RC 20123 (06/30/95)
Computer Science

IBM Research Report

Conflict Resolution in Advice Taking and Instruction for Learning Agents

Benjamin N. Grosof

IBM Research Division
T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598
(914) 784-7783 direct -7455 fax -7100 main
Internet: grosof@watson.ibm.com (alt.: grosof@cs.stanford.edu)
World Wide Web: http://www.watson.ibm.com

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

This is an **extended version of the paper appearing** *in: Proceedings of the ML-95 Workshop on Agents that Learn From Other Agents, which is available on World Wide Web* http://www.cs.wisc.edu/~shavlik/ml95w1/ . *Editor: Diana Gordon gordon@aic.nrl.navy.mil . Workshop held in conjunction with the Twelfth International Conference on Machine Learning, Tahoe City, CA, July 9, 1995. This, the extended version, is available as an IBM Research Report on the World Wide Web http://www.watson.ibm.com .*

Abstract

We raise and discuss several issues of advice taking and instruction, including: the challenges of very small samples of opinion, and of assimilation with a large, complex prior knowledge base. We focus especially on the problem of conflicting yes/no rules. We observe the availability of natural kinds of information, e.g., authority, reliability, freshness, and specificity, as the basis for reasoning about precedence. By precedence we mean in the sense of resolving conflicts between rules on the basis of qualitative ordinal information.

We propose an approach to this problem of conflict: via defaults and reasoning about such precedence, drawn from knowledge representation and commonsense reasoning. We initially developed this approach in previous work. Here, we elaborate it, abstract it to a more conceptual level, and present it for a machine learning audience. Also, we abstract it away from the details of, and dependency on, one particular non-monotonic logical formalism.

We see the value of this approach not as being all-embracing, but as providing a step towards elements of future approaches to advice-taking and instruction.

1 Introduction

1.1 The Importance and Topicality of Advice-Taking and Instruction

To date, a large amount of attention in machine learning has been given to techniques for inductive learning, especially with little or no prior knowledge. There are now a collection of powerful techniques for such learning. Much less attention has been given to techniques for learning from instruction (beyond simply giving or classifying examples) or advice taking. Yet, for symbolic information, people learn mostly from reading, listening, talking, etc.. This is usually vastly more cost-efficient than learning “the hard way”, i.e., from direct observation, experience, or experimentation. Indeed, this advantage is what has made possible cultural and technological progress, over the course of human history.

Until recently, there have not been a great deal of practical applications of instruction (in the above sense) or advice-taking between multiple agents. Rubber is hitting the road at last in networked information applications, where large numbers of agents, acting on behalf of large numbers of people, are beginning to be realized. Here, agents are in a position where they can and must learn by communicating with each other. Economic and social systems to create incentives etc. for such learning by communication are beginning to be created as well.

1.2 Overview of Paper

Outline: (Recall the Abstract.) We begin by raising and discussing some of the problematic issues in advice-taking and instruction (sections 2–3), oriented towards the context of practical, industrial prototyping efforts for intelligent agents (section 4). Then we propose an approach (sections 5–6), illustrate it with an example (section 7), and wind up with discussion (section 8).

The subject of advice-taking and instruction involves more than one research community. Besides machine learning, knowledge representation and common-sense reasoning are involved. The work we present here draws on all three of these areas, while the presentation is formulated especially for a machine learning audience.

2 Challenges

2.1 Assimilation with Prior Knowledge

From the machine learning viewpoint, a distinguishing characteristic of learning by advice-taking or instruction is its close relationship to learning in the presence of a **large, complex prior knowledge** base. This is particularly important now that practical intelligent agents are being realized that are equipped with substantial knowledge bases.

The relationship to learning with prior knowledge is bidirectional. On the one hand, advice-taking and instruction provide a conduit for agents to acquire substantial, and perhaps expressively complex, knowledge bases. On the other hand, substantial knowledge bases are typically necessary for agents to usefully exploit the advice and instruction they receive.

A major challenge for an agent taking advice or instruction is to **assimilate** the advice or instruction into its own beliefs and mental state, in a rationalized and coherent manner.

2.2 Importance of Yes/No Rules and Facts as Knowledge Representation

An important form of knowledge representation is **rules plus facts** of the kind expressible in monotonic first-order logic, perhaps augmented by basic non-monotonic mechanisms, negation-as-failure cf. logic programming (e.g., Prolog) and simple inheritance with exceptions (e.g., frame-based systems). Such **yes/no** (as opposed to probabilistic or fuzzy) rules and facts have been widely used in knowledge-based systems generally, not just in intelligent agents. They have been found useful as well in logic programming languages. Their advantages compared with probabilistic or fuzzy representations include relative conceptual simplicity, as well as relative computational simplicity.

2.3 Learning from Very Small Samples of Opinion

One of the hard-won lessons of applied AI is that knowledge acquisition is often difficult. An implication is that agents should often be designed to make the most of what advice they can obtain.

Humans often use knowledge on matters about which they have received only a very small number of opinions (pieces of advice): e.g., 1 or 2 or 3. This is important especially in realms where direct experience (exemplars) is not available, or is costly. Humans often act when they have received only one piece of advice, e.g., in an office organization or other social situation with a fair degree of trust. In such situations of very small "samples", most inductive learning methods, which are typically statistical in flavor, do not have much help (or confidence) to offer. The work of [Maes, 1994] [Lashkari *et al.*, 1994] is an interesting approach, viewable as advice-taking, which exploits a community of agents via communicating example sets. Nearest-neighbor inductive learning then becomes the basis for the advice-taker's beliefs. However, a drawback is that this kind of advice-taking requires, in effect, large samples, and/or for a great deal of knowledge to be encoded in the distance function (i.e., similarity measure).

