

Abduction in Annotated Logic Programming

Krishnaprasad Thirunarayan *
Department of Computer Science and Engineering,
Wright State University,
Dayton, Ohio 45435.

1 Introduction

Pragmatically, the logic programming paradigm presents a reasonable trade-off between expressive power and computational efficiency [9, 14]. Here, we investigate techniques to make it more expressive for knowledge representation, while simultaneously retaining the computational advantages of efficiency and simplicity. In Section 1.1, we review annotated logic programs. In Section 1.2, we motivate the integration of deductive and abductive reasoning. In Section 2, we present an informal description of the proposed theory of abductive reasoning for annotated logic programs through examples. In Section 3, we describe the detailed syntax and develop a model-theoretic semantics of annotated logic programs by amalgamating concepts from logic programming [1] and multi-valued logics [8]. We identify a class of annotated logic programs called the *stratified* programs which can be given a *unique minimal supported Herbrand model* as their meaning [21]. We then show how to integrate abductive reasoning into this annotated logic framework by generalizing the work described in [18, 19]. We formalize the notion of an *explanation*, and specify when an explanation can be regarded as *acceptable*. Finally, we conclude in Section 5.

1.1 Annotated Logic Programs

[2, 3] pursued a four-valued logic approach as an alternative to classical predicate calculus to localize the effect of a contradiction in a first-order language. [24, 25, 7] used similar ideas to provide semantics to *strict* inheritance hierarchies and logic programs. [8] generalized it to multi-valued logics to make precise the book-keeping operations performed by a non-

monotonic reasoning system that handles both strict and defeasible information. In all the above cases, the semantics is given by mapping literals to truth values that are ordered on two different scales — the truth scale and the information scale. For example, Tweety being a bird contributes a supporting evidence to its flying ability, while Tweety being a penguin contributes a defeating evidence. This can be represented by mapping *fly(Tweety)* to the constants *+bird* and *-penguin* respectively. Furthermore, knowing that Tweety is a penguin is more informative than knowing that it is a bird. This can be represented by letting the constant *penguin* have higher information-content than the constant *bird*.

In approaches described in [10, 11, 12, 13], the underlying language is extended using annotations, which can be thought of as abbreviated justifications or pieces of evidence. Some areas of application are rule-based expert systems with uncertainty [10], temporal reasoning problems [12], reasoning with inconsistency [4, 11], etc. In [13], we use them to represent nonmonotonic multiple inheritance networks.

1.2 Integrating deductive and abductive reasoning

In a number of areas such as medical diagnosis [16, 26], fault diagnosis [22, 18], temporal reasoning [23], recognition [19], planning [6], natural language processing [5], etc. a large fragment of relevant knowledge can be expressed by specifying the “causal” relationships that exist among the various entities. In such domains the two basic operations of interest for information retrieval are:

- Given a set of causes, determine their cumulative effect.
- Given a set of observations, hypothesize a set of possible causes that can explain the observa-

*Work supported in part by NSF grant IRI-9009587.

tions.

Reasoning from causes to effects is predominantly deductive in nature and its formalization is the subject matter of first-order logic and its various nonmonotonic extensions. Reasoning from effects to possible causes has been formalized through the various schemes for abductive reasoning.

In medical diagnosis, the symptoms are explained using disease hypotheses. Similarly, in fault diagnosis, the incorrect behaviour of a digital system can be explained by determining possible circuit faults. In temporal reasoning, the propositions that hold in the current state can be explained by determining a possible set of events that could have occurred. In the context of natural language processing, abductive reasoning can be used for word-sense disambiguation and story understanding. Furthermore, tasks like designing a treatment from a diagnosis of the disease, or reconfiguration to rectify a circuit fault, or inferring additional properties of the current state from plausible events, all call for deductive reasoning using hypothesized explanations. So we believe that marrying both deductive and abductive reasoning strategies in a unified framework can be useful in practice.

2 Abductive Reasoning in Annotated Logic Programs

We first motivate the relevant issues through examples.

