# THE DESIGN OF A LEGAL ANALYSIS PROGRAM*

Anne v.d.L. Gardner
Department of Computer Science
Stanford University
Stanford, California 94305

## ABSTRACT

The analysis of legal problems is a relatively new domain for AI. This paper outlines a model of legal reasoning, giving special attention to the unique characteristics of the domain, and describes a program based on the model. Major features include (1) distinguishing between questions the program has enough information to resolve and questions that competent lawyers could argue either way; (2) using incompletely defined ("open-textured") technical concepts; (3) combining the use of knowledge expressed as rules and knowledge expressed as examples; and (4) combining the use of professional knowledge and commonsense knowledge. All these features may prove important in other domains besides law, but previous AI research has left them largely unexplored.

## I INTRODUCTION

This paper describes a program for analyzing legal problems--specifically, problems about the formation of contracts by offer and acceptance. The work brings together two areas of AI usually treated as distinct. One is research on expert systems (e.g., Buchanan 1981, Davis 1982, Stefik et al. 1982); the other, natural-language understanding and commonsense reasoning (e.g., Schank and Abelson 1977; Winograd 1980). The expert-systems area is obviously relevant, since a legal analysis program requires substantial professional knowledge. The natural-language aspect is present, in part, because of the particular legal subdomain: in offer-and-acceptance problems, the data to be interpreted consist mostly of reported dialogue.

There is also a deeper reason for the natural-language aspect of legal analysis. This reason, explained in the next section, is the *open texture* of many legal predicates. It applies equally to legal subdomains such as assault and battery (Meldman 1975), corporate taxation (McCarty, Sridharan, and Sangster 1979; McCarty and Sridharan 1982), and manufacturers' product liability (Waterman and Peterson 1981), as well as contract law.

The program has been implemented in Maclisp on a DECSYSTEM-20. Database storage, retrieval, and basic inference capabilities are provided by the representation language MRS (Genesereth, Greiner, and Smith 1980).

*This research was supported in part by a fellowship from IBM.

## II DOMAIN CHARACTERISTICS AND DESIGN CONSIDERATIONS

The design of the program is intended to reflect lawyers' own understanding of the nature and uses of legal materials--in other words, to accord with a legally plausible conceptualization of the domain. Some of the distinctive domain features are the following:

1. Legal rules are used consciously by the expert to provide guidance in the analysis, argumentation, and decision of cases. This fact distinguishes them from the rules used in most expert systems or the rules of a grammar, which seek to describe behavioral regularities of which the expert or native speaker may be unaware. Legal reasoning might thus be classified as a *rule-guided* activity rather than a rule-governed activity.

2. As a consequence of (1), the experts can do more with the rules than just follow them. In a field like contracts, where the rules have been developed mainly through decisions in individual cases, lawyers can argue about the rules themselves and can propose refinements, reformulations, or even newly formulated rules to adapt the law to a particular case at hand. Sometimes, it is true, the rules may be taken as fixed--either by long acceptance, in a case-law field, or by statute, in a field like taxation. Even with this simplification, lawyers are free to argue about what counts as following the rules in a particular case.

3. Lawyers are not merely free to disagree; on hard legal questions they are expected to do so. Unlike other domains of expertise, in which consensus among the experts is hoped for, the legal system makes institutional provision for expert disagreement--for instance, in the institutions of opposing counsel, dissenting judicial opinions, and appellate review of lower court decisions.

4. The following question then arises: Is there any class of cases as to which all competent lawyers would reach the same conclusion? This is the problem, recognized but not solved in the legal literature, of whether a dividing line between hard cases and clear cases can be found (see, e.g., Hart 1958, Fuller 1958, M. Moore 1981). Despite the lack of a theoretical solution, most cases are in fact treated as raising no hard questions of law. (Whether they raise hard questions of fact is another matter.)