An implication of the foregoing is that we should enable agents to cope with situations of reasoning and acting where they cannot presume that they will get lots of experiential feedback to furnish a basis for inductive-style learning. Rather, in many situations, an agent should mainly trust what information it manages to learn from communication.

3 Problems: Conflict

3.1 Advice Taking

With rules form of knowledge (in the sense discussed earlier), an important aspect of assimilation is handling **conflict** between the rules (and facts) that are received as various pieces of advice or instruction. Sources of information might include messages from other agents (e.g., humans), reading texts, etc..

We expect that a basic feature of agent life, as of human life, is that one cannot believe everything that one is told. Not only may advice be incorrect; e.g., it may contradicted by direct experience. Worse, different sources of advice information may contradict each other, or even themselves. How is an advice-taking agent to maintain a consistent, yet usefully actionable, set of beliefs, then?

In human life, such conflict (disagreement) occurs often. For example, I once got conflicting advice from two bureaucrats in my company about how to get the light bulb repaired in my office. As another example, the U.S. Presidents have often gotten conflicting advice from their immediate staffs about which issues deserved attention, or which political maneuvers would be most expedient.

3.2 Instruction and Authoring

Instruction can be viewed as how to assimilate one or more pieces of advice from a single source agent into the beliefs of a learning recipient agent. Often these different pieces of advice are received at different times. An interesting pragmatic first step towards this problem is when the single source agent is a *human*, e.g., a user communicating through a user interface. Like an automatic agent, a human source may have only a rough knowledge of what language, ontology, commonsense/background/context knowledge, and interface format are appropriate to communicate to the recipient agent. E.g., the human may be a "lay" (non-programmer) user trying to instruct the recipient agent to act on her behalf.

When the source agent is a human, instruction can thus be viewed as *authoring*, i.e., the incremental specification of a belief set (i.e., creation of a knowledge base) that the recipient agent should hold.

A key, well-known problem in knowledge-based system development, generally, is the “knowledge acquisition bottleneck”. In rule-based systems, for example, even for expert programmers it is difficult to specify, test, and debug a set of rules; and much more difficult for lay users to do such authoring. A major reason for such difficulty is the potential for *conflicting* interaction between rules within a group of rules. A common situation is that a special-case rule may contradict a more-general-case rule.

For example, consider an application of e-mail filtering for importance. I might tell my Mailbot “if the message is from a store, then it’s low importance” (rationale: it’s probably useless junk mail). And I might tell it further that “if the message is from a store from which I’m awaiting an order, then it’s high importance” (rationale: it’s probably about my order and my order is probably important). These two rules conflict in the case of a message from a store from which I’m awaiting an order. The second rule is more specific. Commonsense style of instruction suggests that the second rule should win the conflict. This basis for precedence (winning) between rules is known as **specificity**: the antecedent of the second rule is a condition strictly subsumed by (i.e., more specific than) the antecedent of the first rule. This kind of implicit presumption of specficity precedence has been studied extensively in the non-monotonic reasoning community, e.g., [Touretzky, 1986].

In developing rule bases, this potential for conflict partly arises after knowledge is encoded, due to inference and representation mechanisms that involve logical non-monotonicity, e.g., the use of negation-as-failure and inheritance with exceptions. However, even beyond such programming mechanisms, this potential for conflict also arises during the process of encoding knowledge, due to unforeseen contradictions between “draft” rules. In my view, this is because humans often think in a manner that, in effect, involves logical non-monotonicity: e.g., often special-case rules are *implicitly* regarded as overriding more-general-case rules. Either way (during or after encoding knowledge), such logical non-monotonicity causes non-modularity and complexity of the behavior of incremental changes to the set of rules.

This non-modularity and difficulty of incremental development can be viewed as a difficulty in assimilating instruction, i.e., in assimilating pieces of advice from a single source. (Actually, as a practical matter, knowledge-based systems are often developed from the interleaved contributions of *several* human sources of knowledge, i.e., of programming or domain expertise. These sources sometimes conflict with each other.)

4 Practical Applications Context

The foregoing discussion of challenges and problems is motivated in considerable part by the context of our experiences and aims in building practical intelligent agents at IBM. We are pursuing several projects in intelligent agents architecture and applications. Next, we describe them and their relation to advice taking and instruction.

4.1 Instructing Personal Rule-Based Agents for Network Information Retrieval and Handling

RAISE, mnemonic for Reusable Agent Intelligence Software Environment, is a C++ class library, designed from the ground up to be object-oriented, that provides reasoning, communications, and learning smarts for intelligent agents. RAISE is currently in prototype. Its first phase includes rule-based inferencing and procedural attachments, plus facilities for authoring (instruction) and communication.

Globenet [Grosof and Foulger, 1995] is a pilot application of RAISE. Globenet is a system for retrieval and handling of newsgroup information, oriented initially to customer service support. At Globenet's heart are intelligent agents that lay (non-programmer) users instruct with rules. Instruction is given through a graphical user interface (GUI), with menus and a flexible forms-based approach to specifying conditions, boolean combinations, and consequence actions. A user's rule base controls the retrieval and handling performed on behalf of that user. Globenet is already deployed (since fall 1993) in IBM's OS/2 customer service support organization, where resulting major productivity improvements of over 30% are reported in early empirical experience. More recently, a new version of Globenet has been enhanced by RAISE.

The first prototypes of RAISE, and of Globenet+RAISE, were demonstrated publicly at the IBM T.J. Watson Research Computer Science Expo '95 (on June 22, 1995).

