

**Computer Science Department Technical Report  
University of California  
Los Angeles, CA 90024-1596**

**TOWARDS A COMPUTATIONAL MODEL OF CONCEPT  
ACQUISITION AND MODIFICATION USING CASES  
AND PRECEDENTS FROM CONTRACT LAW**

**Seth R. Goldman**

**November 1990  
CSD-900036**



**Towards a Computational Model of  
Concept Acquisition and Modification  
Using Cases and Precedents from Contract Law**

Seth R. Goldman

February 1989

Technical Report UCLA-AI-89-03



# Towards A Computational Model of Concept Acquisition and Modification Using Cases and Precedents from Contract Law

Seth R. Goldman<sup>1,2</sup>

Artificial Intelligence Laboratory  
3531 Boelter Hall  
University of California, Los Angeles  
Los Angeles, CA 90024  
seth@cs.ucla.edu (arpanet)

## Abstract

Most research efforts in Case-Based Reasoning (CBR) focus on either indexing and retrieving episodes or how to map a previous episode on to a new situation once the retrieval is accomplished (Goldman, 1986; Goldman, Dyer, & Flowers, 1985a, 1985b; Kolodner, Simpson, Sycara-Cyranski, 1985; Kedar-Cabelli, 1984; McCarty & Sridharan, 1981). This research has relied upon using predefined indices and has ignored issues involved with learning new indices and reevaluating existing indices.

In this paper we begin to explore these issues, namely, how indices are acquired and refined through experience. The domain for our exploration is Contract Law and the setting is a first-year Contracts course. It is interesting to note, however, that our description of how new indices are acquired and refined also applies to practicing lawyers when they are operating in an unfamiliar area of the law. We shall discuss our ideas as extensions to an existing computer program, STARE, which can read and reason about a small number of contractual situations.

## 1. Introduction

In order to study how legal concepts are acquired and how lawyers learn to organize their legal knowledge we must go to law school. The purpose of law school is not to teach its students the law, that is, a static body of rules which can be memorized and applied. The law is far from static, it is an evolving institution concerned with the economical and equitable resolution of society's disputes. Law schools strive to teach their pupils how to think about the law. What is the purpose of our legal system? What are the legal procedures? How do individual decisions fit in with the body of prior cases? Learning how to answer these questions will enable the future lawyer to act intelligently in situations where the law is vague and act responsibly and effectively in situations where the law is relatively settled. The most popular method for teaching these skills is the case method.

## The Case Method

The case method operates by providing the students with a brief definition of some new legal concept which is then refined via a series of illustrations. The process continues incrementally

---

<sup>1</sup> This research was supported by a grant from the Louisiana State Board of Regents through the Louisiana Education Quality Support Fund, Contract #LEQSF(86-87)-UNEXP-3.

<sup>2</sup> This paper appeared in the Proceedings of the 5th Israeli Symposium on Artificial Intelligence, December 1988, in Tel Aviv.

throughout the student's legal life. Each new case adds more to the student's previous knowledge. It is important to realize that a single case does not provide enough information to define a legal concept. Therefore a computational model must always be ready to modify its existing knowledge structures as each new case is processed.

## 2. An Example

The best way to present our model of knowledge acquisition is through an example scenario. For our example we shall explore the presentation of two related concepts: *substituted contract* and *accord and satisfaction*. The steps involved are:

1. Build an initial representation for the concepts from the definitions.
2. Index these representations in memory.
3. Read some illustrations (examples) and incorporate them into the previously constructed concepts.
4. Read some cases (from a casebook) and brief<sup>3</sup> them.

The following definitions appear in the 2nd Restatement of Contracts (Restatement, 1981):

### § 279. *Substituted Contract*

1. A substituted contract is a contract that is itself accepted by the obligee in satisfaction of the obligor's existing duty.
2. The substituted contract discharges the original duty and breach of the substituted contract by the obligor does not give the obligee a right to enforce the original duty

### § 281. *Accord and Satisfaction*

1. An accord is a contract under which an obligee promises to accept a stated performance in satisfaction of the obligor's existing duty. Performance of the accord discharges the original duty.
2. Until performance of the accord, the original duty is suspended unless there is such a breach of the accord by the obligor as discharges the new duty of the obligee to accept the performance in satisfaction. If there is such a breach, the obligee may enforce either the original duty or any duty under the accord.
3. Breach of the accord by the obligee does not discharge the original duty, but the obligor may maintain a suit for specific performance of the accord, in addition to any claim for damages for partial breach.