5. When hard legal questions do arise, their basis is quite different from the sources of uncertainty usually described in connection with expert systems. They do not generally involve insufficient data, for example, or incomplete understanding of the workings of some physical process. Instead, an especially important source of hard questions is the open texture of legal predicates--that is, the inherent indeterminacy of meaning in the words by which fact situations are classified into

instances and noninstances of legal concepts (see Hart 1961, pp. 121-132).

The phenomenon of open texture is not limited to law. The term was coined in philosophy and used originally of words like *dog* and *gold* in pointing out that *most* of our empirical concepts are not delimited in all possible directions (Waismann 1945). Recent analyses of such natural-kind words, and other sorts of words too, have involved closely related observations (e.g., Putnam 1975; see generally Schwartz 1977).

6. The final problem is resolving legal questions, hard or easy. How does the judge carry out this task? How should he do it? Having done it, how should he justify his results in a written opinion? These questions--often not distinguished from one another--are central in legal philosophy. Different writers, all intimately familiar with the judicial process, paint rather different pictures of it (e.g., Levi 1949, Llewellyn 1960, Hart 1961, Dworkin 1977). They agree on this much: in a well-developed, relatively stable field of law (like contracts), there are at least two distinct knowledge sources that must be brought to bear. Legal rules are one; and rules exist even in a nonstatutory field (like contracts) where they lack official wording. (For an influential unofficial attempt to state the rules of contract law, see Restatement of Contracts, 1932, and Restatement of Contracts, Second, 1981.) Second, there are decisions in previous cases. There is no tidy consensus about just how the rules and the precedents are used together.

These domain characteristics dictate the main features of the program. The overall objective is not a program that "solves" legal problems by producing a single "correct" analysis. Instead, the objective is to enable the program to recognize the issues a problem raises and to distinguish between those it has enough information to resolve and those on which competent human judgments might differ. Toward this end, a heuristic distinction between hard and easy questions is proposed. The distinction in turn draws on ideas about how rules and examples interact and how their interaction allows for open texture.

## III THE TASK

To provide a definite context for studying legal reasoning, the research uses materials classically taught by the case method in law schools and classically tested by asking the student, given the facts of a new case, to analyze their legal consequences. The specific legal topic, as already mentioned, is the formation of contracts by offer and acceptance. The topic is a standard one for first-year law students. A typical examination question is the following:

> On July 1 Buyer sent the following telegram to Seller: "Have customers for salt and need carload immediately. Will you supply carload at $2.40 per cwt?" Seller received the telegram the same day.
>
> On July 12 Seller sent Buyer the following telegram, which Buyer received the same day: "Accept your offer carload of salt, immediate shipment, terms cash on delivery."
>
> On July 13 Buyer sent by Air Mail its standard form "Purchase Order" to Seller. On the face of the form Buyer had written that it accepted "Seller's offer of July 12" and had written "One carload" and "$2.40 per cwt." in the appropriate spaces for quantity and price. Among numerous printed provisions on the reverse of the form was the following: "Unless otherwise stated on the face hereof, payment on all purchase orders shall not be due until 30 days following delivery." There was no statement on the face of the form regarding time of payment.
>
> Later on July 13 another party offered to sell Buyer a carload of salt for $2.30 per cwt. Buyer immediately wired Seller: "Ignore purchase order mailed earlier today; your offer of July 12 rejected." This telegram was received by Seller on the same day (July 13). Seller received Buyer's purchase order in the mail the following day (July 14).
>
> Briefly analyze each of the items of correspondence in terms of its legal effect, and indicate what the result will be in Seller's action against Buyer for breach of contract.

In automating the analysis of such questions, the first step is to construct a representational formalism to which the English problem statement can be (manually) translated. The primary problem here is to create an ontology of the problem domain and to specify the ways its entities may combine. Many of the issues are discussed in R. Moore (1981).