Globenet+RAISE is an example where instruction is a crucial issue. The whole point is for each lay user to be able to personalize his agent's behavior. We see this kind of application of intelligent agents, i.e., for personalized network information retrieval and handling, as a widely important one in the future, including for electronic commerce and digital libraries.

RAISE is oriented towards facilitating instruction by lay users, i.e., towards facilitating their specification of rule sets. In this regard, we see three aspects as especially important:

1. Conflict handling. This motivates the approach in this paper.

2. Graphical UI. IBM's currently available Workgroup Agent product[1] contains an interesting graphical UI approach, more advanced than that in Globenet.

3. Support for specification via natural language (i.e., a goodly fragment).

4.2 Itinerant Agents and Inter-Agent Communication

Itinerant Agent framework (Agent Meeting Places) [Chess *et al.*, 1995] is a third project, currently in early prototype, which also uses RAISE. By itinerant, we mean that an agent moves its execution locus between network nodes. In this work, we are emphasizing practical issues in communications and security for intelligent agents, with open (as opposed to proprietary) architecture for the framework. The thrust of the project is to help realize the vision of a network full of interoperating agents that communicate at the knowledge level, including taking and giving advice, often as part of economic activities.

5 Approach: Advice as Defaults with Prioritization-type Precedence

5.1 Our Focus: Rule Beliefs and Conflict Handling

Next, we discuss our approach to handling and resolving conflicts in advice taking and instruction.

We address all the challenges and problems discussed earlier: assimilation with substantial prior knowledge, advice in the form of yes/no rules as well as facts, and the presence of conflict.

Methodologically, as a first step, we restrict our focus to learning beliefs rather than plans or goals.

5.2 Inspiration: Human Tactics in Advice Taking

Our approach is based on employing the knowledge representation tools of defaults with prioritization-type precedence. It is inspired by our observation of two tactics people employ.

[1] trademarked; formerly called IntelliAgent (trademarked also)

First, humans often employ the tactic of treating advice sources as credible, but allowing advice information to be defeasible (retractable). That is, they treat advice as working belief, which can be overridden by other advice and direct experience. This suggests the first element of our approach: to **represent advice as default-status premise belief**.

Second, when faced by contradictory conflicts between two (or more) pieces of advice, humans often employ the further tactic of resolving the conflict definitively in favor of the advice that arises from the *source* that has greater precedence. The structure of such precedence often has a qualitative, ordinal flavor, similar to that of the concepts of prioritization-type precedence developed in non-monotonic logical formalisms.

Indeed, some beliefs are about **which sources to believe more** on what subjects in which circumstances. These beliefs themselves may arise from taking advice from multiple, sometimes-conflicting sources. Such beliefs about precedence are thus not only important but also **defeasible**.

All this suggests the second element of our approach: to **represent and reason defeasibly about prioritization-type precedence, including about sources**.

See Appendix A1 for a review of defaults and prioritization-type precedence.

5.3 Bases for Precedence Among Advice Defaults

There are several natural kinds of available information that can furnish the basis for such precedence orderings between pieces of advice or instruction. These bases for such precedence include:

- specificity,
- freshness,
- authority,
- reliability,

(or, more generally, some other property). That is, a precedence ordering among advice is derived from a precedence ordering among sources, which in turn is derived from properties of sources such as authority, reliability, and freshness. More generally, the precedence among sources may also be relative to the subject area of the advice, or to the situations, e.g., times, in which advice is offered or to which it applies.

By **specificity**, we mean the sense we discussed earlier (section 3): instruction or advice which covers a more-specific case takes precedence over that which covers a more-general case.

Specificity is especially relevant and available as a basis for precedence between two pieces of advice taken from the same source, e.g., in instruction.

By **freshness**, we mean the recency of the advice and/or the information upon which the advising source bases that advice. E.g., in classical database updating, there is typically a presumption that more recent updates override previous information. E.g., advice received today from one's broker to buy a particular stock is typically given precedence over advice received from her last month to sell that stock.

Like specificity, freshness is especially relevant and available as a basis for precedence between two pieces of advice taken from the same source, e.g., in instruction. But it applies between sources as well. E.g., advice from a credible stock broker received today is typically given precedence over advice from a credible stock broker received three years ago.

By **authority**, we mean in the legal and organizational senses. For example, federal law takes precedence over state law; directives from the head of an organization take precedence over those from subordinates.

By **reliability**, we mean in the sense of accuracy or likelihood of correctness. For example, I believe the New York Times newspaper to be a more reliable source than the New York Post and the Village Voice newspapers, which in turn I beleve to be more reliable than the National Enquirer newspaper. For example, Mikey the elementary school student believes his parents to be more reliable sources than his schoolfriends and his schoolteachers, and believes those to be more reliable in turn than his schoolenemies.

Reliability may be, in turn, based on (inferred from information about) **expertise**, **judgment**, etc..

Authority and reliability are especially relevant and available as bases for precedence between different sources.

5.4 Essence of Approach

In the advice-taker:

1. Represent advice to be the content of utterance information from a credible source. A source which is adequately honest and competent is usefully trustworthy.

2. Represent each piece of advice as a default premise belief. Each default premise belief has an associated label.

3. Resolve conflicts between advice using precedence between the defaults.

4. Represent precedence explicitly, so that it can be reasoned about automatically by the advice-taking agent.

5. Represent sources explicitly, including their relationship to the advice they give, e.g., to the labels of the associated defaults.

6. Represent the bases for precedence explicitly, e.g., information about the authority, reliability, freshness, and specificity of sources (and thus their advice) relative to subjects and situations / times.

   Likewise, one can extend this to represent the bases for credibility explicitly, e.g., information about the honesty and competence of sources (and thus their advice). However, credibility is less of an immediate issue for handling conflict than precedence is.

