\}$. $\mathbf{T}(\mathbf{D})$ is a distributive lattice as well.

Of course the family $\mathbf{T}(\mathbf{D})$ is closed under the operations \bigvee and \bigwedge , but a moments work will show that so are the family of consistent truth values and the family of exact truth values. Thus, each of these is a complete, distributive lattice. In fact, the family of exact truth values is isomorphic to the family of open subsets of \mathbf{D} , and hence is even a Heyting algebra.

Under the knowledge order:

Again $\mathbf{T}(\mathbf{D})$ is a complete lattice. This time the smallest element is $\langle \emptyset, \emptyset \rangle$, which we denote \perp ; the largest element is $\langle \mathbf{D}, \mathbf{D} \rangle$, which we denote \top . \perp indicates neither belief nor disbelief, while \top is simultaneous total belief and total disbelief. We denote the least upper bound of a non-empty set S in the knowledge ordering by $\sum S$, and the greatest lower bound by $\prod S$. And now, $\sum S = \langle O, C \rangle$, where $O = \bigcup \{O \mid \langle O, C \rangle \in S\}$ and $C = \text{closure} \cup \{C \mid \langle O, C \rangle \in S\}$; and $\prod S = \langle O, C \rangle$, where $O = \text{interior} \cap \{O \mid \langle O, C \rangle \in S\}$ and $C = \bigcap \{C \mid \langle O, C \rangle \in S\}$. We use the notation $a + b$ for $\sum \{a, b\}$, and $a \times b$ for $\prod \{a, b\}$. $\mathbf{T}(\mathbf{D})$ is a distributive lattice under this ordering as well.

The family of exact truth values is never closed under $+$ or \times . For example, *true* and *false* are exact, but $\text{true} + \text{false} = \top$ and $\text{true} \times \text{false} = \perp$, neither of which is exact. The family of consistent truth values is closed under \prod , and under directed \sum , and so constitutes what is sometimes called a complete semi-lattice.

Interconnections:

Of course, if $a \leq_k a'$ and $b \leq_k b'$ then $a + b \leq_k a' + b'$. This is a trivial consequence of $\mathbf{T}(\mathbf{D})$ being a lattice under \leq_k . But more surprisingly, if $a \leq_t a'$ and $b \leq_t b'$ then $a + b \leq_t a' + b'$. More generally, suppose that for $A, B \subseteq \mathbf{T}(\mathbf{D})$ we use $A \leq_t B$ to mean: for each $a \in A$ there is some $b \in B$ with $a \leq_t b$, and for each $b \in B$ there is some $a \in A$ with $a \leq_t b$. Then $A \leq_t B$ implies $\sum A \leq_t \sum B$. In fact, each of the two lattice orderings respects the meet and join of the other. In [9] and [10] this was taken as one of the defining properties of the abstract notion of bilattice.

The features in the preceding paragraph are the interconnections we will need in this paper. But there are others, notably distributive laws. Each of the four operations $+$, \times , \bigvee and \bigwedge is distributive over the others. This played no role in the definition of bilattice in Ginsberg's work.

Negation:

We define negation on members of $\mathbf{T}(\mathbf{D})$ in a straightforward way. Roughly, the idea is to reverse the roles of belief and of disbelief, but we must also respect the topological mechanism.

Definition. $\neg \langle O, C \rangle = \langle \text{interior } C, \text{closure } O \rangle$.

The family of all truth values is closed under negation, and so are the families of consistent, and of exact truth values.

If $a \leq_t b$ then $\neg b \leq_t \neg a$. But also, if $a \leq_k b$ then $\neg a \leq_k \neg b$. (If we know more about b than about a , we also know more about its negation than we do about the negation of a .) These were also among the defining conditions for the abstract notion of bilattice in Ginsberg's work.

Finally, though we will not need it in what follows, we also have $a \leq_t \neg \neg a$, which means our negation is something like that of Intuitionistic logic. This is weaker than what was assumed in Ginsberg's version, which postulated that $\neg \neg a = a$.

4. LOGIC

Formulas in a first-order language may be assigned truth values in $\mathbf{T}(\mathbf{D})$ in two ways, since we could naturally associate logical conjunction and disjunction with the meet and join operations of the \leq_k or of the \leq_t ordering. We use the truth ordering here. The role of the operations arising from the knowledge ordering will be considered in a subsequent paper.