Consider the representation of the diagnostic knowledge for the "burnout problem" in an engine [26].

```
cup_hit_by_stone ← stony_road, low_car_ht.  
oil_cup_holed ← cup_hit_by_stone.  
engine_temp_high ← oil_loss, engine_started.  
oil_loss ← oil_cup_holed.
```

To formalize abduction, we need to (1) specify the "vocabulary" for the hypotheses to account for the observation, and (2) make rigorous when a set of hypotheses can be regarded as *explaining* the observation. In our example, the hypotheses can be constructed using facts that do not appear in any rule-head. An explanation can be thought of as a set of hypotheses that deductively implies the observation. So, the set $\{stony_road, low_car_ht, engine_started\}$ is a valid explanation of the observation *engine_temp_high*.

Consider the chemical identification problem. A base transforms a red litmus paper blue. Both NaOH and KOH are bases. Under flame test, NaOH gives a golden yellow flame, while KOH gives a violet flame. The observation that a sample changes a red litmus blue can be explained by the facts — the sample is a base, or it is NaOH, or it is KOH. However, among these, the explanation *the sample is a base* seems to be the most "appropriate" by virtue of being the least restrictive. So, we wish to regard it as an *acceptable* explanation. Furthermore, using well-known facts from Chemistry, one can predict through deductive reasoning that the sample will be soapy to touch and would neutralize acids. Subsequently, when we learn the *outcome of the flame test*, the acceptable explanation can be refined to one of NaOH or KOH.

Consider information about the flying abilities of birds and mammals. Typically, birds fly, while mammals do not. Similarly, bats fly, while injured-bats do not. This can be expressed in our formalism as follows:

```
fly(X) : +bird ← bird(X) : +⊥  
fly(X) : -mammal ← mammal(X) : +⊥  
fly(X) : +bat ← bat(X) : +⊥  
fly(X) : -injured_bat ← injured_bat(X) : +⊥
```

Informally, the rule $p : \alpha \leftarrow q : \beta$ states that if there is evidence for q that is greater than or equal to β in information-content and that has the same truth-content as β , then one can confer support to p to the degree α . \perp (resp. ω) is a defeasible (resp. strict) evidence with the least (resp. greatest) information-content. The symbol $+$ (resp. $-$, $*$) indicates a supporting (resp. defeating, ambiguous) evidence. The evidences can be written in the ascending order of their information-content as \perp , mammal, bat, injured_bat and ω . bird is incomparable with mammal, bat and injured_bat on the information-scale, but it is larger than \perp and smaller than ω . *This ordering information can be derived by inspecting the corresponding inheritance hierarchy automatically* [13].

Given that the specimen *bates* is a mammal, that is, $mammal(bates) : +\omega$, we wish to conclude that *bates* does not fly, that is, $fly(bates) : -mammal$. Subsequently, when we learn facts like $bat(bates) : +\omega$ and $injured_bat(bates) : +\omega$, we wish to revise it to $fly(bates) : +bat$ and $fly(bates) : -injured_bat$ respectively. We formalize these ideas using the notion of a *supported model*.

Given that *bates* does not fly, we can hypothesize

that *bates* is probably a mammal. However, if *bates* is in fact a bat, then we can explain its inability to fly by assuming that it is an injured-bat. Formally, the two plausible explanations for $\text{fly}(\text{bates}) : -\perp$ are $\{\text{mammal}(\text{bates}) : +\perp\}$ and $\{\text{injured_bat}(\text{bates}) : +\perp\}$ which generate the conclusions $-\text{fly}(\text{bates}) : -\text{mammal}$ and $\text{fly}(\text{bates}) : -\text{injured_bat}$ respectively. However, notice that the former explanation is preferable to the latter one that has redundant information. We define the notion of *acceptable explanation* for this purpose. Furthermore, suppose we also learn that $\text{bat}(\text{bates}) : +\omega$ holds. Then, the explanation $\{\text{fly}(\text{bates}) : -\text{mammal}\}$ is no longer tenable in the face of the “stronger” conclusion $\text{fly}(\text{bates}) : +\text{bat}$. So, the only (acceptable) explanation for the fact that *bates* does not fly is $\{\text{injured_bat}(\text{bates}) : +\perp\}$, which implies $\{\text{fly}(\text{bates}) : -\text{injured_bat}\}$.