In the current representation, the major domain classes include events (with acts by individuals as a subclass), states, physical objects and substances, symbolic objects (namely, sentences and propositions), measures (as of weight and volume), and times. Acts are subdivided into ordinary acts (e.g., uttering the sentence "I offer ..."), speech acts (e.g., declaring, perhaps ineffectually, that an offer by the speaker is being made), and legal acts (e.g., offering).

The classes are arranged in a generalization hierarchy. Each class name is a unary predicate symbol; possible relations among entities are given by binary predicates corresponding to the slot names of a frame representation. Formulas using these predicates are written in logical notation, as is required by the representation language MRS.

## IV THE OUTPUT

Given an encoding of the problem, the program's task, as indicated earlier, is not to produce a single solution but rather to identify the important issues. The output is a graph structure similar to a decision tree, displaying the different analyses of the case that are possible in light of the issues left open. In the problem quoted above, at least four analyses should be reported:

1. The first telegram is an offer and the second an acceptance. Hence a contract was formed, which Buyer later repudiated. Seller wins.

2. The first telegram is an offer, but it expired before Seller replied to it eleven days later. Or, with the same net result, the first telegram is only a preliminary inquiry. But the second telegram is an offer and the purchase order an acceptance. Buyer repudiated the resulting contract. Seller wins.

3. As in (2), the second telegram is an offer and the purchase order an acceptance. But the final telegram operated to revoke the acceptance and reject the offer. So there is no contract, and Seller loses.

4. As in (2), the second telegram is an offer. The purchase order, proposing a change in the terms of payment, operated only as a counteroffer, which the final telegram withdrew. Again, Seller loses.

The graph displaying these results has two levels, which are comparable to the distinct abstraction levels used in hierarchical planning (Sacerdoti 1974, 1980-81). The upper level is a tree in which each node corresponds to the question of what legal characterization to attach to a particular event--in light of the characterizations of any earlier events, as represented along the path from the root to the node in question. On the lower, more detailed level of the graph, a separate tree may be associated with each upper level node. In a detailed tree, nodes correspond to questions encountered in trying to reach a characterization of the event being examined. These include both hard legal questions (e.g., was the July 12 telegram a timely response to the July 1 offer?) and computational choice points (such as which of several candidate bindings for a variable will turn out to be appropriate). Results reached at the detailed level are summarized at the level above, reducing the combinatorics of the problem. With some further refinement, this summary level could be the basis for an essay answer to the examination question.

## V KNOWLEDGE SOURCES AND TASK DECOMPOSITION

To produce the analysis graph just described, the program uses what are conceptually three distinct stages of reasoning, each with its own knowledge source or sources. A fourth stage, not now implemented, should eventually be added. The four stages are described in the following subsections.

### A. Time Sequencing and Basic Domain Categories

Offer-and-acceptance problems require tracing changing legal relations over a period of time marked out by a sequence of discrete events. Reflecting this requirement, the program uses an augmented transition network whose states are elements of the space of possible legal relations and whose arcs are the possible ways of moving among them. The current states are:

| | |
|---|---|
| state 0 | No relevant legal relations exist. |
| state 1 | One or more offers are pending; the offeree has the power to accept. |
| state 2 | A contract exists. |
| state 12 | A contract exists and a proposal to modify it is pending. |

Available state transitions include:

| | |
|---|---|
| 0 to 1 | Offer |
| 1 to 0 | Rejection by the offeree; revocation by the offeror; death of either party |
| 1 to 1 | Counteroffer by the offeree |
| 1 to 2 | Acceptance by the offeree |

Given a problem as simple as "Joe made an offer and Bill accepted it," the program would find a contract without ever going beyond stage 1.

### B. Legal Rules

Attached to each arc is a set of legal rules stating how the arc predicate may be found to be satisfied. Predicates occurring in the preconditions of rules are understood to be technical legal terms. To the extent that these predicates represent concepts to which contract law gives a definite structure, additional rules may be available to be invoked by backward chaining.

