HYPO: A Precedent-Based Legal Reasoner

Edwina L. Rissland

1987

COINS Technical Report 88-59

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003

Acknowledgements: This research supported by the Advanced Research Projects Agency of the Department of Defense contract no. N00014-84-K-0017; and by the Office of Naval Research, under a University Research Initiative Grant, Contract #N00014-86-K-0764.

HYPO: A Precedent-Based Legal Reasoner [1]

Edwina L. Rissland and Kevin D. Ashley [2]
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

Telephone: (413) 545-0332

Abstract

In this paper we discuss several key aspects of case-based reasoning ("CBR") and describe how these are handled in our *HYPO* program which performs legal reasoning in the domain of trade secret law. In particular, we examine the following aspects of HYPO: (1) the structure of a case knowledge base ("CKB"); (2) an indexing scheme ("dimensions") for retrieval of relevant cases from the CKB; (3) techniques for analyzing a current fact situation; (4) techniques for interpreting and assessing the relevancy of past cases by "positioning" the current fact situation with respect to relevant existing cases in the CKB as seen from the viewpoint of the case at hand and finding the most-on-point cases; (5) techniques for manipulating cases (e.g., citing, distinguishing, hybridizing); (6) techniques for perturbing the current fact situation to generate hypotheticals that test sensitivity of the current facts to changes, particularly with regard to potentially adverse effects of new damaging facts coming to light and existing favorable ones being discredited; and (7) the use of "3-ply" arguments to play out an argument. We then present an extended example of HYPO in action on a sample trade secrets case.

---

[1] This work supported in part by: the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract no. N00014-84-K-0017; the University Research Initiative, award no. N00014-86-K-0764; and an IBM Graduate Student Fellowship.

1. Introduction

In many domains, much of the interesting and important reasoning involves the manipulations of cases. Anglo-American law with its methods of *stare decisis*, or the doctrine of case precedent, is an example, par excellence, of such a domain. Of course, from the point of view of jurisprudence, *stare decisis* is rather underdefined [Llewellyn, 1933]. Nonetheless, as defined by the practice of judges, lawyers, and other experts, the core of precedent-based reasoning is reasoning – particularly, justifying an analysis – with past cases [Levi, 1949; Radin, 1933]; this includes, of course, using hypotheticals to limit, challenge, and focus the reasoning.

Strategic planning, philosophical inquiry, and historical political analyses are other good examples of case-based reasoning [Neustadt and May, 1986]. Even in domains like mathematics, where the primary method of justification is not case-based (that is, one does not justify a conclusion by citing cases but rather through the methods of logical inference), cases – here usually called examples or instances – can play a central role. In certain stages of a legal or scientific field's development, cases/examples can motivate critical change by illustrating dissonance between observations and the predictions a theory allows or the lack of precision it affords [Kuhn, 1970; Lakatos, 1976]. They are central tools of scientific discovery and pedagogy [Polya, 1973; Lenat, 1977]. Thus in certain fields like mathematics, reasoning with cases deserves at least a complementary place to other modes of reasoning, like deductive inference. In some fields like the common law, it is the primary mode of reasoning.

Note that even though other aspects of law – like statute- or code-based law – might seem to use more definitional and rule-based reasoning, it almost never obtains that one escapes the necessity to use cases to deal with gaps, ambiguities, inconsistencies, and gray areas in drafting or interpretation [Kennedy, 1980; Levi, 1949]. Whenever there is an opportunity for interpretation, especially of "open-textured" concepts, there is a need for using cases. Thus case-based reasoning has import not only for Anglo-American common law but also for statutory law and even civil law like that used in most European countries. Case-based reasoning is relevant to rule-based reasoning whenever there is room for interpretation, especially of the concepts used to form the rules.

In the research reported here, we concentrate on the use of cases, in and of themselves. We use as a paradigmatic example of case-based reasoning ("CBR"), the legal domain. By examining legal reasoning, one can clearly see key ingredients of CBR. These we have incorporated in our system, called *HYPO*, which models legal reasoning with cases, both actual and hypothetical ("hypos"). In legal reasoning and HYPO, primary elements of reasoning about a new case include:

1. **statement** of a problem situation, (i.e., the current fact situation or "cfs");
2. **analysis** of the cfs;
3. **retrieval** of relevant existing cases from a Case-Knowledge-Base ("CKB");
4. "**positioning**" the current fact situation with respect to relevant existing cases in the CKB as seen from the viewpoint of the case at hand and finding the **most-on-point** cases;

5. **heuristic (hypothetical) variation** of the cfs and attendant analyses;

6. **argument** formulation, experimentation, evaluation and revision;

7. **justification** of analysis and argument in terms of cases;

8. **explanation** of the analysis and argument in terms of cases and hypotheticals that illustrate the importance of various points.

Many of these ingredient tasks of CBR can be formulated as problems in search. For instance, step 5 – exploration with hypotheticals – can be viewed as a heuristic search of the space of all fact situations (which includes all the cases stored in the Case-Knowledge-Base plus all the legally possible hypotheticals) [Rissland and Ashley, 1986]. The amount of search done, for instance in step 3 – retrieval – can be dramatically controlled, of course, by representation techniques, especially domain-appropriate indexing schemes.

In fact, indexing can be considered one of the key problems to CBR and the development of indexing schemes, paramount. This implies, of course, that learning (of indices as well as cases) is ultimately a critical aspect of CBR. Aspects of CBR are also important to learning; for instance, the generation of hypotheticals in step 5 has import for the problem of intelligently selecting training instances for a learning program [Buchanan *et al.*, 1987; Rissland, 1987] In both learning and CBR, picking cases wisely can greatly influence the outcome of events.

Case-based reasoning can be viewed as a special class of more general example-based reasoning (*EBR*). However they are not the same. It is true that in both CBR and EBR, cases/examples are retrieved, generated, analyzed, modified and otherwise manipulated, and used for explanation and learning [Rissland, 1981; Bareiss *et al.*, 1987]. However, in CBR they are the foundation of *justification*. That is, to justify an analysis or argument in CBR, ***one cites and reasons with cases***. In particular, for the cases that support the analysis or argument, one explains the connections, often analogical, and for the troublesome cases, one distinguishes them, using legally important dissimilarities as a wedge [Ashley and Rissland, 1987b; Ashley, 1987b]. In EBR on the other hand, while examples are important components of expertise and reasoning [Rissland, 1978], they are not necessarily the basis of justification. For instance, mathematics does not recognize "proof by example" even though examples may play a central role in mathematical reasoning involved in the discovery, formulation and debugging of a proof, which ultimately must cite definitions, axioms and theorems.