Our approach has several attractive features. By treating each piece of advice as a premise default, any piece involved in conflict can be overridden as a conclusion. Overall consistency is preserved when conflict arises, whether or not it is definitively resolved. Information of kinds that are available and natural for knowledge acquisition is exploited as the basis for resolving conflicts.

5.5 Formulation in DAP Circumscription

In previous work, we began investigation and development of our approach to advice-taking. [Grosof, 1993b] [Grosof, 1993a] There, we employed a variant of circumscription as the non-monotonic logical formalism. We invented this formalism, Defeasible Axiomatized Policy (DAP) circumscription, mainly for this specific purpose. We developed the theoretical aspects of the formalism, including a sketch of inferencing algorithms [Grosof, 1993a].

This work builds upon our other previous work relating learning and non-monotonic reasoning [Grosof, 1992b] [Grosof, 1992a] [Grosof, 1993c], which is largely based on our DAP circumscription formalism, and on our new inference algorithms for prioritized default reasoning expressed in circumscription [Grosof, 1992b] [Grosof, 1995a] [Grosof, 1995b].

6 Meta-Language Formulation of our Approach

To illustrate the approach, in the next section we will illustrate it with a detailed example. Preliminary to that, in this section we give a specification notation $\mathcal{PD}$, acronymic for "Precedence plus Defaults". $\mathcal{PD}$ is a meta-language, new with this paper. We will use it to describe the example.

The $\mathcal{PD}$ notation serves two purposes. First, it is simpler, especially for the reader to follow, than the DAP circumscription formalism. Second, it simplies mapping our approach into other non-monotonic logical formalisms. It thus reduces or removes the dependence of our approach on one particular non-monotonic formalism.

In $\mathcal{PD}$, there are two kinds of premise beliefs (axioms): for-sure and default. By a for-sure premise belief, we mean a non-retractable belief of the kind familiar from first-order logic.

Each premise belief has a label as a prefix, enclosed in $\langle\ \rangle$. E.g.,

$\langle Sure \rangle \quad Small(Isaac) \supset \neg Big(Isaac)$

$\langle advice37 \rangle \quad Smiling(Isaac) \supset \neg Cranky(Isaac)$

The label of the first premise above is $Sure$. A premise is for-sure iff it is labelled by the special label $Sure$, as in the first premise above. A premise is default iff it is labelled by any label other than $Sure$, e.g., $advice37$ in the second premise above. Each default premise's label is required to be unique. The rest of the axiom after the label is called its *formula part*, a formula in a first-order language $\mathcal{L}$. Each label is a constant individual (object) of $\mathcal{L}$. The label is essentially a name for the premise, used in connection with the formal mechanism of precedence. A for-sure premise's formula part is required to be a closed formula (i.e., sentence). A default premise's formula part may be closed, or it may be open, i.e., include free (i.e., unquantified, unbound) variables, e.g.,

$\langle BirdSchema \rangle \quad Bird(x) \supset Flies(x)$

An open (i.e., schema) default can be viewed as the collection of all its instances. An instance of an open default is a closed default formed by instantiating (i.e., substituting a possible binding for) each of its free variables. A closed default has one instance: itself. We refer to a default instance by writing the pair $\langle label, t \rangle$, where $label$ refers to a default, and t refers to an instantiation of that default's free variables. t may be empty. E.g.,

$\langle Bird, Tweety \rangle \;\; Bird(Tweety) \supset Flies(Tweety)$

$\langle Bird, Isaac \rangle \;\; Bird(Isaac) \supset Flies(Isaac)$

$\ldots .$

$PRECEDES(label1, t, label2, u)$ means that the default instance $\langle label1, t\rangle$ has higher precedence than the default instance $\langle label2, u\rangle$. (t and u may be empty.) $PRECEDES$ is constrained to be a well-founded, strict partial order, defined over the default instances.

See Appendix A2 for details of how $\mathcal{PD}$ maps to DAP circumscription.

6.1 Defeasible Precedence

The example we give in the next section uses only non-defeasible (i.e., monotonic) reasoning about precedence. Above, we gave a version of $\mathcal{PD}$ sufficient for this expressive fragment. More generally, DAP circumscription can represent *defeasible* (i.e., non-monotonic) reasoning about precedence, as well.

DAP circumscription involves a finite tower of 1 or more meta-levels of reasoning about precedence. $\mathcal{PD}$ can be viewed as one such meta-level. By stacking together several meta-levels, each formulated in $\mathcal{PD}$, $\mathcal{PD}$ can be extended to specify DAP beyond just DAP1.

Next, we list some examples where it is useful to represent *defeasible* precedence.

In an organization, change of authorities (e.g., promotion and demotion of managers) leads to change of the precedence accorded to their recommendations or to their instructions, e.g., by subordinates in following potentially-conflicting demands from superiors.

In law, rulings concern precedence of principles or of jurisdictions, and are themselves reversible by subsequent rulings.

See [Grosof, 1993a] for more examples of advice-taking using reasoning about precedence, including defeasible reasoning about precedence.

7 Example: Repairing a Dim Bulb in a Bureaucracy

A different version of this example appeared in [Grosof, 1993b].

7.1 A True Story

Consider the following true story. Bold type in the remainder of this section indicates the most essential parts of the reasoning in the story. By "essential" here, we mean with respect to the problem addressed in this paper.