There are also a few predicates that are tested procedurally. For example, an acceptance must concern "the same bargain proposed by the offer" (Restatement of Contracts, Second, sec. 50, comment a). It is easier here to apply a domain-dependent matching procedure to the contents of two documents or utterances than to state declaratively when such a procedure would succeed.

Within a set of rules leading to the same conclusion, two different relationships may hold among the members: the rules are either complementary, in that they provide alternate ways of reaching the conclusion, or they are competing, reflecting an unsettled state of the law where rules have been formulated but there is disagreement about what the rule should be. The possibility that no existing formulation is satisfactory, and that a new rule should be formulated on the fly, is not now provided for.

The choice between competing rules, if it affects the legal characterization of an event, is always considered to raise a hard legal question.

### C. Open-textured Predicates

The property of open texture is understood as attaching to legal predicates at which the rules run out--that is, those predicates which lack an attached procedure and which occur in the antecedent of some rule but not in the consequent of any. At this point, two main knowledge sources become available: knowledge of ordinary language, and knowledge of legal precedents and hypothetical examples.

With respect to ordinary language, the idea is that the same English word (and, correspondingly, the same formal predicate symbol) may have both a technical and a nontechnical sense. The senses are not independent: in choosing words in which to formulate a legal rule, one draws on their ordinary meanings. To decide whether a rule applies to a particular case, one may need to consider both (a) whether the ordinary usage of its words suggests an answer and (b) whether technical usage does or should conform to ordinary usage.

In the implementation, a predicate symbol is considered to have an ordinary or commonsense meaning if it occurs in the generalization hierarchy described earlier. The program's very limited commonsense knowledge is expressed by rules of the following kinds:

- Rules stating subset-superset relations.
- Rules stating that certain subsets are mutually exclusive, exhaustive of the parent set, or both.
- Rules specifying what slots always have fillers, for what slots the filler is unique, and what can be inferred about an entity by virtue of its filling a particular slot.
- Rules giving meaning to further predicates in terms of those occurring in the basic hierarchy.

Deduction using these commonsense rules may produce an answer--but not yet a conclusive one--as to whether the legal predicate is satisfied.

As to the technical usage of the legal predicate, it is here that previous cases, actual and hypothetical, come in. The cases are thought of as giving a partial extensional or semantic definition of the predicate: though we don't know what its full definition by a formal rule might be, we do know that, under our reading of the cases, the facts in Armstrong v. Baker were found to satisfy the predicate and the facts in Carter

v. Dodge were found not to. As indicated, both positive and negative examples may be included.

Representation of the cases at two levels of abstraction is assumed. At one level, the facts of a case are represented similarly to the facts of the input problem. Use of this relatively full representation is reserved for stage 4 of the reasoning process. For use in stage 3, the cases are represented more abstractly, in the form of simple patterns including only the facts relevant to the satisfaction of a particular predicate. In this abstract representation, one case may give rise to several patterns pertaining to different predicates, and one abstract pattern may derive from several cases.

As an example of the sort of patterns used, consider the definition of acceptance. One antecedent calls for deciding whether the offer permits acceptance by promise (as opposed to acceptance by the offeree's simply performing his side of the offered bargain). Positive examples cover offers that ask an appropriate question or request an appropriate speech act. A negative example is an offer of reward: no contract is formed, for instance, when someone merely promises to find a lost object.

Using abstract examples based on the cases, then, exact matches are sought in the facts of the case at hand. The examples may supply a meaning to a predicate where commonsense knowledge is lacking; they may supply a technical meaning that supersedes the ordinary meaning; and they may even conflict with each other, as is indicated if both positive and negative examples are matched in the data.

Heuristically, satisfaction of a legal predicate is considered an easy question--one within the program's competence to resolve--if an answer can be derived from either commonsense knowledge or case knowledge or both, provided that conflicting cases are not found. If the knowledge sources provide no answer or the cases point both ways, a hard legal question has been identified, and a branch point is entered into the output graph.