2. Background

In this section, we consider other work relevant to this work on CBR: (1) memory and indexing; (2) legal reasoning; (3) hypothetical and example-based reasoning; and (4) planning and analogical reasoning. Research in other areas, such as argumentation (e.g., [Birnbaum, 1982; Birnbaum *et al.*, 1980; Cohen, 1983; McGuire *et al.*, 1981; Reichman, 1981; Reichman-Adar, 1984; Toulmin, 1958]), learning, particularly the new term problem and example-based generalization, (e.g., [Mitchell *et* *al.*, 1983; Utgoff, 1983; Rendell, 1985; De Jong and Mooney, 1986]), and explanation (e.g., [Clancey, 1983]) are also clearly relevant to the HYPO Project as a whole, but will not be addressed here.

2.1 Memory and Indexing

Recent research on memory organization, most notably by Kolodner and colleagues, bears on the problem of retrieving relevant cases through the use of indexing schemes. Kolodner in the question-answering system CYRUS used a database that reorganized its indexing scheme and representations of events as new information is added [Kolodner, 1983a; Kolodner, 1983b]. The system indexes events, event memory organization packets, or "E-mops", according to the aspects of an event that differ from the norms of the conceptual category of the event (e.g., whether they violate expectations).

Building on this memory scheme, Simpson and Kolodner developed a case-based reasoning program, MEDIATOR, that solved problems in the domain of dispute mediation [Kolodner *et al.*, 1985]. MEDIATOR's case-base contained information on physical, economic and political disputes and common mediation tactics, their failures and corrections for those failures. Cases were indexed by their features (e.g., the disputants, their goals, disputed objects), in particular, by those causing dispute mediation failure. A classic example involved a dispute between two sisters over an orange and the failure of the tactic of dividing it equally because one sister wanted the fruit and the other the rind. In connection with a physical dispute where "divide equally" fails, the case of the orange would be recalled and lead MEDIATOR to try the alternative fruit-and-rind solution.

In CYRUS and MEDIATOR, as distinguished from HYPO, the evaluation function for selecting a most on point case from the many retrieved by the "reminding" process takes into account only the closeness of fit to selected features which is determined by an *a priori* ranking of features types. In HYPO, on the other hand, the ranking of features is performed dynamically, *first,* by keeping track of how prior cases, to which various combinations of features apply more or less strongly, were treated, and *second* by determining what combinations of features apply, and how strongly, to the current facts situation, thus promoting some prior cases over others as precedents for interpreting the case at hand [Ashley, 1987c; Ashley and Rissland, 1988]. Just because a particular case is considered to be a major leading case does not necessarily mean it is the best, or most-on-point case, in the current case; it may even be inapplicable or irrelevant if the facts are different enough from the case at hand.

Thus, both HYPO and MEDIATOR contain dynamic and static aspects of indexing. In a CBR system, it would be desirable to have both MEDIATOR's ability to generate indices (e.g., based on differences, tactical failures) on the fly and HYPO's dynamic relevancy and case-ranking capabilities. Of course, such a CBR system would need be mindful of potential shortcomings of these approaches: for MEDIATOR potential trouble occurs when growth in the case-base causes the combinatorics of common and differing features to become unmanageable, and for HYPO, potential problems occur when its dimensions no longer accurately reflect the law in the case-base (e.g., because novel combinations of features have been invented). Such weaknesses are related to hard issues in knowledge acquisition and learning, like "bias" or the "new term" problem.

2.2 Legal reasoning

On-going work on AI and legal reasoning has addressed several issues of interest to our work on HYPO. (See the *Proceedings of the First International Conference on AI & Law* for a representative selection of the leading current research.) Previous models of legal reasoning have recognized the desirability of designing a program to reason with cases in order to deal with the "open-textured" meanings of legal predicates, to argue both sides of a legal issue, and to model how one case can be linked to another (often of opposite view) through a sequence of intermediate transformations.

[Gardner, 1984] developed a system to identify legal issues in the analysis of law school examination fact patterns involving the contracts law of offer and acceptance. The program primarily used if-then rules and an ATN to represent its legal knowledge of contract law. It used heuristics for distinguishing "hard" and "easy" legal issues. For instance, if any of the following obtain, the issue is considered hard: (1) there are contradictory rules; (2) the rules "run out", i.e., are inconclusive or have unresolved predicates; and (3) there are both positive and negative examplar cases to resolve whether a predicate or rule applies to the current facts; or (4) in a case when the rules have run out, the absence of any matches. While Gardner does use cases to augment the other two representations of legal knowledge, they are not the primary vehicle for the program's functioning. Further, the cases are streamlined, abstract patterns generalized from real cases – more like rules or templates – containing far fewer and much more generalized "facts" than is usual in legal reasoning. This relieves the program of dealing with facts that might be inconsistent, irrelevant, or otherwise confound reasoning with the case at hand according to the program's legal rules.

In TAXMANII, McCarty presents a knowledge representation scheme for the legal concepts involved in determining whether a particular kind of corporate distribution is taxable [McCarty and Sridharan, 1982]. A concept consists of three parts: (1) a logical template specifying necessary but not necessarily sufficient conditions; (2) a set of exemplars or cases, real or hypothetical; and (3) transformations specifying how to get from one exemplar to another (e.g., maintaining equal before and after ratios of stock owned by distributees). While the system seems to have never become fully functional – and McCarty of late has concentrated on the finer grained representation of deontic logic [McCarty, 1985] – his scheme of "prototypes and deformations" does allow interesting manipulations of cases and are most relevant to step 5 – hypothetical variation – of HYPO's reasoning process.

[Waterman and Peterson, 1981] and recently [Schlobohm and Waterman, 1987] took a classic rule-based approach to legal reasoning. The older work attempted to model how lawyers estimate the value of products liability and negligence cases; the more recent work develops mechanisms to help draft a client's will. While recognizing the problem of using rules to represent the meanings of ill-defined predicates, like "unreasonably dangerous" or "foreseeability of injury", they suggest using ever more refined rules or simply asking the user whether he thinks the predicate obtains. While the latter certainly resolves an open-textured predicate, it is not all that satisfying. As to the former – definitional backchaining – the lessons of jurisprudence, as Gardner points out, imply that it never works.

In earlier, classic work, Meldman developed a system to analyze fact situations involving intentional tort claims like assault and battery [Meldman, 1977]. The program had general rules defining the elements of a claim and more specific rules abstracted from the facts and holdings of real cases involving the claims. The latter rules were in effect abstracted examples of when individual elements of a claim were or were not satisfied. This makes Meldman's use of cases, as rule abstractions, somewhat similar to Gardner's. Also, the use of rules in Meldman (to define the elements of a claim) and Gardner (to define ingredients of contract law) are similar. While HYPO does use rule-like structures in its indexing scheme, they are not used to define elements of a claim or legal predicates, and the representation of cases encodes the particular facts of a case, thus making HYPO's representation of cases closer to what lawyers and law students capture in their squibs and case summaries.

No one to date has attempted to model the kind of adversarial analysis or argument formation that HYPO does, although McCarty is ultimately interested in using his work to do just that (e.g., comparing the positions of an opinion and the dissents) and Gardner touches on it because spotting the existence of a hard question implies the existence of an arguable point. However, none of these other efforts in legal reasoning concentrates on reasoning with cases as much as HYPO or an Anglo-American legal reasoner - lawyer, judge, professor, or law student - does.