Let \mathbf{M} be a non-empty set. By $\mathbf{L}(\mathbf{M})$ we mean the first-order language whose terms are variables and members of \mathbf{M} , and in which formulas are built up using \wedge , \vee , \neg , \forall and \exists in the usual way. We are going to think of quantifiers in this language as being over \mathbf{M} , and we will assign members of $\mathbf{T}(\mathbf{D})$ as truth values.

Definition. An interpretation is a mapping from closed atomic formulas of $\mathbf{L}(\mathbf{M})$ to $\mathbf{T}(\mathbf{D})$. If v and w are interpretations, we write $v \leq_k w$ provided $v(A) \leq_k w(A)$ for every closed atomic formula A . Similarly for $v \leq_t w$.

Interpretations can be extended to maps from all closed formulas to $\mathbf{T}(\mathbf{D})$, valuations. We use the same notation for an interpretation and for its valuation extension. Formally, we have the following.

Definition. Let v be an interpretation. v is extended to all formulas using the following:

$$\begin{aligned} v(X \wedge Y) &= v(X) \wedge v(Y) \\ v(X \vee Y) &= v(X) \vee v(Y) \\ v(\neg X) &= \neg v(X) \\ v((\forall x)\phi(x)) &= \bigwedge \{v(\phi(m)) \mid m \in \mathbf{M}\} \\ v((\exists x)\phi(x)) &= \bigvee \{v(\phi(m)) \mid m \in \mathbf{M}\}. \end{aligned}$$

Proposition 4-1. Let X be a closed formula of $\mathbf{L}(\mathbf{M})$, and let v and w be interpretations. Then

- 1) $v \leq_k w$ implies $v(X) \leq_k w(X)$;
- 2) $v \leq_t w$ implies $v(X) \leq_t w(X)$, provided X does not contain any negations.

Proof. Item 2) follows immediately because the operations of a lattice are monotone with respect to the lattice ordering. Item 1) makes use of the fact that the operations associated with the truth ordering are respected by the knowledge ordering, and also of the fact that the knowledge ordering respects negation.

5. KRIPKE MODELS, GENERALIZED

We give a simple generalization of Kripke Intuitionistic logic models that allows partial or conflicting information. This generalization is then related with the valuations in topological bilattices that we have been considering. We confine things to the propositional case, because the fit between Kripke and topological models for intuitionistic logic is not a good one where the universal quantifier is concerned. This arises from the feature of Kripke models that allows different possible worlds to have different domains of quantification. But the propositional case should be sufficient to illustrate the naturalness of the valuation rules we have been using.

A (propositional) *Kripke Intuitionistic model* is a triple $(\mathcal{G}, \mathcal{R}, \models)$ where \mathcal{G} is a non-empty set (of possible worlds), \mathcal{R} is a transitive, reflexive relation on \mathcal{G} (of accessibility), and \models is a relation between possible worlds and propositional atomic formulas meeting the condition that, for any world $\Gamma \in \mathcal{G}$, if $\Gamma \models A$ and $\Gamma \mathcal{R} \Delta$ then $\Delta \models A$.

Given a Kripke model $(\mathcal{G}, \mathcal{R}, \models)$, the relation \models is extended to all propositional formulas using the conditions:

$$\Gamma \models (X \wedge Y) \iff \Gamma \models X \text{ and } \Gamma \models Y$$

$$\begin{aligned}
\Gamma \models (X \vee Y) &\iff \Gamma \models X \text{ or } \Gamma \models Y \\
\Gamma \models \neg X &\iff \text{for every } \Delta \text{ with } \Gamma \mathcal{R} \Delta, \\
&\quad \Delta \not\models X \\
\Gamma \models (X \supset Y) &\iff \text{for every } \Delta \text{ with } \Gamma \mathcal{R} \Delta, \\
&\quad \Delta \models X \implies \Delta \models Y
\end{aligned}$$

In a Kripke model, if $\Gamma \models A$ and $\Gamma \mathcal{R} \Delta$, then $\Delta \models A$, where A is atomic. It is easily shown by induction on formula complexity that this extends to all formulas. Since we have not been taking \supset into account in earlier sections, we will not consider it further here, though we note that the work below extends naturally to incorporate it.

