A Bilattice Based Fixed Point Semantics for Integrating Imperfect Information

Daniel Stamate
Department of Computing, Goldsmiths, University of London
Lewisham Way, New Cross, London SE14 6NW, United Kingdom
d.stamate@doc.gold.ac.uk

Abstract

We present an approach to reasoning non-uniformly by default with uncertain, incomplete and inconsistent information using sets of rules/extended logic programs in the context of logics with a bilattice structure. A fixed point semantics for extended logic programs used in the process of inference is described, along with its computational approach. We show how this theoretic approach is applicable to the problem of integration of imperfect information coming from multiple sources.

1 Introduction

Information integration has received much attention for a number of year now in Database, Artificial Intelligence, Logic Programming, Multimedia Information Systems, World Wide Web and other research communities. Various approaches to information fusion have been proposed, adapted to the particular research areas, as the integration of data in a distributed database or from different databases, or the integration of information collected by an agent from other sources, or merging belief bases represented using logic programs, or integrating information coming from different medium sources as text, sound or image (as it is the case, for instance, in the query "find the full description of albums containing music played by piano and having elements of classical and jazz as genre") or Web sources, etc.

In order to propose an approach to information integration, two main questions may arise: (1) How is information coming from multiple sources combined?, and (2) Given the problems of possible conflicting information coming from mutually contradictory sources, of missing information coming from incomplete sources, or of uncertain information coming from sources of limited reliability, what meaning can one assign to the fused information (that is, what is the result of the integration)? The information that is incomplete or totally or partially inconsistent or uncertain will be called imperfect information in what follows.

With respect to the first question, the approach to the information integration that we propose in this paper is based on the logic programming paradigm, as it uses inference rules to integrate information in a logic based context. The logic rules we use, however, form extended logic programs as there is a need to employ, apart operations as the conjunction $\wedge$, the disjunction $\vee$ and the negation $\neg$, two more operations, that can be easily given a particular meaning in information merging, called the consensus $\otimes$ and the collecting together operation $\oplus$, to be formally and most generally defined in the next section.

With respect to the second question, we first choose an appropriate formalism based on multiple valued logics expressed by the concept of bilattice, that is very powerful in expressing the three aspects of imperfect information, namely the uncertainty, the incompleteness and the inconsistency.

In order to illustrate the concept of bilattice, assume first that we want to express the truthness of an information A. In the ideal case we can employ the logical values true or false, but in many situations this approach is simplistic and not acceptable. If we use a degree between 0 and 1 instead of a classical logical value, the approach is more appropriate in expressing uncertainty but less helpful in expressing lack of information, or the presence of contradiction in information. Indeed, no value from [0,1] can express, alone, incompleteness or inconsistency. A natural idea would then be to assign an information

a pair $\langle c,d\rangle$ instead of one value, that would consist in a degree of confidence c and a degree of doubt d in [0,1], which do not necessarily add up to 1 (otherwise the single value c would suffice and we would be again in the previous case). In this setting $\langle 0,1\rangle$ and $\langle 1,0\rangle$, represent no confidence, full doubt, and full confidence, no doubt, so they would correspond to the classical values $false$ and $true$, respectively. On the other hand $\langle 0,0\rangle$ and $\langle 1,1\rangle$, represent no confidence, no doubt, and full confidence, full doubt, and they express a total lack of information or a total inconsistency, respectively. Two orders, namely the truth and the information (or knowledge) orders denoted $\leq_t$ and $\leq_i$, can naturally be defined on the set of confidence-doubt pairs, denoted $\mathscr{L}^{\mathscr{C}\mathscr{D}}$ and called the confidence-doubt logic [9], as follows: $\langle x,y\rangle \leq_t \langle z,w\rangle$ iff $x \leq z$ and $w \leq y$, and $\langle x,y\rangle \leq_i \langle z,w\rangle$ iff $x \leq z$ and $y \leq w$, where $\leq$ is the usual order between reals. Intuitively speaking, an increase in the truth order corresponds to an increase in the degree of confidence and a decrease in the degree of doubt, while an increase in the information order corresponds to an increase in both degrees of confidence and doubt. The meet and join operations w.r.t. $\leq_t$ and $\leq_i$ are denoted $\wedge$, $\vee$, $\otimes$ and $\oplus$, respectively. $\wedge$ and $\vee$ are the extensions of the classical conjunction and disjunction, while $\otimes$ and $\oplus$ are two new operations with potential of use in information integration as they naturally express the idea of consensus and of collecting together of two pairs of confidence-doubt degrees. A natural extension of the classical negation is defined by $\neg\langle x,y\rangle = \langle y,x\rangle$. We conclude the section by noting that the double structure of lattice induced by the two orders on $\mathscr{L}^{\mathscr{C}\mathscr{D}}$ is the basis of the general concept of bilattice introduced in [5].