2.3 Hypothetical and Example-Based Reasoning

Aside from [Rissland, 1981; Rissland, 1983] and the more recent work by Porter and colleagues [Bareiss *et al.*, 1987] on "PROTOS", a system that reasons with prototypical examples, there has not been that much work done on example-based reasoning ("EBR").

An underlying similarity between this work on EBR and our more recent work on CBR and hypothetical reasoning [Rissland *et al.*, 1984; Rissland and Ashley, 1986] is the use of a "space" of examples or cases - an Examples- or Case-Knowledge-Base - from which to retrieve relevant examples/cases and then manipulate and modify them. With regard to examples, the modifications are undertaken with the goal of making the retrieved example satisfy needed constraints - for instance, those needed to create a counter-example to a conjecture - the whole process is called *constrained example generation* or *CEG*. Rissland's "retrieval-plus-modification" idea in CEG and McCarty's "prototype-plus-deformation" ideas are quite similar [Rissland, 1980; McCarty, 1980]. In CEG, selection of modification technique is done in a means-ends manner, that is, modifications are indexed on the attribute they effect. In HYPO, potential modifications to a fact situation - that is, the generation of a hypothetical - are indexed via dimensions and chosen according to several heuristics (e.g., make a case weaker/stronger, enable a near-miss dimension) [Rissland and Ashley, 1986] as well as to higher level argumentation and explanation goals [Rissland, 1985]. Further investigation of indexing schemes for modification procedures (e.g., on failure) would bring this work into interesting juxtaposition to work by Hammond [Hammond, 1986a; Hammond, 1987] and Carbonell [Carbonell, 1986].

Thus, creation of hypotheticals in HYPO has certain similarities to the generation of examples in CEG: (1) both assume the existence of a knowledge base of cases (CKB) or examples (EKB), which is organized (typically in a net); (2) both use indexing schemes for retrieval of examples/cases and modification procedures; (3) both link new examples/cases (e.g., hypos) into the EKB/CKB to reflect their construction or derivation from already existing examples/cases. However, CEG was a much more simple and static system since there was no attempt in CEG, as there is in HYPO, to dynamically and case-dependently view the existing case base *modulo* the case at hand [Ashley, 1987c; Ashley and Rissland, 1987b].

2.4 Planning and Analogical Reasoning

In any CBR program there is a reliance on analogical reasoning in one form or another, the simplest of which is matching. Slightly more complex are functional/role analogies (e.g., maintaining a common numerical ratio between analogues). More complex analogies involve "mapping" of the underlying justification (e.g., proof, plan, argument). The most difficult analogies involve the purposes underlying the justifications (e.g., maximize benefit/minimize cost, maintain social equality). HYPO only deals with the first two types of analogy, although its evaluation of an argument starts to approach the third kind. See [Ashley, 1987a]. Other researches have tried to tackle the latter two.

Perhaps most ambitious is Carbonell's work on analogy in the context of planning [Carbonell, 1983b; Carbonell, 1982; Carbonell, 1983a]. A key component in this approach is examination of the underlying structure of the plan. In "derivational analogy" one maps over the underlying structure which can be thought of as the proof, plan or justification for the result. In "transformational" analogy, one examines the reasoning, particularly purposes, that lead to the underlying structure. Carbonell's work delves much deeper into the underlying structure of a case than does HYPO. For HYPO to do this, especially where existing cases in its CKB are involved, would require examination of the past analyses and arguments, the *ratio decidendi* as it is called, advanced by the opposing counsel, discussed in the briefs, opinion, and dissents, and espoused by the court deciding the case; this approaches classical legal scholarship. For cases that HYPO works through, it might be possible to keep some of this information available for future use although at this point HYPO would need significant expansion of its reasoning techniques to make use of such traces.

Kedar-Cabelli has tried to address reasoning with the purposes underlying an analogy [Kedar-Cabelli, 1984]. She suggests, for example, that the purpose of a statute can be used to select the kind of analogy to draw to show that the statute has or has not been violated. Unfortunately, it is well-known that the purposes of statutes are notoriously difficult to ascertain.

A more tractable approach is taken by [Hammond, 1986a; Hammond, 1986b; Hammond, 1987]. In his CHEF program, recipes are the cases. In the course of encountering planning failures, CHEF generalizes descriptions of combinations of features that lead to the failures in order to predict future failures before they happen. The recipes are indexed both by the gastronomical goals that they satisfy and by the problems that they avoid. As a case-based reasoner, there is a fundamental difference between CHEF and HYPO: CHEF needs a "strong" causal theory – that is, a predictive model or simulation – for purposes of explanation and credit assignment. This is an interesting limitation since CBR is often most needed in just those domains in which there is *no* strong causal theory. See [Ashley, 1987a]. Hammond, like Koldoner, can also suffer from the problem that cumulative interactions among features and his method of evaluating corrective measures may lead to situations in which the program would be better off starting from scratch rather than developing new plans which unlearn previously learned plans.

Two earlier efforts on analogy which relied on matching were Winston's and Burstein's [Winston, 1980; Burstein, 1983]. Winston's program reasoned about legal fact situations like simple assault cases. It used a matching process to decide if a base situation is analogous to a target. It attempts to place the parts of the target into correspondence with the base by matching up objects, their classes and properties, and acts and relations linking objects. The match with the highest total score is deemed to indicate the best base analogy. Burstein uses a more refined context or purpose-driven matching. His CARL program restricts the matching to information that will be useful in making a desired inference. (It is thus related to some of the newer work on example-based generalization.)

3. HYPO's Overall Architecture

Before going through an extended example of how HYPO reasons about cases, we sketch its ingredient processes and knowledge sources. See [Ashley, 1987c] for a full discussion and description.

Input to HYPO is the **current fact situation** or "cfs", entered directly into HYPO's scheme by the user who instantiates "slots" of the general "frame" and subframes used to represent a trade secret misappropriation case. [3] In its current version, HYPO has no natural language capability, although in a previous project (the "COUNSELOR" Project), HYPO was embedded in a natural language environment supporting some understanding of discourse and paragraph–long summaries. The output of HYPO is in four forms: (1) a **"case-analysis-record"** which summarizes HYPO's analysis of the input case; (2) a **"claim-lattice"** showing graphically the relationship of cases retrieved from the CKB to the case at hand; (3) **"3-ply" arguments** which play out the presentations of cases and responses of the plaintiff and defendant to them; and (4) a **"case citation summary"** according to the format of the Harvard Law School "Blue Book" [Blue Book, 1986; Ashley and Rissland, 1987a]. HYPO is currently implemented in Zeta-LISP and runs on a Texas Instruments "Explorer".

3.1 Knowledge Sources in HYPO

Domain knowledge in HYPO resides in three places: (1) the **CASE-KNOWLEDGE-BASE** - **"CKB"**; (2) the library of **dimensions**; (3) normative standards used to select best and most-on-point cases.