One Saturday a few years ago, one of the fluorescent overhead lights in my office at IBM suddenly stopped working. I knew that I had to make some kind of official request for the repair, but I did not know exactly who and how to ask. I thus phoned Security and asked them what was the procedure to officially request the lighting repair. The woman answering the phone told me (**Security advice**) that the procedure was simply to type the command "MAINT" to the local mainframe computer system, then obey the instructions it gave me. **I now believed that this was the correct procedure**; in general, in the past, I had found the Security people to be pretty competent, hence credible. I thus acted by following her advice and did the MAINT command: this turned out to be merely to fill out a structured e-mail message to a relevant IBM bureaucrat in charge of maintenance-type repairs. After I sent the message, the system replied automatically that my request was being forwarded first to my departmental Administrative Assistant (AA), who had to approve the request.

Monday afternoon came, and I was just wondering impatiently about my lighting repair, when I received an e-mail message from my (departmental) Administrative Assistant. In it, she said that she had *not* approved my repair request, that it was inappropriate. She said (**AA advice**) that the correct procedure for requesting the lighting repair was to phone Security and to ask them to officially request on my behalf to the building landlord (a separate, non-IBM company, named Robert Martin), to have their maintenance people come and fix it. ("Classic!", I mumbled to myself.) **I now believed that this was the correct procedure. Though it contradicted (i.e., conflicted with) what the weekend Security person had told me, I believed that my AA had more organizational authority than Security on this kind of matter** and, in the past, had generally also been competent, hence credible (at least as much as Security). **Hence I resolved the conflict definitively (for the time being) in the AA's favor (more precisely, in favor of her advice.) That is, I assigned greater prioritization-type precedence to the AA advice than to the Security advice.**

I thus acted by following the AA's advice. (Though Security again repeated their previous advice, I insisted that they contact the landlord.) Sure enough, within fifteen minutes a landlord maintenance person came to my office and completed the repair.

7.2 Automatable Representation using our Approach

Next, we show in $\mathcal{PD}$ how to represent the beliefs of the advice-taking agent in the above story (a simplified version, of course). This is an automatable representation. (Subsection 8.1 discusses implementation.)

Lower-case arguments indicate variables. sit stands for situation. Sec stands for Security. My_Pb stands for my particular problem situation. $Light$ stands for lighting repair. $Pesky(subj, sit)$ means that sit is a pesky problem of kind $subj$ that the bureaucracy views as needing to be solved. $Do(proced, sit)$ means that the (unique) correct bureaucratic procedure is $proced$ in situation sit. $MAINT$ stands for the procedure of performing the MAINT command on the mainframe. $Sec\_Ask\_Landlord$ stands for the procedure of asking Security to ask the landlord. $More\_Authority(source1, source2, subject)$ means that $source1$ is more authoritative than $source2$ about $subject$. $Source(a, s)$ means that s is the source of the advice a. $Subject(a, s)$ means that s is the subject of the advice a. $\approx\!\!\!\mid\; p$ indicates that the sentence p is non-monotonically concluded from the (current) premise set. $\exists! x. \ldots$ stands for "there exists a *unique* x such that ...".

Before receiving the Security advice or the AA advice, the premise set includes exactly: the uniqueness of names (this can be formulated explicitly as an included for-sure premise), plus

$$\begin{array}{l}
\langle Sure \rangle \quad \forall subj, sit.\ Pesky(subj, sit) \supset \\
\qquad \exists! proced.\ Do(proced, sit) \\
\langle Sure \rangle \quad Pesky(Light, My\_Pb) \\
\langle Sure \rangle \quad \forall advice1, advice2, source1, \\
\qquad\qquad source2, subject, sit. \\
\quad [Source(advice1, source1) \\
\quad \wedge\ Source(advice2, source2) \\
\quad \wedge\ Subject(advice1, subj) \\
\quad \wedge\ Subject(advice2, subj) \\
\quad \wedge\ More\_Authority(source1, source2, subject)] \\
\qquad\quad \supset \\
\qquad PRECEDES(advice1, sit, advice2, sit) \\
\langle Sure \rangle \quad More\_Authority(AA, Sec, Light)
\end{array}$$

The Security advice update corresponds to adding the premises:

$$\begin{array}{l}
\langle Sec\_Advice \rangle \quad Pesky(Light, sit) \\
\qquad\qquad\qquad \supset Do(MAINT, sit) \\
\langle Sure \rangle \quad Source(Sec\_Advice, Sec) \\
\langle Sure \rangle \quad Subject(Sec\_Advice, Light)
\end{array}$$

After this update, the advice-taker (non-monotonically) concludes that it should do the MAINT command on the mainframe:

$$\begin{array}{l}
\approx\!\!\!\mid\ \ Pesky(Light, My\_Pb) \\
\qquad\qquad \supset Do(MAINT, My\_Pb)
\end{array}$$

which implies

$$\approx\!\!\!\mid\ \ Do(MAINT, My\_Pb)$$

The later, AA advice update corresponds to adding the premises:

$$\langle AA\_Advice \rangle \quad Pesky(Light, sit) \\ \supset Do(Sec\_Ask\_Landlord, sit)$$

$$\langle Sure \rangle \quad Source(AA\_Advice, AA)$$

$$\langle Sure \rangle \quad Subject(AA\_Advice, Light)$$

After this update, the advice-taker concludes (monotonically) that the AA advice has higher precedence than the Security advice:

$$\models \quad \forall sit.\ PRECEDES(AA\_Advice, sit, \\ Sec\_Advice, sit)$$

As a result, it retracts the previous non-(monotonic) conclusion above, and (non-monotonically) concludes instead that it should have Security contact the landlord:

$$\models\!\!\approx \quad Pesky(Light, My\_Pb) \\ \supset Do(Sec\_Ask\_Landlord, My\_Pb)$$

which implies

$$\models\!\!\approx \quad Do(Sec\_Ask\_Landlord, My\_Pb)$$

Furthermore, the agent is able to (non-monotonically) **learn an entire rule from conflicting advice** that this is the preferred procedure in further lighting repair situations:

$$\models\!\!\approx \quad \forall sit.\ Pesky(Light, sit) \\ \supset Do(Sec\_Ask\_Landlord, sit)$$

E.g., suppose the premises are further updated with

$$\langle Sure \rangle \quad Pesky(Light, Leoras\_Pb)$$

where $Leoras\_Pb$ stands for another (similar) problem of lighting repair that colleague Leora has tomorrow. Then the agent concludes

$$\models\!\!\approx \quad Do(Sec\_Ask\_Landlord, Leoras\_Pb)$$

This learned rule is defeasible, e.g., the agent may learn exceptions to it later from further experience or advice taking.