2 Extended Logic Programs on Bilattices

Bilattices offer one of most capable frameworks to express, in the same time, the characteristics of the information to be incomplete, totally or partially inconsistent or uncertain. In addition, bilattices have an algebraic structure that allows to express approaches built on this concept in an elegant manner, and to facilitate elegant and often shorter proofs of results.

Definition 1. *A bilattice is a triple $\langle \mathscr{B}, \leq_t, \leq_i\rangle$, where $\mathscr{B}$ is a nonempty set, and $\leq_t$ and $\leq_i$ are partial orders each giving $\mathscr{B}$ the structure of a complete lattice.*

Given the bilattice $\mathscr{B}$, join and meet operations under $\leq_t$ are denoted $\vee$ and $\wedge$, called extended disjunction and conjunction, and join and meet operations under $\leq_i$ are denoted $\oplus$ and $\otimes$, called collecting together and consensus, respectively. The greatest and least elements under $\leq_t$ are denoted $true$ and $false$, and the greatest and least elements under $\leq_i$ are denoted $\top$ and $\bot$. A bilattice has a negation, denoted $\neg$, if $\neg$ is a unary operation which is antimonotone w.r.t. the truth order and monotone w.r.t. the information order. In addition $\neg true = false$, $\neg false = true$, $\neg\bot = \bot$ and $\neg\top = \top$.

Note that $\mathscr{L}^{\mathscr{C}\mathscr{D}}$ that we described in the previous section is a bilattice whose binary operations can be expressed as follows:

$$\langle x,y\rangle \wedge \langle z,w\rangle = \langle min(x,z), max(y,w)\rangle, \qquad \langle x,y\rangle \vee \langle z,w\rangle = \langle max(x,z), min(y,w)\rangle,$$
$$\langle x,y\rangle \otimes \langle z,w\rangle = \langle min(x,z), min(y,w)\rangle, \qquad \langle x,y\rangle \oplus \langle z,w\rangle = \langle max(x,z), max(y,w)\rangle.$$

In what follows we consider only bilattices for which all the distributive laws hold, an example of whom is $\mathscr{L}^{\mathscr{C}\mathscr{D}}$. These bilattices are called distributive, and it was proven that their non-unary operations of finite or infinite arity, are monotone w.r.t. both the truth and the information orders [2].

Fitting [2] extended the notion of logic program, that we will call *extended program*, to bilattices as follows. Let $\mathscr{B}$ be a bilattice, whose elements will be referred to as logical values.

Definition 2. *(1) A formula is an expression built up from literals and elements of $\mathscr{B}$, using $\wedge, \vee, \otimes, \oplus, \neg, \exists, \forall$. (2) A rule r is of the form $H(v_1,...,v_n) \leftarrow F(v'_1,...,v'_m)$ where the atomic formula $H(v_1,...,v_n)$ is the head,*

and the formula $F(v'_1,...,v'_m)$ is the body. It is assumed that the free variables of the body are among $v_1,...,v_n$. (3) A program is a finite set of rules assuming that no predicate letter appearing in the head of more than one rule.

Note that the restrictions/assumptions from (2) and (3) in the above definition cause no loss of generality, since any program as above, but without these restrictions, can be rewritten into another program respecting the restrictions [2]. On the other hand, any classical logic program can be written in the form described in definition 2, if one employs $\wedge$, $\vee$ and $true$ only, from the operations and elements of the bilattice $\mathscr{B}$, which obviously embeds the classical bivalued logic.For technical reasons, from now on, we consider any extended program to be instantiated (that is, all the free variables are replaced by ground terms). Note that, due to the way extended programs have been defined, their instantiated versions have no more than one rule with the same head.