The CKB contains hierarchical sets of frames which describe the main components of a case including: plaintiff (π), defendant (δ), product, legal claim, prevailing party (plaintiff or defendant) and holding. Some of these (e.g., plaintiff, defendant, secret) are further expanded into frames. See [Rissland *et al.*, 1984] for an example. At this point, the Case-Knowledge-Base contains about

[3] HYPO supports a basic editing environment to input or edit cases for analysis by HYPO or for inclusion into HYPO's CKB.

30 cases, including a few classic contracts cases needed for reasoning about non-disclosure and non-compete agreements. [4]

From this basic level of representation, HYPO computes the values of **factual predicates** that state whether or not a particular legal fact is true (e.g., there-exist-disclosees, employee-has-switched-employers). Factual predicates form the building blocks of the second source of legal knowledge in HYPO, the dimensions. [5]

Dimensions encode legal knowledge expressing which clusters of facts have legal relevance for a legal claim, in particular, for arguing about a claim from a certain point of view. In effect, dimensions are compiled knowledge from the case law; they relate legally operative facts to decisions from various perspectives. [6] As their name implies, they can be thought of as a "slice" or hyperplane through a case space of high dimensionality. A key aspect of a dimension is that it encapsules the legal sense of what makes a situation better or worse, from a given perspective. For instance, one perspective of a trade secret misappropriation case concerns the disclosures made (by the plaintiff) to other parties. With respect to this dimension, *Disclose-Secrets*, the more disclosees there are, the worse off the plaintiff is *vis-à-vis* claiming his putative secret was misappropriated. Another dimension, *Employee-Sole-Developer* captures the idea that the plaintiff's position is weaker to the extent that the defendant gained access to the confidential information from plaintiff's former employee who developed the information on his own initiative and all by himself. *Competitive-Advantage* captures the idea that the plaintiff's position is stronger to the extent that the defendant's access to plaintiff's secrets saved it time and expense in developing a competing product. Other examples of dimensions can be found in [Rissland *et al.*, 1984; Rissland and Ashley, 1986; Ashley and Rissland, 1987b; Ashley, 1987c]. Currently, thirteen dimensions are implemented; we know of many more.

A dimension, itself, is a frame-like knowledge source. It has the following facets: (1) **prerequisite[s]**, which are necessary factual predicates for the dimension to apply; (2) **focal-slot[s]** which of all the prerequisites are the ones that really matter; (3) **range[s]** of values for the focal slots; (4) **direction-to-strengthen-plaintiff** which specifies how to change the focal slots; (5) **significance** which lists the claims for which the dimension has relevance; and (6) **cases-indexed** from the CKB. Note that a focal slot can be a symbolic value and that the direction-to-strengthen might specify climbing or descending a value hierarchy tree.

---

[4]Some of the cases in the CKB are *Telex Corp. v. IBM Corp.*, 510 F.2d 894 (10 Cir. 1975), *Wexler v. Greenberg*, 160 A.2d 430 (Sup. Ct. Pa. 1960), *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470 (1974), and *Dougherty v. Salt*, 227 N.Y. 200, 125 N.E. 94 (1919).

[5]There could be legal argument about whether a factual predicate is true, and thus, they can be thought of as stating a lower level legal conclusion. One can view these as facts established or assumed by a lower court or not the subject of legal debate between the parties of this dispute. Posing hypos assuming that the factual predicate is true or not true is one way to decide if it's worth trying to establish its truth.

[6]We do not compile these ourselves but rather take them from scholarly analyses and treatises like [Gilburne and Johnston, 1982; Milgrim, 1985].

3.2 HYPO's Component Modules and the Basic Processing Loop

Upon receiving a new current fact situation, the **CASE-ANALYSIS** module runs through the library of known dimensions and produces the **case-analysis-record** which lists: (1) factual predicates applicable to the current fact situation; (2) applicable dimensions; (3) near-miss dimensions (i.e., those that miss being applicable because one or two prerequisites are not satisfied); (4) potential claims (e.g., misappropriation, breach of contract); (5) names of relevant cases from the CKB. The combined list of dimensions that are applicable or near-misses with respect to the current fact situation is called the **D-list**.

On the basis of this analysis, the **FACT-GATHERER** module might come back to the user and ask for more facts (e.g., because HYPO can draw no legal conclusions or needs to resolve a factual predicate). [7]

Once no more facts are solicited from the user and a case-analysis is complete, the **CASE-POSITIONER** uses the case-analysis-record to create a claim-lattice for the current fact situation. A **claim-lattice** is a lattice such that: (1) the root is the current fact situation together with its D-list, that is, its list of applicable or near-miss dimensions; and (2) successor nodes contain (names of) cases sharing a subset, usually proper, of the dimensions listed for the root in the D-list.

The claim-lattice captures a sense of closeness to the current fact situation of cases retrieved from the CKB. Those sharing more dimensions are "nearer" to the root. Those nodes directly below the root can be considered **most-on-point** cases (**"mopc"**), for the case at hand; leaf nodes are least-on-point. Different major branches of the lattice indicate essentially different (dimensional) ways to argue the case. The claim-lattice **re-topologizes** the CKB from the viewpoint of the current fact situation; it creates neighborhoods of cases centered on the cfs [Ashley and Rissland, 1988]. Note that hypothetical cases, if they exist in the CKB, are potentially members of the claim-lattice. The claim-lattice is used by the other modules to: (1) create the basic "3-ply" skeleton of an argument; (2) to spot troublesome contrary cases; and (3) to suggest fruitful combinations of facts for new hypotheticals. This last use is particularly valuable when the CKB is sparse.

Once the claim-lattice is created, the **BEST-CASE-SELECTOR** chooses the best, most-on-point cases, to cite in support of the cfs. Criteria used include: (1) the position of the case in the lattice; (2) which side the case is good for; (3) which side the case held for; (4) whether there exists a most-on-point case for the other side which can "trump" the case.

Starting with a best case, the **3-PLY-ARGUER** constructs the skeleton of a three ply argument about the current fact situation. In a 3-ply-argument, (1) first, *side1* puts forth its best argument, cites its most-on-point cases, abstracts a holding or "rule" of the case (i.e., an abstract summary stating the relevant connection between the cfs and the cited case from which the desired conclusion follows), and perhaps distinguishes the most-on-point cases of the opponent; (2) next, *side2* forms its response to *side1*, for instance by discrediting or distinguishing *side1*'s cited case, citing a more-on-point case contrary to *side1*, or offering a hypothetical that refutes *side1*'s position or sets it up as the first step of a slippery slope argument in which *side1* ultimately fails; and (3)

[7] In a sense, fact-gathering analysis is never done (e.g., when it is clear that a case needs to be distinguished, new facts could be solicited.)

finally, *side1* rebuts *side2*.