3 Annotated Logic Programs

3.1 Syntax

A *term* is an individual constant or a variable, or a “pattern” $f(t_1, \dots, t_n)$, where f is an n -ary function symbol and t_i ’s are terms. An *atom* is a propositional constant or a formula $q(t_1, \dots, t_n)$, where q is an n -ary predicate symbol and t_i ’s are terms. *Literals* are of the form $p : \tau$, where p is an atom and τ is an *annotation* drawn from the domain of annotations \mathcal{A} . A *rule* is an expression of the form $p : \tau \leftarrow q_1 : \gamma_1, \dots, q_m : \gamma_m$ where p and q are atoms and τ and γ_i ’s are annotations. The literal $p : \tau$ is referred to as the *rule-head*, while $q_1 : \gamma_1, \dots, q_m : \gamma_m$ are referred to as the *rule-body*. A *fact* is a *ground* (i.e., variable-free) literal. A *clause* is either a fact or a rule. An *annotated logic program* is a set of clauses.

Following [2, 3, 7, 8, 13], the annotations are ordered along two different dimensions: in one, they are partially ordered by \leq_k on the basis of their *information-content*; in the other, they are related by an equivalence relation \approx_t on the basis of their *truth-content*. We define:

$(\tau <_k \gamma)$ **iff** $(\tau \leq_k \gamma) \wedge (\tau \neq \gamma)$; and
 $(\tau \leq_{tk} \gamma)$ **iff** $(\tau \leq_k \gamma) \wedge (\tau \approx_t \gamma)$.

Furthermore, for our purposes, we assume that \approx_t has only the following three equivalence classes: \mathcal{C}^- , \mathcal{C}^* , and \mathcal{C}^+ , corresponding to the defeating, ambiguous, and supporting evidences, respectively.

We impose certain acyclicity restrictions on the annotated logic programs for reasons that will become

clear in the next section. This acyclicity requirement is analogous to the local stratification condition of [21] for ordinary logic programs. We formalize this requirement by defining a relation $<$ on the ground literals as follows: We say that $q : \beta <_{\mathbf{P}} p : \alpha$ if there is a ground instance $p : \alpha \leftarrow q : \beta$ of a rule in the program \mathbf{P} , or there is a ground instance $p : \alpha \leftarrow r : \gamma$ of a rule in \mathbf{P} and $q : \beta <_{\mathbf{P}} r : \tau$ and $\gamma \approx_t \tau$. We demand that, for the annotated logic programs of interest to us, the following restriction on the relation $<_{\mathbf{P}}$ be met: For all annotations α and β , $(q : \alpha <_{\mathbf{P}} q : \beta) \Rightarrow (\alpha \approx_t \beta)$. This condition prohibits *recursion through negation*. In the sequel, we refer to annotated logic programs that satisfy this condition as *stratified annotated logic programs*.

3.2 Model-theoretic Semantics

Let \mathbf{P} be an annotated logic program. The *domain* \mathcal{D} of Herbrand interpretations of \mathbf{P} is a collection of all individual constants mentioned in \mathbf{P} , and the ground (variable-free) terms that can be formed from them; it is often called the *Herbrand universe* of \mathbf{P} . A *Herbrand base*, $\mathcal{B}_{\mathbf{P}}$ of \mathbf{P} is a collection of all ground (variable-free) atoms p that use only terms of \mathcal{D} and the predicate symbols mentioned in \mathbf{P} .