In any Kripke model, at a world Γ , some formulas will hold, others will not. We introduce some notation that will allow us to state facts of this sort directly. For this purpose we use *signed formulas*, expressions of the forms TX and FX , where X is a formula and T and F are two new symbols. We write:

$$\begin{aligned}
\Gamma \models TX &\text{ for } \Gamma \models X \\
\Gamma \models FX &\text{ for } \Gamma \not\models X
\end{aligned}$$

Then the general properties of this extended notation follow easily from the conditions satisfied by Kripke models stated above.

$$\begin{aligned}
\Gamma \models T(X \wedge Y) &\iff \Gamma \models TX \text{ and } \Gamma \models TY \\
\Gamma \models F(X \wedge Y) &\iff \Gamma \models FX \text{ or } \Gamma \models FY \\
\Gamma \models T(X \vee Y) &\iff \Gamma \models TX \text{ or } \Gamma \models TY \\
\Gamma \models F(X \vee Y) &\iff \Gamma \models FX \text{ and } \Gamma \models FY \\
\Gamma \models T\neg X &\iff \text{for every } \Delta \text{ with } \Gamma \mathcal{R} \Delta, \Delta \models FX \\
\Gamma \models F\neg X &\iff \text{for some } \Delta \text{ with } \Gamma \mathcal{R} \Delta, \Delta \models TX \\
\Gamma \models TX &\implies \text{for every } \Delta \text{ with } \Gamma \mathcal{R} \Delta, \Delta \models TX \\
\Delta \models FX &\implies \text{for every } \Gamma \text{ with } \Gamma \mathcal{R} \Delta, \Gamma \models FX \\
&\text{exactly one of } \Gamma \models TX \text{ and } \Gamma \models FX
\end{aligned}$$

These conditions can be thought of as saying under what circumstances (in what worlds) we have positive information and under what circumstances we have negative information. Further, we can take these conditions as basic, and forget the original Kripke model conditions that gave rise to them. Now, it is the final condition above that requires consistency and completeness in our information. At no world can we ever have both TX and FX , though we must have one of them. Suppose we drop this condition, thus allowing information to be inconsistent or incomplete.

Definition. A weak Kripke model is a structure $\langle \mathcal{G}, \mathcal{R}, \models \rangle$ that meets all the signed formula conditions stated above, except possibly the final one.

Let $\langle \mathcal{G}, \mathcal{R}, \models \rangle$ be a weak Kripke model. As sketched in §2, we can associate a topological space with a Kripke structure: in this case the set of points is \mathcal{G} , and a set is topologically open if it is closed under the \mathcal{R} relation. Then a topological bilattice can be associated with this space. Further, we define a valuation as follows. For each atomic formula A , set $v(A) = \langle \mathbf{O}, \mathbf{C} \rangle$ where $\mathbf{O} = \{ \Gamma \in \mathcal{G} \mid \Gamma \models TA \}$ and $\mathbf{C} = \{ \Gamma \in \mathcal{G} \mid \Gamma \models FA \}$. This interpretation then extends to all formulas in the usual way. The principle fact about the resulting map is the following.

Proposition 5-1. For any propositional formula X , $v(X) = \langle \mathbf{O}, \mathbf{C} \rangle$ where $\mathbf{O} = \{ \Gamma \in \mathcal{G} \mid \Gamma \models TX \}$ and $\mathbf{C} = \{ \Gamma \in \mathcal{G} \mid \Gamma \models FX \}$.

We omit the proof of this. It says that the evaluation of truth values in this topological bilattice can be thought of as corresponding to evaluation in a Kripke model, but treating positive and negative information as if each came from different sources, and thus is independent of the other.

Finally we observe that, for one-world weak Kripke models, the conditions required above essentially collapse to those of saturated sets in [3] or [4] that are not required to be consistent or complete.

6. PROGRAMS

We present a logic programming language that generalizes Horn clause programming. The choice of underlying data structure is left open; in this we follow [5]. And of course we allow truth values in $\mathbf{T}(\mathbf{D})$, a topological bilattice.

Definition. A data structure is a tuple $\langle \mathbf{M}; \mathbf{R}_1, \dots, \mathbf{R}_n \rangle$ where \mathbf{M} is a non-empty set and $\mathbf{R}_1, \dots, \mathbf{R}_n$ are relations on \mathbf{M} , called the given relations of the data structure.

From now on we assume that a unique relation symbol R_i has been associated with each given relation \mathbf{R}_i of the data structure $\langle \mathbf{M}; \mathbf{R}_1, \dots, \mathbf{R}_n \rangle$. We refer to these relation symbols as reserved, and think of them as representing the given relations.