The **HYPO-GENERATOR** uses both the case-analysis-record and the claim-lattice plus a half dozen heuristics, which use information about the case-analaysis-record and claim-lattice, to spawn legally relevant and interesting hypotheticals [Rissland and Ashley, 1986]. Among the heuristics for deciding what hypo to generate are: (1) make a case weaker or stronger; (2) make a case extreme; (3) enable a near-miss; (4) dis-able a near-win; (5) make a conflict case. These enable HYPO to pose hypotheticals where, for instance, one party of the current fact situation is made more vulnerable/invincible or two potentially conflicting ways of analyzing the cfs are pitted against each other in a "hybridized" case. In general, hypos allow a reasoner to test the robustness of the case, particularly to vulnerable or potentially damaging facts, default assumptions, etc. [Rissland, 1983].

Once HYPO has analyzed the current fact situation, faciliated exploration of legal ramifications with hypotheticals, and proposed the skeleton of an argument, HYPO's **EXPLANATION** module generates an explanation of its reasoning. Included in HYPO's explanation are case citations with illustrative hypotheticals to show the import and impact of the cited case. Hypothetical "what if's" and "for instances" are used to illustrate hard or troublesome points. For example, HYPO poses hypotheticals to show how a point or response could be strengthened with the addition of facts that would draw a closer analogy to other precedents. Higher level discussion – for instance, in terms of policy arguments, the underlying purposes of the law – is not attempted.

Finally, the **CASE-CITATION** module presents the user with the relevant cases organized by the signals (e.g., *See, But, But Cf,*) used to cite cases according the style defined in [Blue Book, 1986]. This organized citation summary provides strong hints to the user as to where the strengths and weaknesses of his case lie *vis-à-vis* what cases can be drawn on for support.

4. An Example of HYPO in Action

In this section, we walk through an example in which HYPO analyzes a hypothetical case, patterned after a real case called *Amoco Production Co. v. Lindley*, 609 P2d 733 (Okla. 1980). The processing of HYPO follows the basic CBR steps outlined in the introduction. By using a sample derived from real cases, we can compare HYPO's performance (e.g., the cases it cites) with that of the courts deciding the real ones.

4.1 Step 1: Statement of the Current Fact Situation

The current fact situation for our sample case, *AMICABLE Oil Co. v. AGGRESSIVE Oil Co.*, is as follows. The plaintiff AMICABLE has brought a claim against defendant AGGRESSIVE for misappropriation of trade secrets in connection with its STRIKER system, a computer program that analyzes drilling logs of oil wells. AMICABLE alleges that AGGRESSIVE gained access to confidential information about STRIKER through its former employee, Ian Smart. When he was first employed by AMICABLE in 1979, Ian Smart entered into an agreement not to disclose any proprietary information of AMICABLE to others. While working for AMICABLE, on his own initiative and over a period of four years, Ian Smart developed the STRIKER program. In 1984, Ian Smart left AMICABLE over a dispute about the use of the STRIKER program and subsequently took a job with AGGRESSIVE. Within ten months, AGGRESSIVE was employing an oil well log anaysis program similar to STRIKER.

4.2 Steps 2 & 3: Analysis of the Case; Retrieval from the CKB

On the basis of this current fact situation, HYPO produces the case-analysis-record in Figure 1.

Applicable Factual Predicates: exists-claimant-corp1, claimant-makes-product1, exists-claimants-info-re-product1, exists-employment-change, corp2-access-product1-via-employee1 ...
Applicable Dimensions: *Defendant-Nondisclosure-Agreement*, *Competitive-Advantage*, *Employee-Sole-Developer*
Near-Miss Dimensions: *Brought-Tools, Bribe-Employee, Vertical-Knowlege, Secrets-Disclosed-Outsiders, Security-Measures, Disclose-in-Negotiations, Nondisclose-Agreement-Specific*
Potential Claims: Trade Secrets Misappropriation (TSM), Breach of Nondisclosure Agreement (BNA)
Relevant CKB cites: *Wexler, Structural Dynamics, Eastern-Marble, Analogic, Telex, Motorola, Space Aero, ...*

Figure 1: The Case Analysis Record for *AMICABLE v. AGGRESSIVE*

To produce this case-analysis-record, the CASE-ANALYSIS module has made use of HYPO's knowledge of factual predicates and dimensions. First, the CASE-ANALYSIS module uses the list of applicable factual predicates to state what dimensions apply or are near-misses. For instance, it determines that the prerequisites for the *Defendant-Nondisclosure-Agreement* dimension are met: that two corporations, plaintiff and defendant, compete with respect to a product, plaintiff has confidential product information to which defendant has gained access, and former employees of plaintiff with knowledge of the information who now work for defendant, had entered into nondisclosure agreements with the plaintiff.

Second, dimensional knowledge allows HYPO to compare cases along a dimension: The focal slot of *Defendant-Nondisclosure-Agreement* is whether plaintiff's employees entered into agreements not to disclose and its range is a simple binary set consisting of some agreements or no agreements. To strengthen the plaintiff's position in a fact situation to which this dimension applies, add a nondisclosure agreement.

Third, the dimensions allow HYPO to find similar cases from the CKB. For instance, there are at least two cases in the CKB indexed by the *Defendant-Nondisclosure-Agreement* dimension, both of which held for the plaintiff: the *Structural Dynamics* case in which the plaintiff's extensive use of contractual protections was held sufficient to make its employees aware of the confidentiality of computer programs and *Telex v. IBM* in which the IBM employees entered into nondisclosure agreements acknowledging a listing of IBM's proprietary information. Other dimensions have more complex ranges, partially ordered sets or quantitative ranges. Knowledge about relative strength along such dimensions allows the CASE-ANALYSIS module to position the case among others on this "axis" of the case law contained in the CKB.

4.3 Step 4: Positioning with the Claim-Lattice

At this point, HYPO has retrieved relevant cases but has not considered these from the point of view of the current fact situation. The analysis, so far, only indicates where the current fact situation falls along various CKB dimensions. The CASE-POSITIONER takes these relevant cases and produces the claim-lattice which will show, *from the point of view of the current fact situation*, which cases are near and which are far according to its metric for evaluating how on-point they are. The claim-lattice for the *Amicable* case is shown in Figure 2. A much larger, extended claim-lattice, not shown, includes cases that are potentially relevant in that dimensions apply to them that are near-misses for the current fact situation.