Example 1. *Consider the following set of rules / extended program in the context of the bilattice $\mathscr{L}^{\mathscr{C}\mathscr{D}}$.*

$A \leftarrow B \oplus F$; $\quad B \leftarrow \neg E$;
$D \leftarrow B \vee C$; $\quad E \leftarrow \langle 0.7, 0.3\rangle$;
$C \leftarrow C \otimes E$.

Intuitively speaking, the information represented by E is assigned a confidence of 0.7 and a doubt of 0.3, as this fact is specified. B is the contrary of this information so it is assigned a confidence of 0.3 and a doubt of 0.7. F is an information whose confidence and support cannot be derived from the program as there is no rule defining F, so we assign F a confidence and doubt given by the reliability of the source providing it. That is, a default confidence and support will be assigned to F, specific to that source, as for instance a confidence of 0 and a doubt of 1 in a pessimistic, "not believing that source at all", approach, or a confidence of 1 and a doubt of 0 in an optimistic, "believing that source", approach. Let us assume the pessimistic approach for this source. C is information that should agree with E, as their consensus should be C. In particular C can be assigned a degree 0 of confidence and a degree 0 of doubt, if we are completely skeptical about the reliability of its source (that is, that source is not considered reliable, nor unreliable, so any default degrees of confidence and doubt from an information coming from that source will be 0 and 0 respectively). D is the information that is assigned the largest confidence and the least doubt from the confidence and doubt degrees of B and C, so 0.3 and 0, respectively. Note that C and D are incomplete as their degrees of confidence and doubt add up to less than 1. Finally A collects together the information B and F, so it will be assigned a confidence of 0.3 and a doubt of 1, so A is a partially inconsistent information as its confidence and doubt degrees add up to more than 1.

Roughly speaking, the example above illustrates the computation of the meaning/semantics of an extended program. In particular, the atoms (representing information to be integrated by rules) are assigned logical values from the underlying bilattice, process that needs the concept of interpretation and default interpretation.

Formally speaking, an interpretation is a mapping that assigns a logical value to each atom from the Herbrand base. In particular, if an atom cannot be derived from the rules then a default value, not necessarily the same for all atoms, is assigned to it. This default value is related to the degrees of reliability of the sources the information represented by the atom comes from. For instance if a source has a reliability of 90 percent, then the atom A, representing information coming from the source, and not being derived by any rule, is assigned by default a confidence of 0.9 and a doubt of 0.1. Formally speaking, a default interpretation is an interpretation. It is to be used to compensate the incompleteness of information derived using the program rules.

The derivation of information intuitively illustrated in the example above, will be formalised in the following section. In particular, as it is the case in most logic programming based approaches, the information deduction process will be expressed in terms of application of operators specific to the

program, until no new information is obtained, that is, until a fixed point is reached. However, in our framework one should take into account also the particularities of the underlying logic provided by a set of values ordered w.r.t. a truth order and an information order, and the process of deduction by default regarding an atom when there is no rule to apply for that atom.

3 Program Operators and Fixed Point Semantics

The following defines the order $\leq_p$ and naturally extends the truth and information orders to the set of interpretations denoted by Int_P.

Definition 3. *If I and J are interpretations then*
(1) $I \leq_t J$ if $I(A) \leq_t J(A)$
(2) $I \leq_i J$ if $I(A) \leq_i J(A)$
(3) $I \leq_p J$ if $I(A) \neq \bot$ implies $I(A) = J(A)$
for any ground atom A.