Definition. A definition of the n place relation symbol P is an expression of the form $P(x_1, \dots, x_n) \leftarrow F(x_1, \dots, x_n)$, where the body, $F(x_1, \dots, x_n)$, is any formula of $\mathbf{L}(\mathbf{M})$ whose free variables are among x_1, \dots, x_n . A program is a finite set of definitions such that no relation symbol has more than one definition, and no definition is for a reserved relation symbol. A program is positive if no definition body contains a negation symbol.

Conventional Horn clause programs are a special case of the programs defined above. Horn clause bodies can be taken to be conjunctions; multiple Horn clauses containing the same relation symbol in the head can be combined using disjunction; and free variables in bodies that do not appear in heads can be thought of as existentially quantified. Not every program in our sense corresponds to a Horn clause, however.

Definition. An interpretation v from the language $\mathbf{L}(\mathbf{M})$ to the topological bilattice $\mathbf{T}(\mathbf{D})$ is said to be in the data structure $\langle \mathbf{M}; \mathbf{R}_1, \dots, \mathbf{R}_n \rangle$ provided, for each $i = 1, \dots, n$, $v(R_i) = \mathbf{R}_i$.

Definition. v is a model for a program \mathbf{P} in the data structure $\langle \mathbf{M}; \mathbf{R}_1, \dots, \mathbf{R}_n \rangle$ provided v is an interpretation in this data structure and, for each n -place relation symbol P : if P has no definition in \mathbf{P} then $v(P(a_1, \dots, a_n)) = \text{false}$ for each $a_1, \dots, a_n \in \mathbf{M}$; and if P has a definition $P(x_1, \dots, x_n) \leftarrow F(x_1, \dots, x_n)$ in \mathbf{P} , then for each $a_1, \dots, a_n \in \mathbf{M}$, $v(P(a_1, \dots, a_n)) = v(F(a_1, \dots, a_n))$.

Thus a model is a valuation that assigns instances of definition heads the same values it assigns to their bodies. The condition covering relation symbols without definitions is related to the idea of negation as failure. It would also be reasonable to take the 'default' value to be \perp in this case, though doing so would affect several key results below.

The problem now is to show that models exist, and that among them there is a simplest. Of course the word 'simplest' can be given a meaning with respect to either the knowledge ordering or the truth ordering. In what follows we consider both possibilities, and establish relationships between them. For this purpose, we associate an operator with each program, mapping interpretations to interpretations. We denote this operator by Φ . It is a generalization of the T operator of [1].

Definition. Let \mathbf{P} be a program and $\langle \mathbf{M}; \mathbf{R}_1, \dots, \mathbf{R}_n \rangle$ be a data structure. $\Phi_{\mathbf{P}}$ is the map on interpretations given by the following conditions: for any interpretation v , $\Phi_{\mathbf{P}}(v)$ is the interpretation w such that

- 1) if R_i is a reserved relation symbol, $w(R_i) = \mathbf{R}_i$;
- 2) if P is an unreserved relation symbol, with definition $P(x_1, \dots, x_n) \leftarrow F(x_1, \dots, x_n)$ in program \mathbf{P} , then for $a_1, \dots, a_n \in \mathbf{M}$, $w(P(a_1, \dots, a_n)) = v(F(a_1, \dots, a_n))$;
- 3) if P is neither reserved nor has a definition in \mathbf{P} then $w(P(a_1, \dots, a_n)) = \text{false}$.

It is easy to see that the models of a program \mathbf{P} are exactly the fixed points of $\Phi_{\mathbf{P}}$. So from now on we concentrate on the behavior of $\Phi_{\mathbf{P}}$. The key result is the following, which follows immediately from Proposition 4-1. We omit a detailed proof.

Proposition 6-1. For an arbitrary program \mathbf{P} , $\Phi_{\mathbf{P}}$ is monotone with respect to the \leq_k ordering and, if \mathbf{P} is positive, $\Phi_{\mathbf{P}}$ is monotone with respect to the \leq_t ordering.

$\mathbf{T}(\mathbf{D})$ is a complete lattice with respect to both knowledge and truth orderings. Since the family of all maps from a set to a complete lattice yields another complete lattice using the induced pointwise ordering on the maps, then the family of interpretations becomes a complete lattice under both the \leq_k and the \leq_t orderings. Then it follows from the Knaster-Tarski Theorem that $\Phi_{\mathbf{P}}$ always has a smallest and a greatest fixed point in the knowledge ordering, and also does in the truth ordering provided program \mathbf{P} is positive.

In §3 we defined notions of *exact* and *consistent* for truth values. These are extended pointwise to interpretations. Thus we call an interpretation v *exact* if it assigns to each closed atomic formula an exact truth value. Similarly for *consistent*.