A *Herbrand interpretation* \mathcal{I} of \mathbf{P} is a partial mapping from the Herbrand base, $\mathcal{B}_{\mathbf{P}}$, of \mathbf{P} to the set of annotations \mathcal{A} . Given two interpretations \mathcal{I} and \mathcal{J} , $\mathcal{J} \sqsubseteq_k \mathcal{I}$ (resp., $\mathcal{J} \sqsubseteq_{tk} \mathcal{I}$) iff for every atom $p \in \mathcal{B}_{\mathbf{P}}$, such that $\mathcal{J}(p)$ is defined, $\mathcal{I}(p)$ is also defined and $\mathcal{J}(p) \leq_k \mathcal{I}(p)$ (resp., $\mathcal{J}(p) \leq_{tk} \mathcal{I}(p)$). An interpretation \mathcal{I} is *minimal* in a set of interpretations \mathcal{S} if and only if there is no $\mathcal{J} \in \mathcal{S}$ such that $\mathcal{J} \sqsubseteq_k \mathcal{I}$ and $\mathcal{J} \neq \mathcal{I}$; \mathcal{I} is *maximal* if and only if there is no $\mathcal{J} \in \mathcal{S}$ such that $\mathcal{I} \sqsubseteq_k \mathcal{J}$ and $\mathcal{J} \neq \mathcal{I}$. A ground atom $p : \tau$ is *satisfied* in \mathcal{I} , denoted $\mathcal{I} \models_k p : \tau$, if and only if $\mathcal{I}(p)$ is defined and $\tau \leq_k \mathcal{I}(p)$; it is *strongly satisfied* in \mathcal{I} , denoted $\mathcal{I} \models_{tk} p : \tau$, if also $\tau \leq_{tk} \mathcal{I}(p)$. A ground (variable-free) rule $p : \tau \leftarrow q_1 : \gamma_1, \dots, q_n : \gamma_n$ is *satisfied* in \mathcal{I} if and only if $(\forall i : 1 \leq i \leq n : \mathcal{I} \models_{tk} q_i : \gamma_i)$ implies $\mathcal{I} \models_k p : \tau$. A nonground rule $p : \tau \leftarrow q : \gamma$ is *satisfied* in \mathcal{I} if and only if all its ground instances are satisfied in \mathcal{I} . (A *ground instance* of a rule r is a ground rule obtained from r by replacing variables with elements in the Herbrand universe, where different occurrences of the same variable are replaced by the same element.) An annotated logic program \mathbf{P} is *satisfied* in \mathcal{I} , if all its clauses are satisfied in \mathcal{I} . An interpretation \mathcal{I} is a *model* of a set of clauses \mathbf{P} if

and only if \mathbf{P} is satisfied under \mathcal{I} . A Herbrand model \mathcal{I} of \mathbf{P} is *supported* by \mathbf{P} if for every atom p such that $\mathcal{I}(p)$ is defined, we have $\mathcal{I}(p) = \text{lub}_k \{ \tau \mid p : \tau \leftarrow \text{BODY} \}$. We assume that \mathcal{I} is undefined on any atom that does not appear in the head of a ground instance of a rule in \mathbf{P} . Informally, \mathcal{I} is supported by \mathbf{P} if the evidences that it assigns to atoms are not stronger than what is warranted by \mathbf{P} .

It is possible that there does not exist a single supported model, or that there are several different supported models. This is in contrast with the ordinary logic programs, and the annotated programs for inheritance specifications considered in [13] where a unique minimal supported model always exists. For instance, consider the following program: $(p : -1 \leftarrow q : +1. q : +1 \leftarrow p : +1. p : +1.)$ The minimal Herbrand model of this program is: $\{p : *1, q : +1\}$. This model is not supported because $q : +1$ is unsupported as $p : +1$ is not strongly satisfied in the model. Similarly, the program $(p : \alpha \leftarrow q : \beta. q : \beta \leftarrow p : \alpha.)$ has two supported models: $\{\}$ and $\{p : \alpha, q : \beta\}$. However, if \leq_k is a semilattice (that is, every pair of elements has a least-upper bound) then we have the following result.

Theorem 3.1 *Every stratified annotated logic program has a unique minimal supported model.*

4 Formalization of Abductive Reasoning

We now formalize abduction along the lines of [18, 19]. The abductive framework consists of three sets — the set \mathbf{P} consisting of the annotated logic program, the set O of observations, and the set A of predicate names corresponding to the abducible literals. An observation is syntactically an atom with its truth value. That is, an observation is not required to specify the information-content of the annotation. The ground literals formed from A serve as possible hypotheses to explain the observations O .