The interpretations can be extended to closed formulas (i.e. formulas not containing free variables) as follows: $I(X \wedge Y) = I(X) \wedge I(Y)$, and similarly for the other operations of $\mathscr{L}^{\mathscr{C}\mathscr{D}}$, $I((\exists x)F(x)) = \bigvee_{s \in GT} I(F(s))$, and $I((\forall x)F(x)) = \bigwedge_{s \in GT} I(F(s))$, where GT stands for the set of all ground terms. If $I(B) = \beta$ we say that the formula B evaluates to the logical value β with respect to I. However, in some cases we can find out the value a closed formula evaluates to, no matter if some atoms are assigned the value $\bot$ - let us call them underdefined, thus the following concept:

Definition 4. *The closed formula B ultimately evaluates to the logical value β w.r.t. interpretation I, denoted by $B \equiv_I \beta$, if $J(B) = \beta$ for any interpretation J s.t. $I \leq_p J$.*

Let $I_\top$ be the interpretation obtained from I by assigning the value $\top$ to any underdefined atom. We have $B \equiv_I \beta$ iff $I(B) = I_\top(B) = \beta$.

The first inference operator assigned to an extended program P, called the *production operator* denoted Φ_P and defined below, intuitively corresponds to the activation of the program rules:

$$\Phi_P(I)(A) = \beta \text{ if } (\exists A \leftarrow B \in P \text{ and } B \equiv_I \beta), \text{ or}$$
$$\bot, \text{ otherwise.}$$

The second type of inference assumes the use of a fixed interpretation $\mathscr{D}$ called the *default interpretation*. Roughly speaking, the value of each atom A in the default interpretation is seen as being derived from the reliability degrees of the sources the information represented by A is coming from. For instance if two sources consisting in two medical studies found evidence, based on statistical tests with a 0.05 significance level, that medication m is effective in treating ailment a, while the other that the evolution of the ailment a is independent of whether or not the medication m was administered to the tested patients, then the atom $Effective(m,a)$ would be assigned the partially inconsistent value $\langle 0.95, 0.95 \rangle$ in the interpretation $\mathscr{D}$. This value will be used whenever no other value can be inferred for this atom from the program.

We introduce now an intermediary operator called the *refining operator*, denoted by Ψ_P, whose role is to refine an arbitrary default information X (part of the interpretation $\mathscr{D}$), in the sense that X either has to safely complete the information I obtained by activating the rules, in which case X is not modified by Ψ_P, or has to be modified into a new interpretation $\Psi_P(X,I)$ that safely completes I. Formally,

$$\Psi_P(X,I) = Rev(X, \Phi_P(Rev(X,I) \oplus I))$$

where $Rev(X,J)$ is an interpretation X' s.t. $X'(A) = X(A)$ for any ground atom A for which either $J(A) = \perp$ or $X(A) = J(A)$, and $X'(A) = \perp$ for any other ground atom A. We say that X' is the revision of X w.r.t. J.

Note that, by employing the refining operator, we wish to obtain the "best" default information X used to complete the interpretation I. Formally, we have the following requirements: (1) $X \leq_p \mathscr{D}$; (2) $X = \Psi_P(X,I)$; and (3) under the previous two conditions X is maximal w.r.t. $\leq_p$. Roughly speaking X is to be a part of the default interpretation $\mathscr{D}$ (condition 1), that, when it is revised by the sure information encoded in I and then is further revised by the information that is deducted using the rules applied to I completed with X, it is stable, that is, it does not change to refinement (condition 2). In addition X is supposed to complete as much as possible the information encoded in I (condition 3). That is, we are interested in the maximal fixed points of the operator $\lambda X \Psi_P(X,I)$ that are parts of $\mathscr{D}$, which we call *actual default interpretations* with respect to I and $\mathscr{D}$. We show below, via algebraic methods, that there exists a unique actual default interpretation w.r.t. I and $\mathscr{D}$.

Let $(S, \leq)$ be a complete semilattice. We define a diagonal contraction on S as being a binary operator $T' : S^2 \to S$ that satisfies $T'(X,X) \leq X$ for any $X \in S$. We provide the following useful lemmas.

Lemma 1. *If T is a monotone operator defined on the complete semilattice $(S, \leq)$ the following hold:*
(1) T has a least fixed point w.r.t. $\leq$.
(2) if Y is an element of S s.t. $T(Y) \leq Y$ then T has a greatest fixed point X below Y. Moreover X can be obtained as the limit of the following sequence: $X_0 = Y$, $X_n = T(X_{n-1})$ if n is a successor ordinal and $T(X_n) = inf_{\leq, m<n} T(X_m)$ if n is a limit ordinal.