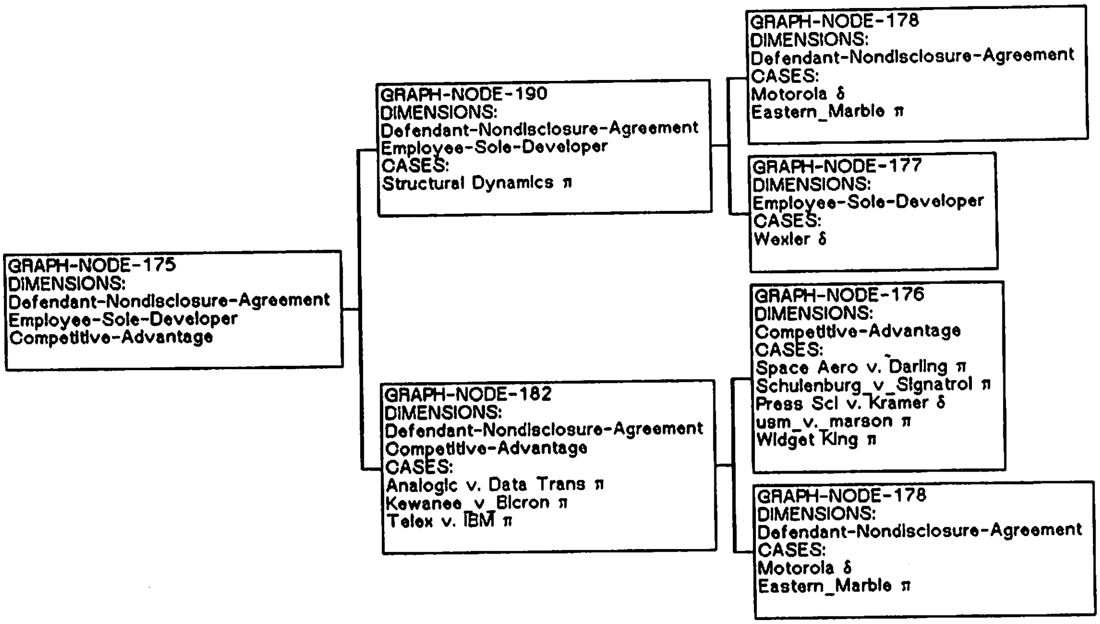

Figure 2: Claim-Lattice for the *Amicable* Case.

A summary of the results of the best case selection process is shown in Figure 3. From the claim-lattice in Figure 2, one can see that there are four most-on-point cases. These are listed in Figure 3. Also listed are a number of potential most-on-point cases which were selected from the extended claim-lattice which is not shown. The latter cases are only "potentially" most-on-point because they are indexed by dimensions which thus far are known only to be near-miss dimensions for the current fact situation. Should it come to light, through further fact gathering, that the near-miss dimensions are enabled, these cases would become most-on-point.

Most-on-point Cases:

Structural Dynamics (π; *Defendant-Nondisclosure-Agreement, Employee-Sole-Developer*)
Telex v. IBM (π; *Defendant-Nondisclosure-Agreement, Competitive-Advantage*)
Kewanee (π; *Defendant-Nondisclosure-Agreement, Competitive-Advantage*)
Analogic (π; *Defendant-Nondisclosure-Agreement, Competitive-Advantage*)

Potential Most-on-point Cases:

Midland Ross (δ; *Disclose-Secrets, Bribe-Employee*);
Automated Systems (δ; *Vertical-Knowledge, Disclose-in-Negotiations*)
Eastern-Marble (π; *Security-Measures*)
...

Figure 3: Most-On-Point Cases from the Claim-Lattice for *AMICABLE v. AGGRESSIVE*. Title of case is followed by who won and dimensions from cfs's D-List that apply to case.

Potential most-on-point cases also provide clues to the FACT-GATHERER and HYPO-GENERATOR modules, as well as the attorney using HYPO, to potential questions to be asked or hypotheticals to be considered. They are the source of "assume for the moment facts x and y and let's see what happens to our client's position" sort of reasoning.

4.4 Steps 5 & 6: Artful Hypotheticating and Other CBR Manipulations

At this point, HYPO uses its case analysis knowledge to spawn useful hypotheticals. Obvious triggers for hypos include the near-miss dimensions from the case-analysis-record and most-on-point cases and potential most-on-point cases from the claim-lattice.

One hypo that HYPO could pose to strengthen the defendant AGGRESSIVE's position would be to suppose that AMICABLE disclosed the confidential information to 100 outsiders as in the *Midland Ross* case, a potential mopc where the defendant won. Or HYPO could suppose that the confidential information was general knowledge about customer business relations as in the pro-defendant mopc *Automated Systems*. HYPO can tighten the analogy between the current fact situation and the most-on-point cases *Telex* or *Structural Dynamics*, and strongly improve plaintiff's argument if the facts included that AGGRESSIVE bribed Ian Smart to change employers or if Ian Smart brought AMICABLE's product-related tools with him.

4.5 Step 7: The Skeletal Outline of an Argument

Based on its analysis, including which cases are actual or potential mopc's for each opponent, the ARGUMENT module can now summarize points and responses for the cfs.

For instance, starting from the point of view of the plaintiff, there are really two ways to argue the case, one for each of the two main branches of the claim-lattice shown in Figure 2. Here are two points that HYPO makes for Amicable: (1) citing *Structural Dynamics* from the top branch of the claim-lattice; (2) citing *Telex* from the bottom branch.

Points (for plaintiff) :

AMICABLE should win claim for trade secrets misappropriation:

(1) *See Telex v. IBM* (Plaintiff IBM won trade secret misappropriation claim where defendant Telex gained competitive headstart by saving 50% in development time and cost by using confidential information of former IBM employees who had agreed not to disclose IBM's proprietary information.)

(2) *See Structural Dynamics v. Engineering Mechanics* (Plaintiff Structural Dynamics won trade secret misappropriation claim, eventhough plaintiff's former employee was sole developer of product, where employee agreed not to disclose confidential information .)

In the absence of any pro-defendant most-on-point cases, (see Figure 3) defendant has no cases to cite in response. As discussed above, HYPO poses hypos based on the pro-defendant *potential* most-on-point cases to try to generate some cases to cite for defendant. All that is left for HYPO to do to respond to the points is to distinguish plaintiff's cases by pointing out significant factual differences:

Responses (for defendant) :

(1) The *Telex* case is distinguishable because it had stronger facts for plaintiff: In the *Telex* case, defendant Telex bribed IBM's employees to join Telex by offering a $500,000 bonus, stock options and high salaries. This was not so in the *Amicable* case.

(2) The *Structural Dynamics* case is distinguishable because it had stronger facts for plaintiff: In *Structural Dynamics* the employee brought product-related tools like a notebook and copies of the code and the nondisclosure agreement specifically applied to the product. This was not so in the *Amicable* case.

Where a response for *side2* cites cases as counter-examples, HYPO also makes a rebuttal on behalf of *side1* by distinguishing those cases from the current fact situation. In this example, since no cases could be cited for defendant in the Responses, there are no Rebuttals for the plaintiff.

4.6 The Real Case

In several important respects, HYPO's analysis compares favorably with that of the Court in the real case of *Amoco Production Co. v. Lindley*, 609 P2d 733 (Okla. 1980):

First, in its opinion, the Court cites both the *Telex* and *Structural Dynamics* cases, focussing on the existence in each case of nondisclosure agreements between the plaintiff and its employees. *Id.* at 743-745.

Second, the Court also *distinguishes* the *Structural Dynamics* case by pointing out that the terms of the nondisclosure agreement in *Structural Dynamics* were more restrictive than those in *Amoco Production.Id.* at 745.