We can recast the complete flying-ability-problem of Section 2 into the abductive framework as follows. The set \mathbf{P} is:

$$\begin{aligned} \text{fly}(X) : +\text{bird} &\leftarrow \text{bird}(X) : +\perp. \\ \text{fly}(X) : -\text{mammal} &\leftarrow \text{mammal}(X) : +\perp. \\ \text{fly}(X) : +\text{bat} &\leftarrow \text{bat}(X) : +\perp. \\ \text{fly}(X) : -\text{injured_bat} &\leftarrow \text{injured_bat}(X) : +\perp. \\ \text{mammal}(X) : +\omega &\leftarrow \text{bat}(X) : +\omega. \end{aligned}$$

$$\begin{aligned} \text{mammal}(X) : +\text{bat} &\leftarrow \text{bat}(X) : +\perp. \\ \text{bat}(X) : +\omega &\leftarrow \text{injured_bat}(X) : +\omega. \\ \text{bat}(X) : +\text{injured_bat} &\leftarrow \text{injured_bat}(X) : +\perp. \end{aligned}$$

where, $\perp <_k \text{bird} <_k \omega$ and $\perp <_k \text{mammal} <_k \text{bat} <_k \text{injured_bat} <_k \omega$. The first four rules specify conditions under which we can derive that an entity flies or does not fly, while the last four rules capture the class-subclass relationship. For instance, bats are a subclass of mammals. So if an entity is a bat then it is a mammal too. However if the entity is a bat only by default, then it is a mammal by default, to a degree determined by the evidence bat . The set A consists of the “class” names $\{\text{mammals}, \text{bats}, \text{injured_bats}\}$. The set O is either $\{\text{fly}(\text{bates}) : +\}$ or $\{\text{fly}(\text{bates}) : -\}$.

Now we make rigorous the notion of an explanation. Informally, an explanation of an observation O is a set of hypotheses H that, in conjugation with the program \mathbf{P} , implies O .

Definition 4.1 *A scenerio of (\mathbf{P}, A, O) is a set H of ground literals containing only the predicates in A .*

Definition 4.2 *An extension of (\mathbf{P}, A, O) is the unique minimal supported model associated with $\mathbf{P} \cup H$, where H is a scenerio of \mathbf{P} .*

Definition 4.3 *An explanation of O in (\mathbf{P}, A, O) is a scenerio H such that the observations O are strongly satisfied (that is, \leq_{kt}) in the extension of $\mathbf{P} \cup H$.*

In our example, the observation $\text{fly}(\text{bates}) : -$ can be explained by the two scenerios: $\{\text{mammal}(\text{bates}) : +\perp\}$ and $\{\text{injured_bat}(\text{bates}) : +\perp\}$. Similarly, the observation $\text{fly}(\text{bates}) : +$ can be explained by the scenerio $\{\text{bat}(\text{bates}) : +\perp\}$. Notice that, the scenerio $\{\text{mammal}(\text{bates}) : +\perp\}$ is a “better” explanation of $\text{fly}(\text{bates}) : -$ than the scenerio $\{\text{injured_bat}(\text{bates}) : +\perp\}$ because the latter explanation makes much “stronger” assumptions about the object in question than the former. To filter out such intuitively unsatisfactory explanations from all possible explanations the notion of acceptability is developed.

As a first cut, we say that an explanation H of O in (\mathbf{P}, A, O) is *acceptable* if there does not exist another explanation H' such that $H' \sqsubset_{kt} H$. But, this definition is unsatisfactory because, it does not rule out the scenerios such as $\{\text{injured_bat}(\text{bates}) : +\perp\}$ as unacceptable. (Note that the literals corresponding to the predicates mammal and injured_bat are

incomparable wrt $<_{kt}$.) We can attempt fixing this flaw by asserting that, an explanation H of O is acceptable if there does not exist another explanation H' such that $\text{extension}(H') \sqsubset_{kt} \text{extension}(H)$. This criterion rules out the scenario $\{\text{injured_bat}(\text{bates}) : +\perp\}$ as unacceptable in the presence of the scenario $\{\text{mammal}(\text{bates}) : +\perp\}$. However, this approach does not work in general as illustrated below: Given an abductive framework $(\{p : +1 \leftarrow q : \alpha, p : +2 \leftarrow r : \beta\}, \{q, r\}, \{p : +\})$, the above criterion does not unequivocally prefer the explanation $\{q : \alpha\}$ over the explanation $\{r : \beta\}$ as the literals corresponding to the atoms q and r are not related. This analysis leads us to the following final definition of acceptability that concentrates on the strength of evidence supporting the explained observations.