Proposition 6-2. The least fixed point of $\Phi_{\mathbf{P}}$ in the knowledge ordering is consistent.

Proof. The least fixed point of a monotone operator in a complete lattice can be 'constructed' as the limit of a transfinite sequence in the following way. The initial term of the sequence is the smallest member of the lattice. The $\alpha + 1^{\text{st}}$ term is the result of applying the monotone operator to the α^{th} term. And at limit ordinals we take the least upper bound of the family of earlier terms (which will constitute a chain).

In the lattice of interpretations under the \leq_k ordering, the smallest member is the interpretation that maps every closed atomic formula to \perp , which is a consistent interpretation. We noted in §3 that the family of consistent truth values was closed under \wedge , \vee , \bigwedge , \bigvee and \neg . It follows that $\Phi_{\mathbf{P}}$ applied to a consistent interpretation yields another consistent interpretation. Finally, again in §3, we observed that the family of consistent truth values was closed under directed \sum . It follows that the lub, in the \leq_k ordering, of a chain of consistent interpretations is another consistent interpretation.

Then, by transfinite induction, the least fixed point of $\Phi_{\mathbf{P}}$ must be a consistent interpretation.

Proposition 6-3. Suppose \mathbf{P} is a positive program. Then both the least and the greatest fixed points of $\Phi_{\mathbf{P}}$ in the \leq_t ordering must be exact.

Proof. The argument that the smallest fixed point is exact is similar to that of Proposition 6-2. Now, of course, we need to use the closure of the family of exact truth values under \wedge , \vee , \bigwedge and \bigvee . The smallest interpretation in the truth ordering is the one that maps every closed atomic formula to *false*, which is an exact interpretation.

The argument to establish the assertion concerning the greatest fixed point is dual. This time we use a transfinite sequence of interpretations that begins with the biggest, and comes down to a fixed point, rather than starting with the smallest and working upward. The biggest interpretation now is the one that maps every closed atomic formula to *true*, which is exact. And the rest of the argument dualizes in a similar, straightforward manner.

For the rest of this section let \mathbf{P} be a fixed positive program with $\Phi_{\mathbf{P}}$ the corresponding operator. Let v_t and V_t be the least and greatest fixed points of $\Phi_{\mathbf{P}}$ in the \leq_t ordering, and let v_k be the least fixed point of $\Phi_{\mathbf{P}}$ in the \leq_k ordering. Then, by the preceding propositions, v_t and V_t are exact, and v_k is consistent, and each of these is a model for the program \mathbf{P} . There are some simple relationships between these models which are easy to establish.

Since v_k is the smallest fixed point in the \leq_k ordering, and both v_t and V_t are fixed points, then $v_k \leq_k v_t$ and $v_k \leq_k V_t$.

Since v_t is the smallest and V_t is the biggest fixed point in the \leq_t ordering, and v_k is a fixed point, then $v_t \leq_t v_k \leq_t V_t$.

Proposition 6-4. *Let v_t , V_t and v_k be as above. Then v_k and v_t give the same closed atomic formulas the value true, and v_k and V_t give the same closed atomic formulas the value false.*

Proof. Let A be a closed atomic formula. Since $v_t \leq_t v_k$, and *true* is the largest truth value in the \leq_t ordering, $v_t(A) = \text{true}$ implies $v_k(A) = \text{true}$. In the other direction, since $v_k \leq_k v_t$ then $v_k(A) = \text{true}$ implies $\text{true} \leq_k v_t(A)$. It is easy to see that no exact, or even consistent truth value can be strictly above an exact truth value in the knowledge order. Since *true* is exact and v_t is exact, it follows that $v_t(A) = \text{true}$. This establishes the first claim.

Since $v_k \leq_t V_t$, and *false* is the smallest truth value in the \leq_t ordering, $V_t(A) = \text{false}$ implies $v_k(A) = \text{false}$. Further, since $v_k \leq_k v_t$, $v_k(A) = \text{false}$ implies $\text{false} \leq_k V_t(A)$. But again, V_t is exact, so if $\text{false} \leq_k V_t(A)$ then $\text{false} = V_t(A)$. This establishes the second claim.

The simplest bilattice, arising from a one-world or Classical Kripke model, was shown earlier as Figure 1. We repeat it as Figure 3, but without the inessentials due to Kripke models shown. Instead we only notate items by their role in the bilattice. The four-valued logic depicted in Figure 3 was examined in detail in [2].