Third, the Court in the *Amoco Production* case did not decide the merits of the trade secrets claim. Instead, it sent the case back to the trial court for further action. In effect, the Court's citing cases like *Telex* and *Structural Dynamics* was to guide the lower court as to what factual findings to seek and how to legally evaluate the facts. HYPO uses cases in much the same way.

Fourth, in the *Telex* and *Structural Dynamics* cases, defendants raised, and the trial courts rejected, the defenses that there were no trade secrets because plaintiffs disclosed the information to outsiders and that the information was too general. *Telex Corp. v. IBM Corp.*, 367 F.Supp 258, 358 (N. D. Okla. 1973); *Structural Dynamics Research Corp. v. Engineering Mechanics Research Corp.*, 401 F. Supp. 1102, 1117 (E. D. Mich. 1975). Both of these defenses are implicit in the hypotheticals posed by HYPO based on the *Midland Ross* and *Automated Systems* cases. Indeed, in the *Telex* case, the court cites another *Midland Ross* case involving the same plaintiff and the same defense that the alleged trade secrets had been disclosed.

5. Conclusions

In this paper, we have described the key ingredients of case-based reasoning (CBR) and shown how our system HYPO performs CBR in the legal domain of trade secrets law. In discussing HYPO, and other related research, we emphasized how CBR depends critically on a Case-Knowledge-Base (CKB) and indexing schemes and that the hallmark of CBR is the use of cases in justification. The law is an excellent domain to study such CBR problems since by its very nature it espouses a doctrine of precedent and a realization that there are "no right answers". The law is also a paradigm for *adversarial* CBR; interpretations are pitted against each other.

Our system HYPO performs indexing and relevancy assessment of past cases dynamically by (1) analyzing how prior cases can be viewed from the point of view of the current fact situation (cfs) and (2) determing what aspects of these prior cases apply, and how strongly, to the current fact situation. This sort of analysis – accomplished through HYPO's "dimensions", "case-analysis-record" and "claim-lattice" mechanisms – allows HYPO to promote some prior cases over others as precedents for interpreting and arguing about the current fact situation. While the case-analysis-record views the cfs with respect to the existing cases in the CKB, the claim-lattice views these cases with respect to the cfs. Through creation of neighborhoods of relevancy, centered on the cfs, HYPO can easily select most-on-point cases (mopc's) both pro and con a position and generate the skeletal structure of an argument, a "3-ply argument".

In HYPO, and CBR in general, relevancy and indexing issues touch on central concerns of AI, particularly, machine learning. In assessing relevancy, a CBR system is in effect grappling with the problem of credit assignment, and in indexing, the bias and new term problems. While CBR research will thus benefit from fundamental advances in machine learning, CBR expertise in selecting, creating, and manipulating cases should in a reciprical way benefit learning resarch, particularly, on problems concerning the selection of training instances.

Current work on HYPO is focussed on the modules that do explanation and argumentation, as well as, of course, on expanding the knowledge bases. We are also working on tests to pit and benchmark HYPO's performance against legal experts, for instance, by generating sample test cases and evaluation measures (e.g., based on cases cited). Ultimately, such research should shed light on the nature of *stare decisis* and give detailed, well-explicated computational models of its component reasoning processes.

REFERENCES

[Ashley, 1987a] Kevin D. Ashley. Arguing by Analogy in Law: an A.I. Model. To appear in David H. Helman, editor, *Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*. D. Reidel, 1987.

[Ashley, 1987b] Kevin D. Ashley. Distinguishing – A Reasoner's Wedge. In *Ninth Annual Cognitive Science Society Conference*, University of Washington, Seattle, 1987.

[Ashley, 1987c] Kevin D. Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. PhD thesis, Department of Computer and Information Science, University of Massachusetts, 1987.

[Ashley and Rissland, 1987a] Kevin D. Ashley and Edwina L. Rissland. But, See, Accord: Generating *Blue Book* Citations in HYPO. In *First International Conference on Artificial Intelligence and Law*, Northeastern University, Boston MA, 1987.

[Ashley and Rissland, 1987b] Kevin D. Ashley and Edwina L. Rissland. Compare and Contrast, A Test of Expertise. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, August 1987. Seattle.

[Ashley and Rissland, 1988] Kevin D. Ashley and Edwina L. Rissland. Dynamic Assessment of Relevancy in a Case-Based Reasoner. In *Proceedings of the Fourth IEEE Conference on Artificial Intelligence Applications*, The Institute of Electrical and Electronics Engineers, Inc., March 1988. San Diego.

[Bareiss *et al.*, 1987] E. Ray Bareiss, Bruce W. Porter, and Craig C. Wier. Protos: An Exemplar-Based Learning Apprentice. In *Proceedings Fourth International Workshop on Machine Learning*, pages 12–23, University of California at Irvine, June 1987.

[Birnbaum, 1982] L. Birnbaum. Argument Molecules: A Functional Representation of Argument Structure. In *Proceedings of the Second National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Pittsburgh, PA, August 1982.

[Birnbaum *et al.*, 1980] L. Birnbaum, M. Flowers, and R. McGuire. Towards an AI Model of Argumentation. In *Proceedings of the First National Conference on Artificial Intelligence*, pages 313–315, American Association for Artificial Intelligence, August 1980. Stanford, CA.

[Blue Book, 1986] Blue Book. *A Uniform System of Citation*. The Harvard Law Review Association, fourteenth edition, 1986.

[Buchanan *et al.*, 1987] B. G. Buchanan, E.L. Rissland, P. R. Rosenbloom, H. T. Ng, and Sullivan. The Role of Intelligent Instance Selection in Learning Systems: The Near Miss. Submitted to the Journal of Machine Learning, 1987.

[Burstein, 1983] M. H. Burstein. A Model of Learning by Incremental Analogical Reasoning and Debugging. In *Proceedings of the Third National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Washington, D.C., August 1983.

[Carbonell, 1982] J. G. Carbonell. Experiential Learning in Analogical Problem Solving. In *Proceedings of the Second National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Pittsburgh, PA, August 1982.

[Carbonell, 1983a] J. G. Carbonell. Derivational Analogy and its Role in Problem Solving. In *Proceedings of the Third National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Washington, D.C., August 1983.

[Carbonell, 1983b] J. G. Carbonell. Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In Michalski, J.G. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing, CA, 1983.

[Carbonell, 1986] J. G. Carbonell. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In R. S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach, Volume II*, Morgan Kaufmann, Los Altos, CA, 1986.

[Clancey, 1983] W.J. Clancey. The Epistemology of a Rule-based Expert System: A Framework for Explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

[Cohen, 1983] Robin Cohen. *Computational Model for the Analysis of Arguments*. Technical Report CSRG-151, Department of Computer Science, University of Toronto, 1983.

[De Jong and Mooney, 1986] G. De Jong and R. Mooney. Explanation-Based Learning: An Alternative View. *Machine Learning*, 1(2), 1986.

[Gardner, 1984] A. vdL. Gardner. *An Artificial Intelligence Approach to Legal Reasoning*. PhD thesis, Department of Computer Science, Stanford University, 1984.