Lemma 2. *Let T' be a binary operator defined on the complete semilattice $(S, \leq)$ which is monotone in its first argument and antimonotone in its second argument and is a diagonal contraction. If Y is an arbitrary element of S then T' has a greatest fixed point $X = T'(X,X)$ below Y. Moreover X can be obtained as the limit of the following sequence: $X_0 = Y$, $X_n = T'(X_{n-1}, X_{n-1})$ if n is a successor ordinal and $T(X_n) = inf_{\leq, m<n} T'(X_m, X_m)$ if n is a limit ordinal.*

We have the following properties of the production and the refining operators.

Proposition 1. *Φ_P is monotone w.r.t. $\leq_i$ and $\leq_p$ orders.*

Let $\Theta(X,Y,I) = Rev(X, \Phi_P(Rev(Y,I) \oplus I))$. Obviously $\Psi_P(X,I) = \Theta(X,X,I)$. We have the following:

Lemma 3. *$(\lambda X, \lambda Y)\Theta(X,Y,I)$ is monotone in its first argument and antimonotone in its second argument and is a diagonal contraction w.r.t. $\leq_p$.*

As a consequence of Lemmas 2 and 3 we get:

Proposition 2. *$(\lambda X)\Psi_P(X,I)$ has a greatest fixed point below $\mathscr{D}$ w.r.t. $\leq_p$, denoted by $Def_P^{\mathscr{D}}(I)$.*

Note that Proposition 2 involves that $Def_P^{\mathscr{D}}(I)$ is the unique actual default interpretation, while Lemma 2 provides a means of computation for $Def_P^{\mathscr{D}}(I)$ consisting in starting with the default interpretation $\mathscr{D}$ and iterating the operator $(\lambda X)\Psi_P(X,I)$ until a fixed point is reached. We call $Def_P^{\mathscr{D}}$ the *default operator*, as it is obvious that it reflects the application of the inference by default.

The two types of inference described above are now combined via a new operator, denoted Γ_P and called the *integrating operator*. Formally $\Gamma_P(I) = \Phi_P(I) \oplus Def_P^{\mathscr{D}}(I)$. Roughly speaking, given an interpretation I encoding the information inferred from the program so far, $\Gamma_P(I)$ encodes the new information currently inferred from the program.

Roughly speaking, in order to generate the information that can be derived from the extended program P we start with the least degree of information characterized by an interpretation I_0 in which all

ground atoms are underdefined, denoted by $Const_{\perp}$ (i.e. nothing is known). We apply the two types of inference to the current information, which corresponds to an application of Γ_P operator, and we get a new interpretation I_1. This process is continued until nothing changes, that is, until a fixed point is reached. Formally we define the sequence $\mathscr{S}$ as follows:

$I_0 = Const_{\perp}$,
$I_n = \Gamma_P(I_{n-1})$ for a successor ordinal $n \geq 1$,
$I_n = inf_{\leq_p, m<n} I_m$ for n a limit ordinal.

We have:

Theorem 1. *The following hold:*
(1) $\mathscr{S}$ is increasing w.r.t. $\leq_p$ order (and thus w.r.t. $\leq_i$) and reaches a limit denoted by s.
(2) $\Gamma_P(s) = s$
(3) for any x s.t. $\Gamma_P(x) = x$ we have $s \leq_i x$.

Thus s is the least fixed point of Γ_P, and represents the minimal information that can be inferred from the extended program P completed with the default information $\mathscr{D}$. We chose s to designate the semantics of P. Note that any fixed point of Γ_P is deductively closed w.r.t. the program P and the default interpretation $\mathscr{D}$, and the two types of inference. Computationally speaking, we have:

Proposition 3. *If the program P does not contain any functional symbol, the semantics of P can be generated by iterative application of the integration operator in a finite number of steps , even if the underlying bilattice is infinite.*

Lemma 4. *If $Values(P)$ is the set of logical values appearing in the program P, and Closure(S) is the closure of the set of logical values from a subset S of the bilattice $\mathscr{B}$, to which one adds the elements $true$, $false$, $\top$, and $\perp$, w.r.t. the negation and the finite and infinite join and meet operations of $\mathscr{B}$, then $Closure(Values(P))$ is a finite bilattice.*

The proof of Proposition 3 is based on Theorem 1 and Lemma 4. Indeed, note that the logical values of any atom in the process of the computation of the semantics of P are elements of $Closure(Values(P))$, and are obtained as an increasing sequence w.r.t. $\leq_p$, and thus the evaluation of the semantics of P finishes in a finite number of steps since the Herbrand base is also finite.