In Figure 3 the only exact truth values are *false* and *true*, those of conventional two-valued Classical logic. Also the only consistent truth values are *false*, *true* and \perp , and the operations on these corresponding to the truth ordering are those of Kleene's three-valued logic, which was the logic used in [3]. Since v_t and V_t must be exact, Proposition 6-4 completely determines their characteristics for this bilattice. This result was essentially established in §7 of [3].

Conventional logic programming can be considered to be working with the sublogic of this four-valued logic consisting of exact truth values, namely *false* and *true*. Further, the ordering used is \leq_t ; if there is nothing in a program to force an atomic formula to be *true*, then the default is *false*. Proposition 6-3 then says why only the Classical two truth values arise for logic programming without negation.

In [3] we used Kleene's three-valued logic, which amounts to working with just the consistent truth values of the four-valued logic above. The key innovation of that paper, though we did not think of it in these terms at the time, was to use the logical operations associated with the truth ordering, but take least fixed points with respect to the knowledge ordering. Proposition 6-2 accounts for why this approach never got us beyond a three-valued logic.

Just as negation moves us from a two to a three valued logic, truth values that are not consistent can arise naturally if more general constructs are allowed in writing logic programs. For example, suppose a logic program is distributed over two sites which do not communicate. If I issue the query Q and one site responds *true* while the other responds

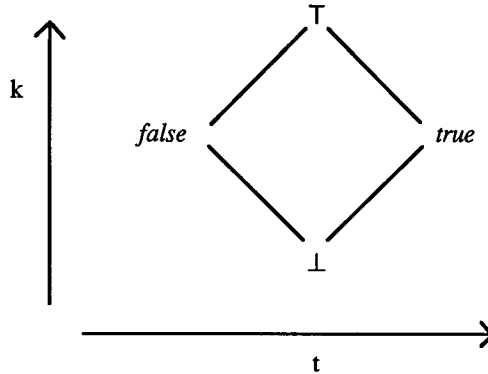


Figure 3.

false, how am I to merge these answers? One possibility is to accept both, thus assigning Q the value \top , or *overdefined*. Another possibility is to insist on consensus, and so in this case assign Q the value \perp or *undefined*. These informal actions correspond to the bilattice operations $+$ and \times . It is reasonable then, to consider extensions of conventional logic programming languages that allow such operations. But this is a topic for further research.

REFERENCES

- [1] Apt, K.R. and Van Emden, M.H., Contributions to the theory of logic programming, *J. Assoc. Comput. Mach.*, 29 (1982), 841–862.
- [2] Belnal, N.D. Jr., A useful four-valued logic, in *Modern Uses of Multiple Valued Logic*, J. M. Dunn, J.M. and Epstein, G. (eds.), Reidel (1977), Dordrecht, 8–37.
- [3] Fitting, M., A Kripke-Kleene semantics for logic programs, *Journal of Logic Programming*, 2 (1985), 295–312.
- [4] Fitting, M., Notes on the mathematical aspects of Kripke's theory of truth, *Notre Dame J. of Formal Logic*, 27 (1986) 75–88.
- [5] Fitting, M., *Computability Theory, Semantics, and Logic Programming*, Oxford University Press, New York (1987).
- [6] Fitting, M., Partial models and logic programming, *Theoretical Computer Science*, 48 (1987) 229–255.
- [7] Fitting, M., Pseudo-Boolean valued Prolog, forthcoming in *Studia Logica*.
- [8] Foo, N.Y. and Rao, A.S., Open world and closed world negations, manuscript (1987).
- [9] Ginsberg, M., *Multi-valued Logics*, technical report 86–29 KSL, Stanford University (1986).
- [10] Ginsberg, M., Multi-Valued logics, *Proc. AAAI-86, fifth national conference on artificial intelligence*, (1986), 243–247, Morgan Kaufmann, Los Altos, CA.
- [11] Kripke, S., Semantical analysis of intuitionistic logic I, in *Formal systems and recursive functions*, (1963), 92–130, North-Holland, Amsterdam.
- [12] Kunen, K., Negation in logic programming, K. Kunen, *J. of Logic Programming*, 4 (1987), 289–308.
- [13] Lassez, J.L. and Maher, M.J., Optimal fixedpoints of logic programs, *Theoretical Computer Science*, 39 (1985), 15–25.
- [14] Mycroft, A., Logic programs and many-valued logic, *Proc. first STACS Conf.* (1983).