Definition 4.4 An explanation H of O in (P, A, O) is acceptable if and only if there does not exist another explanation H' such that $\text{extension}(H')|O \sqsubset_{kt} \text{extension}(H)|O$, where $E|O$ stands for the extension E restricted to the observation literals.

The extension corresponding to the explanation $\{\text{mammal}(\text{bates}) : +\perp\}$ contains $\text{fly}(\text{bates}) : -\text{mammal}$, while that corresponding to the explanation $\{\text{injured_bat}(\text{bates}) : +\perp\}$ contains $\text{fly}(\text{bates}) : -\text{injured_bat}$. Thus, the scenario $\{\text{mammal}(\text{bates}) : +\perp\}$ is the acceptable explanation of the observation $\text{fly}(\text{bates}) : -$. Similarly, for the abductive framework \mathcal{E} , the explanation $\{q : +\alpha\}$ is acceptable, while the explanation $\{r : +\beta\}$ is not.

In case we also know the following facts:

$\text{hairry}(X) : +\text{mammal} \leftarrow \text{mammal}(X) : +\perp.$

$\text{winged}(X) : -\text{mammal} \leftarrow \text{bat}(X) : +\perp.$

$\text{winged}(X) : +\text{bat} \leftarrow \text{bat}(X) : +\perp.$

then, $\{\text{mammal}(\text{bates}) : +\perp\}$ supports

$\{\text{hairry}(\text{bates}) : +\text{mammal}, \text{winged}(\text{bates}) : -\text{mammal}\},$

and $\{\text{injured_bat}(\text{bates}) : +\perp\}$ supports $\{\text{hairry}(\text{bates}) : +\text{mammal}, \text{winged}(\text{bates}) : +\text{injured_bat}\}.$

This illustrates how abductive reasoning allows us to generate explanations from observations, and how deductive reasoning can generate further predictions in this integrated framework [18, 23].

There can be several acceptable explanations for a particular observation depending on the underlying semi-lattice structure. For instance, consider the framework $\mathcal{F} = (\{p : \alpha \leftarrow q : \gamma, p : \beta \leftarrow r : \tau\}, \{q, r\}, \{p\})$.

- If $\alpha \approx_t \beta \approx_t +$, but they are unrelated wrt $<_k$, then there are two acceptable explanations for $p : +$, namely, $\{q : \gamma\}$ and $\{r : \tau\}$.
- If $\alpha <_{tk} \beta$, then there is only one acceptable explanation for $p : +$, namely, $\{q : \gamma\}$.
- If $\alpha \not\approx_t \beta$, but they are unrelated wrt $<_k$, then there is one acceptable explanation for $p : +$ and one acceptable explanation for $p : -$.

5 Conclusion

In this paper, we extended the annotated language of [13] in various directions to obtain an enriched representation language. In particular, we permitted rule-bodies to be conjunction of literals, and the rules to be recursive. We identified a class of annotated logic programs called the stratified programs which can be given a unique supported minimal Herbrand model as their meaning. We then smoothly integrated abductive reasoning into this annotated logic framework by generalizing the work described in [18, 19]. We formalized the notion of an explanation, and specified when an explanation can be regarded as acceptable.

The literals in our language do not contain variables as annotations, in contrast with that in [12]. As a consequence, the annotation in the rule-head does not explicitly depend on the annotations in the rule-body. The semantics of $-$ in [12] is also different from our interpretation of it here. Similarly, we differ from the approach of [20] in that we do not interpret annotations probabilistically.