4 Related Work

Our approach can be related in the first instance to other works regarding reasoning under uncertainty based on multivalued logics, in particular on bilattices, as those authored by Fitting. [3] defined the (multivalued) stable models for extended programs in bilattices that generalize the concept of stable models in the conventional bivalued logic [4]. We show below that if we consider the default interpretation $\mathscr{D}$ assigning the value $false$ to any ground atom, the semantics of P defined by our approach coincides with Fitting's multivalued stable model that has the least degree of information:

Proposition 4. *Let P be an extended program considered on the bilattice $\mathscr{B}$, and $mstable(P)$ be its multivalued stable model, as defined in [3], which is the lowest w.r.t. the information order. Then the semantics of P w.r.t. the default interpretation $\mathscr{D}$ coincides with $mstable(P)$.*

We show also that our semantics captures the α-fixed models of extended programs on bilattices, introduced by the author in [7]. For different logical values α, in particular false, true and $\perp$, the α-models provide various meanings to the same program, depending on how one chooses to complete

the missing information by adopting a pessimistic, optimistic, or skeptical approach respectively. It was proven in [7] that α-fixed models capture successful conventional semantics as the well-founded semantics [10], the three-valued stable semantics [8], the bi-valued stable semantics [4] and the Kripke-Kleene semantics [1]. Thus the semantics for information integration presented in this work is a natural extension of the above successful conventional bi-valued or three-valued semantics of conventional logic programs.

Proposition 5. *Given an extended program P considered on the bilattice $\mathscr{B}$, for any logical value α from $\mathscr{B}$, the α-fixed model of P, as defined in [7], coincides with the semantics of P w.r.t. the default interpretation that uniformly assigns the value α to any ground atom, as defined in the current approach.*

We are currently studying how the semantics defined for information integration in this approach, strongly related to the multivalued stable models (that in turn generalize the conventional stable models), can be related to recent work, as for instance [6], which provides a logic programming based approach making use of a program semantics based on stable models, for merging belief bases. We currently investigate how the two approaches integrating information/beliefs and their corresponding program semantics can be compared, given the link with the stable model concept, although the present framework seems more general as based on multivalued logics.

References

[1] M. C. Fitting. A Kripke-Kleene semantics for logic programs. *J. of Logic Program.*, 2(4):295–312, 1985.

[2] M. C. Fitting. Bilattices and the semantics of logic programming. *J. of Logic Program.*, 11(1–2):91–116, 1991.

[3] M. C. Fitting. Fixpoint semantics for logic programming—a survey. *Theor. Comput. Sci.*, 278(1–2):25–51, 2002.

[4] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R. A. Kowalski and K. A. Bowen, eds., *Proc. of 5th Int. Conf. and Symp. on Logic Programming (Washington, DC, Aug. 1988)*, pp. 1070–1080. MIT Press, 1988.

[5] M. L. Ginsberg. Multivalued logics: a uniform approach to reasonning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.

[6] J. Hue, O. Papini, and E. Würbel. Merging belief bases represented by logic programs. In C. Sossai and G. Chemello, eds., *Proc. of 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2009 (Verona, July 2009)*, v. 5590 of *Lect. Notes in Artif. Intell.*, pp. 371–382. Springer, 2009.

[7] Y. Loyer, N. Spyratos, and D. Stamate. Parameterised semantics for logic programs—a unifying framework. *Theor. Comput. Sci.*, 308(1–3):429–447, 2003.

[8] T. C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundam. Inform.*, 13(4):445–463, 1990.

[9] D. Stamate. Information representation through extended logic programs in bilattices. In B. Bouchon-Meunier et al., eds., *Uncertainty and Intelligent Information Systems*, pp. 419–432. World Scientific, 2008.

[10] A. van Gelder, K. S. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *J. of ACM*, 38(3):620–650, 1991.