8 Discussion and Related Work

8.1 Implementation Status

The lighting repair example in the last section illustrated our approach and showed how it can be formalized in terms of knowledge representation (beliefs and inferencing both). The results in [Grosof, 1993a], together with those in [Grosof, 1992b] and more recently in [Grosof, 1995a] [Grosof, 1995b], imply how to implement inferencing for substantial expressive fragments (i.e., subclasses) of DAP circumscription, and thus of $\mathcal{PD}$. In particular, they suffice to mechanize the lighting repair example.

In current work, we are pursuing the detailed development of more inferencing algorithms for DAP circumscription and thus $\mathcal{PD}$. However, we are also pursuing other non-monotonic formalisms as the avenue to implement the approach, including $\mathcal{PD}$. Our aim is to find a relatively conceptually simple, computationally tractable form of non-monotonic reasoning, together with associated practical (polynomial-time) algorithms. This will then be suitable for "technology transfer", as it were, from the knowledge representation community to the machine learning community, thence to experiment in building advice-taking learning agents using our approach.

8.2 Combining with Direct Experience

A motivation and feature of our approach is that it can assimilate direct experience of the agent, as well as advice or instruction. The agent's direct experience can be represented in $\mathcal{PD}$, for example, by updating with additional premises. E.g., in the lighting repair example, the agent might predict that the correct procedure for Leora's lighting repair problem is to have Security ask the Landlord ($Do(Sec\_Ask\_Landlord, Leoras\_Pb)$), but then learn from direct experience that it is not. Accordingly, the agent might update its premises to assimilate that fact, e.g., :
$\langle Sure \rangle \quad Do(Call\_Harry, Leoras\_Pb)$.

This capability of our approach is discussed, for example, in [Grosof, 1992a].

8.3 Probabilistic-Flavored Approaches

One alternative approach to advice taking is to base beliefs on (and resolve conflicts based on) statistical or weighting schemes, after lots of example facts have been accumulated, either from advice or direct experience.

Another alternative approach to advice taking is to represent advice as opinions or "evidence" in a probabilistic-flavor knowledge representation (we include fuzzy logic and Dempster-Shafer in this category). These opinions may then be fused using combination rules such as conditional independence, triangular norms (e.g., min for conjunction), or Dempster's Rule. This approach has the drawback of requiring extensive numerical information, and probabilistic-flavor (in)dependency information, to be provided by the sources of advice or instruction.

Our approach is motivated by situations where the above kinds of information are (mainly) not available. An interesting issue is how to combine the best features of the above style of approaches with ours, and/or to determine what the boundary for optimizing transition from one to the other might be.

8.4 When our Approach is most Suitable

Approaching advice by representing it in a non-monotonic logical system for defaults plus precedence has the usual advantages of being a declarative account and having expressive power.

This is especially helpful when there is significant reasoning with the knowledge, when knowledge and inferencing about the bases for resolving conflicts is itself interestingly contentful.

There are many other relevant issues we did not address here: e.g., operationalization [Gordon and Subramanian, 1993], tight integration with probabilistic-flavor knowledge and techniques, and tight integration with induction, especially from direct experience and large amounts of exemplars.

8.5 Applicability of Approach to Multiple Non-Monotonic Formalisms

Part of this paper's contribution is to elaborate upon our approach (to conflict resolution in advice-taking) in more conceptual terms, and to detach it from the particular non-monotonic formalism, DAP circumscription, in which we first formulated it.

$\mathcal{PD}$ is a meta-language, It is capable of being interpreted in (i.e., mapped into) potentially several different non-monotonic formalisms.

There are many different formalisms for defaults, and quite a few are equipped with prioritization-flavored precedence. (See [Grosof, 1992b] and [Grosof, 1995b] for a review.)

However, we are aware of only two that are capable of reasoning about the precedence, e.g., on basis of relative authority. The first one to enable such was DAP circumscription, which we developed for this kind of capability. The second one was developed independently by Brewka [1994], whose formalism, like DAP circumscription, enables reasoning about prioritization. He extends Default Logic, rather than circumscription as we do. Both DAP circumscription and Brewka's formalism need more work to achieve efficient algorithms, guarantees of well-behavior, and better conceptual understanding. As we do with our approach, Brewka remarks that his basic approach should be extensible / applicable to other non-monotonic formalisms that can express defaults and prioritization-type precedence.

8.6 Learning Vs. Knowledge Representation

Perspective: in advice-taking and instruction, the boundary between knowledge representation and machine learning (which heretofore have been regarded mainly as distinct AI communities) becomes considerably fuzzier. In some sense, communicating a piece of advice can be viewed as sending a TELL method message to a knowledge representation service object which is viewable as a black box recipient/learner. Learning is then the incorporation of the TELL's content into the recipient's knowledge base within the black box. This view of learning agent is discussed, for example, in our previous work on non-monotonic updating [Grosof, 1992b]. A similar view of this is as "belief revision" cf. [Gärdenfors, 1988] [Nebel, 1989].