### D. Arguing the Hard Questions

The first three stages of reasoning are sufficient to produce the output graph described in section IV, identifying the significant issues in the case. By hypothesis, these are the questions requiring human judgment. Still, the program might do more. The final stage, which remains for future development, would be to produce arguments on both sides of the hard questions. This is the aspect of legal reasoning with which McCarty and Sridharan's current work (1982) is concerned. In the present design, the arguments are envisioned as annotations to the output graph. If their relative merits can be evaluated, the result would be a set of recommendations as to how to prune the output graph. With enough pruning recommendations to leave only one path through the graph, the annotations would correspond to one possible decision.

## VI CURRENT STATUS AND FUTURE DIRECTIONS

The current program contains all the mechanisms described except for those of section V.D. The transition network (section V.A) has 4 states and 20 arcs. There are 14 legal rules (section V.B) defining such concepts as offer, acceptance by promise, and rejection. The major definitions have from 13 to 20 antecedents, some 40% of which have attached examples. There are about 125 commonsense rules of the kinds mentioned in section V.C. The knowledge base is sufficient for processing of the test problem quoted above.

The test problem has been somewhat simplified. Some concepts have been omitted for lack of a good representation, notably "immediate" and "ignore." Fortunately nothing in the analysis turns on the presence of these concepts. As another simplification, the dialogue has been reconstructed as if it consisted of complete sentences.

The program produces 9 analyses of the problem; that is, the summary level of the output graph contains 9 paths from the root to a terminal node. The first 7 paths correspond to the 4 analyses listed at the beginning of section IV. (The 3 extras arise because the program is not yet able to conclude that treating the first telegram as an expired offer is equivalent, in this problem, to treating it as a preliminary inquiry.) The remaining paths reflect the possibility that the first two telegrams are both only preliminary negotiation. The legal question that raises this possibility is whether they state the terms of the sale definitely enough for a court to enforce them.

Much of the programming effort has gone into avoiding a combinatorial explosion of alternatives. At the detailed level of the output graph, the trees generated in characterizing a single event may have half a dozen terminal nodes. Before the program goes on to the next event, it combines these at the summary level, usually reducing them to a single two-way branch.

Within the detailed level there is also a potential for unnecessary computation. To characterize the current event, it may be necessary also to access (1) assertions belonging to previous events and their interpretation by the program and (2) propositions that are embedded in assertions about the current event and whose truth value is unknown. MRS has a context mechanism that makes it easy to create distinct worlds for asserting the latter propositions hypothetically and for segregating believed assertions into groups that can be made available or unavailable as desired. The trick is to find general formulations for which contexts should be accessible at any given time--in order to produce the correct matches without several superfluous ones that are bound to fail later. Experience is gradually yielding the needed formulations. At present, the processing time for the more complicated events--those including documents whose content must be analyzed--ranges from 2 to 6 minutes.

In summary, the program is very close to analyzing the test problem satisfactorily, and the general design continues to seem appropriate. To permit stronger conclusions, the knowledge base will have to be enlarged considerably. More legal rules should be added to remove artificial restrictions on the kinds of problems that can be handled. For example, reasonable coverage would require knowing about accepting an offer by a nonverbal act or even by doing nothing. Most importantly, more examples are needed at the technical-commonsense boundary. These can come incrementally, from the casebooks. The fact to be taken advantage of here is that court decisions do more than resolve the hard issues in litigated cases. They also describe the contexts in which these issues arise and, in doing so, provide a rich source of information about the nonproblematical aspects of cases, on which commonsense knowledge and technical knowledge agree.

With these enlargements, a sharper critique of the program will become possible. It will then be time to consider some difficult long-range problems: What changes would be necessary to enable the program to reason about cases involving mistake or misunderstanding between the parties? Having identified the significant issues in a case, how could the program then go about reasoning from detailed descriptions of the precedents to produce arguments on both sides of these questions?