References

- [1] K. R. Apt, H. A. Blair and A. Walker, Towards a theory of declarative knowledge, in: Jack Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, pp. 89-148, Morgan Kaufman, 1987.
- [2] N. Belnap, How a computer should think, in: G. Ryle (ed.), *Contemporary Aspects of Philosophy* (Oriel Press, 1977) 30-56.
- [3] N. Belnap, A useful four-valued logic, in: J. Dunn and G. Epstein (ed.), *Modern uses of multi-valued logic* pp. 8-37, D. Reidel, 1977.
- [4] H. A. Blair and V.S. Subrahmanian, Paraconsistent logic programming, in: *Theoretical Computer Science*, **68**, pp. 135-154, 1989.
- [5] W. Chen, A general logic-based approach to structured data, *Ph.D. Dissertation*, State University of New York at Stony Brook (1990).

- [6] K. Eshghi, Abductive Planning with Event Calculus, in: *Proceeding of the Fifth International Conference on Logic Programming*, pp. 562-579, 1988.
- [7] M. Fitting, Bilattices and the semantics of logic programming, in: *Journal of Logic Programming*, 1991.
- [8] M. L. Ginsberg, Multi-valued logics: A Uniform Approach to Reasoning in Artificial Intelligence, in: *Computational Intelligence*, 4, pp. 265-316, 1988.
- [9] M. L. Ginsberg (ed.), *Readings in Nonmonotonic Reasoning*, Morgan-Kaufmann, 1987.
- [10] M. Kifer, and A. Li, On the semantics of rule based expert systems with uncertainties, in: *Proc. of ICDT*, LNCS 326, pp. 102-117, Springer-Verlag, 1988.
- [11] M. Kifer and E.L. Lozinskii, A logic for reasoning with inconsistency, in: *Journal of Automated Reasoning*, Kluwer Academic Publ., 1992. to appear.
- [12] M. Kifer and V.S. Subrahmanian, Theory of generalized annotated logic programming and its applications, in: *Journal of Logic Programming* (1992). to appear.
- [13] Krishnaprasad Thirunarayan and Michael Kifer, A theory of nonmonotonic inheritance based on annotated logic, *Artificial Intelligence* (1992). to appear. (Also: An evidence-based framework for a theory of inheritance, in: *Proc. of IJCAI-89*, pp. 1093-1098, 1989.)
- [14] J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988.
- [15] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufman, 1988.
- [16] Y. Peng and J. A. Reggia, *Abductive Inference Models for Diagnostic Problem-Solving*, Springer-Verlag, 1990.
- [17] D. Poole, A logical framework for default reasoning, in: *Artificial Intelligence*, 36, pp. 27-47, 1988.
- [18] D. Poole, Explanation and prediction: an architecture for default and abductive reasoning, in: *Computational Intelligence*, vol. 5, pp. 97-110, 1989.
- [19] D. Poole, A methodology for using a default and abductive reasoning system, in: *International Journal of Intelligent Systems*, vol. 5, (1990) 521-548.
- [20] D. Poole, Representing diagnostic knowledge for probabilistic Horn abduction, in: *Proc. of IJCAI-91*, vol. 2, pp. 1129-1135, 1991.
- [21] T.C. Przymusiński, On the declarative semantics of deductive databases and logic programs, in: J. Minker (ed), *Foundations of Deductive Databases and Logic Programming*, pp. 193-216, Morgan Kaufmann, 1988.
- [22] R. Reiter, A Theory of Diagnosis from First Principles, in: *Artificial Intelligence*, 32, pp. 57-090, 1987.
- [23] M. Shanahan, Prediction is Deduction but Explanation is Abduction, in: *Proc. of IJCAI-89*, vol. 2, pp. 1055-1060, 1989.
- [24] R. Thomason, J. Horty, and D. Touretzky, A calculus for inheritance in monotonic semantic nets, in: Z. Ras and M. Zemankova (eds.), *Proc. of ISMIS-87* pp. 280-287, North Holland, 1987.
- [25] R. Thomason and J. Horty, Logics for inheritance theory, in: M. Reinfrank, J. de Kleer, M. Ginsberg, and E. Sandewall (eds.), *Non-Monotonic Reasoning*, Springer-Verlag, 1989.
- [26] P. Torasso and L. Console, *Diagnostic Problem Solving*, Van Nostrand, 1989.