A large body of work on non-monotonic reasoning can be viewed as treating the fundamental knowledge representation theory of a single agent learning from advice provided by a single instructor.

In addition, there is more recently some work on multi-agent non-monotonic reasoning that provides some theoretical basis for advice-taking from multiple conflicting sources, but more distantly than our work. E.g., [Morgenstern, 1990] treats agents reasoning non-monotonically about the beliefs of other agents that themselves reason non-monotonically. Finally, there is a whole body of work on integrating yes/no and probabilistic and non-monotonic reasoning, e.g., [Grosof, 1988] [Bacchus, 1990].

Summary of Paper: See the Abstract.

A1 Appendix: Review of Defaults and Precedence

Defaults and Qualitative Precedence Orderings Among Defaults

By default, we mean the sense used in the knowledge representation community, i.e., the following concept from non-monotonic logical formalisms. In non-monotonic formalisms, one distinguishes between a set of premise beliefs, which may grow by accumulation, and its associated set of conclusions, some of which may be retracted as that set of premises grows. A default is a premise belief, e.g., that if Tweety is a Bird then Tweety Flies. Informally, the default "goes through" and generates a corresponding conclusion belief when it is *consistent* with other premise beliefs plus inferencing principles (e.g., *modus ponens*) associated with the formalism. Exactly how the notion of consistency is formalized, and how those inferencing principles are formalized, varies among different non-monotonic formalisms.

Continuing informally, let us consider some examples (and their behavior in one particular formalism, circumscription). Suppose the premises include exactly the above bird default plus that, for-sure, Tweety is a Bird. Then the default "goes through" and Tweety Flies is concluded.

Suppose that the set of premises also includes exactly that: for-sure, Tweety is a Penguin; and, for-sure, Penguins do not Fly. Then the default does not "go through", because it is inconsistent with the other premises plus modus ponens.

Suppose the premises are exactly that: by default, if Nixon is a Quaker, then Nixon is Pacifist; and, by default, if Nixon is Republican, then Nixon is a non-Pacifist; and, for-sure, Nixon is a Pacifist; and, for-sure, Nixon is a Republican. Then each of the two defaults is individually consistent with the for-sure premises plus *modus ponens*. However, taken together, they contradict each other, i.e., conflict. There is no basis in the given premises for resolving the conflict one way or the other. Neither default "goes through"; Nixon is not concluded to be a Pacifist, nor is he concluded to be a non-Pacifist. This conservative behavior in the presence of unresolved conflict is called **skepticality**.

A relatively simple knowledge representation approach to resolving conflict between defaults generally, is to employ a qualitative precedence ordering between the defaults. For example, the first non-monotonic formalism to do so was circumscription [McCarthy, 1980] [McCarthy, 1986], which has the formal concept of **prioritization** [Lifschitz, 1985] [Grosof, 1991]. Prioritization involves a strict partial order of precedence between defaults. Prioritization is inspired by the spirit of lexicographic orderings, e.g., alphabetic orderings. Each premise default can be viewed as the preference to believe the default's associated conclusion. When two defaults conflict, if the first has higher precedence than the second, then the first "wins" and its conclusion "goes through" and is believed, while the second's is not. Vice versa, if the second has higher precedence than the first, then the second "wins". If neither default has higher precedence than the other, then neither's conclusion is believed (skeptically).

For example, suppose the premise belief set includes exactly those premises in the example above that involves Nixon's Pacifism, plus additionally that the Republican default has higher precedence than the Quaker default. Then Nixon is concluded to be a non-Pacifist.

Today, many non-monotonic formalisms are equipped with concepts of precedence partial orders between defaults that are roughly similar to prioritization in circumscription.

A2 Appendix: Mapping of $\mathcal{PD}$ to DAP Circumscription

Our meta-language notation $\mathcal{PD}$ maps straightforwardly to its **interpretation in DAP1 circumscription** (a subset of DAP circumscription), as follows.

Every $\mathcal{PD}$ premise of the form

$\langle Sure \rangle \quad B$

where B is a formula, is mapped into the DAP1 base axiom

B

Every other $\mathcal{PD}$ premise, having the form

$\langle i \rangle \quad D[x]$

(that is, any premise with label other than $Sure$), where D is a formula with (possibly empty) tuple of free variables x, is mapped into the pair of DAP1 base axioms

$\forall x. \neg abi(x) \ \equiv \ D[x]$

$\forall x. N1('abi, x)$

Here, the predicate symbol abi, and its (0-ary function symbol) name object $'abi$, are each introduced as new (unique) symbols into the first-order language.

As a final step, every $PRECEDES(i, t, j, u)$ atom (i.e., primitive subformula) is mapped isomorphically to the atom

$R1('abi, t, 'j, u)$

Here, i and j are labels, and t and u are tuples of terms.

The first-order language in the DAP1 interpretation thus differs from that in the $\mathcal{PD}$ premise set, as follows. It does not include the predicate symbol $PRECEDES$. It does include the predicate symbols $R1$ and $N1$, the abi's predicate symbols, and the 0-ary function symbols $'abi$'s.

References

[Bacchus, 1990] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge.* MIT Press, Cambridge, Mass., 1990.

[Brewka, 1994] Gerhard Brewka. Reasoning about priorities in default logic. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 940–945, Menlo Park, CA / Cambridge, MA, 1994. AAAI Press / MIT Press.

[Chess *et al.*, 1995] David Chess, Benjamin Grosof, Colin Harrison, David Levine, Colin Parris, and Gene Tsudik. Itinerant agents for mobile computing. *IEEE Personal Communications Magazine*, October 1995. To appear. Sequence listed of authors is alphabetic. Preliminary version is available as IBM Research Report RC20010 (March 1995). IBM T.J. Watson Research Center, Hawthorne, NY 10532. World Wide Web home page http://www.watson.ibm.com .