## REFERENCES


Buchanan, Bruce G. 1982. "New Research on Expert Systems." In J. E. Hayes, Donald Michie, and Y-H Pao, eds., *Machine Intelligence 10* (New York: Halsted Press, John Wiley & Sons).

Davis, Randall. 1982. "Expert Systems: Where Are We? And Where Do We Go from Here?" *AI Magazine* 3(2), 3-22.

Dworkin, Ronald. 1977. *Taking Rights Seriously.* Cambridge: Harvard University Press.

Fuller, Lon L. 1958. "Positivism and Fidelity to Law: A Reply to Professor Hart." *Harvard Law Review* 71, 630-672.

Genesereth, Michael R.; Greiner, Russell; and Smith, David E. 1980. "MRS Manual." Memo HPP-80-24, Stanford Heuristic Programming Project, Stanford University.

Hart, H. L. A. 1958. "Positivism and the Separation of Law and Morals." *Harvard Law Review* 71, 593-629.

Hart, H. L. A. 1961. *The Concept of Law.* Oxford: Clarendon Press.

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

Llewellyn, Karl N. 1960. *The Common Law Tradition: Deciding Appeals.* Boston: Little, Brown.

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

McCarty, L. Thorne; Sridharan, N. S.; and Sangster, Barbara C. 1979. "The Implementation of TAXMAN II: An Experiment in Artificial Intelligence and Legal Reasoning." LRP-TR-2, Laboratory for Computer Science Research, Rutgers University.

Meldman, Jeffrey A. 1975. "A Preliminary Study in Computer-Aided Legal Analysis." MAC-TR-157, M.I.T.

Moore, Michael S. 1981. "The Semantics of Judging." *Southern California Law Review* 54, 151-294.

Moore, Robert C. 1981. "Problems in Logical Form." In *Proceedings, 19th Annual Meeting, Association for Computational Linguistics.* Pp. 117-124.

Putnam, Hilary. 1975. "The Meaning of 'Meaning.'" In Keith Gunderson, ed., *Language, Mind, and Knowledge,* Minnesota Studies in the Philosophy of Science, vol. 7. Minneapolis: University of Minnesota Press. Pp. 131-193. (Reprinted in H. Putnam, *Philosophical Papers,* vol. 2. *Mind, Language and Reality.* Cambridge: Cambridge University Press, 1975. Pp. 215-271.)

*Restatement of the Law of Contracts.* 1932. 2 vols. St. Paul: American Law Institute Publishers.

*Restatement of the Law, Second: Contracts 2d.* 1981. 3 vols. St. Paul: American Law Institute Publishers.

Sacerdoti, Earl D. 1974. "Planning in a Hierarchy of Abstraction Spaces." *Artificial Intelligence* 5, 115-135.

Sacerdoti, Earl D. 1980-81. "Problem Solving Tactics." *AI Magazine* 2(1), 7-15.

Schank, Roger C., and Abelson, Robert P. 1977. *Scripts, Plans, Goals and Understanding.* Hillsdale, N. J.: Lawrence Erlbaum.

Schwartz, Stephen P., ed. 1977. *Naming, Necessity, and Natural Kinds.* Ithaca: Cornell University Press.

Stefik, Mark, et al. 1982. "The Organization of Expert Systems: A Tutorial." *Artificial Intelligence* 18(2), 135-173.

Waismann, Friedrich. 1945. "Verifiability." *Proceedings of the Aristotelian Society,* Suppl. 19, 119-150. Reprinted in Antony Flew, ed., *Logic and Language: First and Second Series.* Garden City: Anchor Books, 1965. Pp. 122-151.

Waterman, D. A., and Peterson, Mark A. 1981. "Models of Legal Decisionmaking." Report R-2717-ICJ, Rand Corporation, Institute for Civil Justice.

Winograd, Terry. 1980. "What Does It Mean to Understand Language?" *Cognitive Science* 4, 209-241.