[Gilburne and Johnston, 1982] M. R. Gilburne and R. L. Johnston. Trade Secret Protection for Software Generally and in the Mass Market. *Computer Law Journal*, III(3), 1982.

[Hammond, 1986a] Kristian J. Hammond. CHEF: A Model of Case-based Planning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, August 1986. Philadelphia, PA.

[Hammond, 1986b] Kristian J. Hammond. Learning to Anticipate and Avoid Planning Problems through the Explanation of Failures. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, August 1986. Philadelphia, PA.

[Hammond, 1987] Kristian J. Hammond. Explaining and Repairing Plans that Fail. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., August 1987. Milan, Italy.

[Kedar-Cabelli, 1984] S. Kedar-Cabelli. *Analogy with Purpose in Legal Reasoning from Precedents A Dissertation Proposal.* Technical Report LRP-TR-17, Laboratory for Computer Science Research, Rutgers University, 1984.

[Kennedy, 1980] Duncan Kennedy. Utopian Proposal. Draft memo, Harvard Law School, 1980.

[Kolodner, 1983a] Janet L. Kolodner. Maintaining Organization in a Dynamic Long-Term Memory. *Cognitive Science*, 7(4):243–280, 1983.

[Kolodner, 1983b] Janet L. Kolodner. Reconstructive Memory: A Computer Model. *Cognitive Science*, 7(4):281–328, 1983.

[Kolodner *et al.*, 1985] Janet L. Kolodner, Robert L. Simpson, and Katia Sycara-Cyranski. A Process Model of Case-Based Reasoning in Problem Solving. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[Kuhn, 1970] T. S. Kuhn. *The Structure of Scientific Revolutions.* U. Chicago, 1970.

[Lakatos, 1976] I. Lakatos. *Proofs and Refutations.* Cambridge University Press, London, 1976.

[Lenat, 1977] D. B. Lenat. Automated Theory Formation in Mathematics. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 833–841, International Joint Conferences on Artificial Intelligence, Inc., Cambridge, MA, August 1977.

[Levi, 1949] Edward H. Levi. *An Introduction to Legal Reasoning.* University of Chicago Press, 1949.

[Llewellyn, 1933] Karl N. Llewellyn. Praejudizienrecht und Rechtssprechung in Amerika. 1933. Section 52, Certainty in Case Law; In Doubtful Cases the Legal Rule is not Decisive, pages 72–86.

[McCarty, 1980] L. Thorne McCarty. Example Generation. In *Third Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, University of Victoria, Victoria, B.C., 1980.

[McCarty, 1985] L. Thorne McCarty. Permissions and Obligations. In Charles Walter, editor, *Computing Power and Legal Reasoning*, West Publishing Co., St. Paul, MN, 1985.

[McCarty and Sridharan, 1982] L. Thorne McCarty and N.S. Sridharan. *A Computational Theory of Legal Argument.* Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University, 1982.

[McGuire *et al.*, 1981] R. McGuire, L. Birnbaum, and M. Flowers. Opportunistic Processing in Arguments. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 58–60, International Joint Conferences on Artificial Intelligence, Inc., Vancouver, B.C., August 1981.

[Meldman, 1977] Jeffrey A. Meldman. A Structural Model for Computer-Aided Legal Analysis. *Journal of Computers and the Law*, 6:27–71, 1977.

[Milgrim, 1985] Roger M. Milgrim. *Business Organizations, Milgrim on Trade Secrets.* Volume 12, Matthew Bender, New York, 1985.

[Mitchell *et al.*, 1983] T.M. Mitchell, P.E. Utgoff, and R.B. Banerji. Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics. In Michalski, Carbonell, and Mitchell, editors, *Machine Learning An Artificial Intelligence Approach*, pages 163–190, Tioga, 1983.

[Neustadt and May, 1986] R. E. Neustadt and E. R. May. *Thinking in Time.* The Free Press, New York, 1986.

[Polya, 1973] G. Polya. *How To Solve It.* Princeton University Press, second edition, 1973.

[Radin, 1933] Max Radin. Case Law and Stare Decisis. *Columbia Law Review*, 33:199, 1933.

[Reichman, 1981] R. Reichman. Modeling Informal Debates. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Vancouver, B.C., August 1981.

[Reichman-Adar, 1984] R. Reichman-Adar. Extended Person-Machine Interface. *Artificial Intelligence*, 22:157–218, 1984.

[Rendell, 1985] L. Rendell. Substantial Constructive Induction Using Layered Information Compression: Tractable Feature Formation in Search. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 650–658, International Joint Conferences on Artificial Intelligence, Inc., Los Angeles, CA, August 1985.

[Rissland, 1978] Edwina L. Rissland. Understanding Understanding Mathematics. *Cognitive Science*, 1(4), 1978.

[Rissland, 1980] Edwina L. Rissland. Example Generation. In *Third Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, University of Victoria, Victoria, B.C., 1980.

[Rissland, 1981] Edwina L. Rissland. *Example Generation.* Technical Report 81-24, Computer and Information Science Department, University of Massachusetts, Amherst, MA, 1981.

[Rissland, 1983] Edwina L. Rissland. Examples in Legal Reasoning: Legal Hypotheticals. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence, Inc., Karlsruhe, Germany, August 1983.

[Rissland, 1985] Edwina L. Rissland. Argument Moves and Hypotheticals. In Charles Walter, editor, *Computing Power and Legal Reasoning*, West Publishing Co., St. Paul, MN, 1985.

[Rissland, 1987] Edwina L. Rissland. The Problem of Intelligent Example Selection. In *Proceedings of the Second Banff Knowledge Acquisition Workshop*, To appear in International Journal Man-Machine Studies, 1988, Banff, Alberta, October 1987.

[Rissland and Ashley, 1986] Edwina L. Rissland and Kevin D. Ashley. Hypotheticals as Heuristic Device. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, August 1986. Philadelphia, PA.

[Rissland *et al.*, 1984] Edwina L. Rissland, E. M. Valcarce, and Kevin D. Ashley. Explaining and Arguing with Examples. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, Austin, TX, August 1984.

[Schlobohm and Waterman, 1987] Dean A. Schlobohm and Donald A. Waterman. Explanation for an Expert System that Performs Estate Planning. In *First International Conference on Artificial Intelligence and Law*, Northeastern University, Boston MA, 1987.

[Toulmin, 1958] Stephen Toulmin. *The Uses of Argument.* Cambridge U. Press, 1958.

[Utgoff, 1983] Paul E. Utgoff. Adjusting Bias in Concept Learning. In *Proceedings of the International Machine Learning Workshop*, pages 105–109, Monticello, IL, 1983.

[Waterman and Peterson, 1981] D. A. Waterman and M. Peterson. *Models of Legal Decisionmaking.* Technical Report R-2717-1CJ, The Rand Corporation, Santa Monica, CA, 1981.

[Winston, 1980] Patrick H. Winston. Learning and Reasoning by Analogy. *Communications of the ACM*, 23(12):689–703, 1980.