[Gärdenfors, 1988] Peter Gärdenfors. *Knowledge In Flux: Modeling the Dynamics of Epistemic States.* MIT Press, Bradford Books, Cambridge, Mass., 1988.

[Gordon and Subramanian, 1993] Diana Gordon and Devika Subramanian. A multistrategy learning scheme for agent knowledge acquisition. *Informatica*, 17:331–346, 1993.

[Grosof and Foulger, 1995] Benjamin N. Grosof and Davis A. Foulger. Globenet and raise: Intelligent agents for networked newsgroups and customer service support. In *Proceedings of the 1995 AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, Menlo Park, CA; World Wide Web http://www.aaai.org, November 1995. American Association for Artificial Intelligence. Proceedings available through AAAI on the World Wide Web. Symposium to be held at MIT, Cambridge, MA. Also to be available as IBM Research Report. IBM T.J. Watson Research Center, Hawthorne, NY 10532. World Wide Web home page http://www.watson.ibm.com .

[Grosof, 1988] Benjamin N. Grosof. Non-monotonicity in probabilistic reasoning. In J. Lemmer and L. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 237–249. Elsevier Science Publishers, 1988. Volume containing revised versions of papers appearing in Proceedings of the Second International Workshop on Uncertainty in Artificial Intelligence, held Philadelphia, PA, August 1986.

[Grosof, 1991] Benjamin N. Grosof. Generalizing prioritization. In *Proceedings of the Second International Conference on Principle of Knowledge Representation and Reasoning*, pages 289–300, April 1991. Also available as IBM Research Report RC15605, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598.

[Grosof, 1992a] Benjamin N. Grosof. Representing and reasoning with defaults for learning agents. In Diana Gordon, editor, *Proceedings of the ML92 Workshop on Biases in Inductive Learning.* Proceedings Available from the Workshop Chair, Diana Gordon: Naval Research Laboratory, Washington, D.C. 20375., 1992. Held July 4, 1992 in conjunction with the ML92 Machine Learning Conference, Aberdeen, Scotland.

[Grosof, 1992b] Benjamin N. Grosof. *Updating and Structure in Non-Monotonic Theories.* PhD thesis, Computer Science Dept., Stanford University, Stanford, California 94305, October 1992. Published by University Microfilms, Inc.. Also available as Research Report from Stanford University Computer Science Dept. and/or IBM T.J. Watson Research Center.

[Grosof, 1993a] Benjamin N. Grosof. Defeasible and pointwise prioritization: Preliminary report. Technical report, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598. World Wide Web http://www.watson.ibm.com ., January 1993. Revised from Working Paper of April 1992. Further revised version soon available as IBM Research Report.

[Grosof, 1993b] Benjamin N. Grosof. Prioritizing multiple, contradictory sources in common-sense learning by being told; or, advice-taker meets bureaucracy. In Leora Morgenstern, editor, *Proceedings of the Second Symposium on Logical Formalizations of Common-Sense Reasoning (Common-Sense '93)*, IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, 1993. Available as an IBM Research Report. Copies of the Proceedings are available via the editor (wider publication being arranged). Held Guest Quarters Hotel, Austin, Texas, Jan. 11-13, 1993.

[Grosof, 1993c] Benjamin N. Grosof. Relationships between non-monotonic reasoning and incremental learning. In Antoine Cornuejols, editor, *Proceedings of the 1993 AAAI Spring Symposium on Training Issues in Incremental Learning.* Copies of the proceedings available from the editor. Paper to be available as an IBM Research Report., 1993. Held Stanford, CA, March 23–25, 1993.

[Grosof, 1995a] Benjamin N. Grosof. Implementing prioritized defaults and specificity by transforming to parallel. In *Proceedings of the IJCAI-95 Workshop on Applications and Implementations of Nonmonotonic Reasoning Systems*, August 1995. Longer version with proof soon to be available as IBM Research Report.

[Grosof, 1995b] Benjamin N. Grosof. Transforming prioritized defaults and specificity into parallel defaults. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, August 1995.

Morgan Kaufmann. Longer version with proof soon to be available as IBM Research Report.

[Lashkari *et al.*, 1994] Yezdi Lashkari, Max Metral, and Pattie Maes. Collaborative interface agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 444–449, Menlo Park, CA / Cambridge, MA, 1994. AAAI Press / MIT Press. Held Seattle, WA, Aug. 1994.

[Lifschitz, 1985] V. Lifschitz. Computing circumscription. In *Proceedings IJCAI-85*, pages 121–127, Los Angeles, CA, 1985.

[Maes, 1994] Pattie Maes. Social interface agents: Acquiring competence by learning from users and other agents. In Oren Etzioni, editor, *Working Notes of the 1994 AAAI Spring Symposium on Software Agents*, Menlo Park, CA; http://www.aaai.org, 1994. American Association for Artificial Intelligence. Working Notes are available as a AAAI Technical Report.

[McCarthy, 1980] J. McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[McCarthy, 1986] J. McCarthy. Applications of circumscription to formalizing commonsense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[Morgenstern, 1990] Leora Morgenstern. A formal theory of multiple agent non-montonic logics. In *Proceedings of AAAI-90*, San Francisco, California, 1990. Morgan Kaufmann.

[Nebel, 1989] Bernhard Nebel. A knowledge-level analysis of belief revision. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 301–311, San Francisco, CA, 1989. Morgan Kaufmann. Held Toronto, Canada.

[Touretzky, 1986] D. Touretzky. *The Mathematics of Inheritance Systems.* Pitman, London, 1986.