Ignoring for now the language issues involved in building a conceptual representation from the above English definitions, we turn our attention to what the conceptualizations look like. The first question is: What does STARE already know when it reads these definitions? Clearly STARE needs representations for: contract, obligor, obligee, duty, performance, enforce, breach, suit, damages, satisfaction, and right.

---

<sup>3</sup> Briefing a case means writing a summary which details the participants, the dispute, and the outcome.

## Representation

We represent these concepts using frame-like structures called *schemas*. Schemas are used to organize other schemas as well as goals, plans, and events (Schank & Abelson, 1977). The latter are connected using intentional links (I-links) (Goldman, 1986; Dyer, 1983). Schemas are indexed using a strategy derived from (Kolodner, 1984). The dynamic organization of schemas is adapted from (Schank, 1982). As an example let's consider the schema for contract:

### M-CONTRACT<sup>3</sup>

**Participants:** offeror, offeree

**Properties:** what offeror will do (cons-a), offeree will do (cons-b)

**Scenes:** M-OFFER, M-ACCEPT

**Preconditions:** offeree reasonably believes that offeror has a *right* to extend the offer, offeror reasonably believes that offeree has a *power* to accept the offer, offeree reasonably believes that offeror can do cons-a, offeror reasonably believes that offeree can do cons-b}.

**Expectations:** offeror has a *duty* to offeree to do cons-a, offeree has a *duty* to offeror to do cons-b

The parameters of the schema are specified by the participants and properties slots. The scenes of the schema represent an ordering of the events which cause the schema to be instantiated. The preconditions verify that the schema should be instantiated. The expectations make predictions about what will happen after the schema is instantiated. Consider the following example:

A offers to deliver a tractor to B on July 1 for \$5000. B agrees to pay A the \$5000.

The offeror is A and the offeree is B. CONS-A is the ATRANS<sup>4</sup> of the tractor to B. CONS-B is the ATRANS of the \$5000 to A. The expectations are that A now has a *duty* to deliver the tractor by July 1 and that B now has a *duty* to pay A \$5000 when the tractor is delivered.

*DUTY* is a legal relation which relates two participants and a particular task. It derives from the Hohfeldian primitives (Hohfeld, 1913, 1917) put forth by Wesley Hohfeld in an attempt to improve the precision and accuracy of legal writings. The other Hohfeldian relations include: right, privilege, no-right, power, immunity, liability, and disability. A's duty in the above example may be represented as:

---

<sup>3</sup> All the schemas are described using a mixture of English and pseudo-code predicates.

<sup>4</sup> ATRANS, MTRANS, and others are primitives from Conceptual Dependency (CD) theory and are used for representing physical actions (Schank, 1975).

## M-DUTY

**Participants:** obligor(B), obligee(A), authority(local-government)

**Properties:** task(A ATRANS tractor to B), source(M-CONTRACT)

**Expectations:** A ATRANS tractor to B satisfies the duty, A NOT ATRANS tractor to B results-in B petitioning local-government to enforce contract

*Obligor* and *obligee* are represented as humans who fill the appropriate participant role in an as yet unspecified *duty*. Similarly *performance* and *breach* refer to the doing or not doing of some action which appears as the cons-a or cons-b of a contract. More details on the representation used in STARE can be found in (Goldman, Dyer, & Flowers, 1987; Goldman, 1986).

### **Building the Initial Schema**

These definitions must now be turned into schemas and integrated into memory. We have chosen to ignore the natural language issues for the present. Currently we construct the initial knowledge structures manually. The schema for substituted contract is built as:

## M-SUBSTITUTED-CONTRACT

**Participants:** offeror, offeree

**Properties:** cons-a, cons-b, new-cons-b

**Scenes:** M-CONTRACT, M-OFFER, M-ACCEPT

**Preconditions:** M-CONTRACT(offeree, offeror, cons-a, cons-b), M-OFFER(offeror, offeree, new-cons-b instead of cons-b), M-ACCEPT(offeree, offeror, new-cons-b instead of cons-b)

**Expectations:** DUTY(offeror, new-cons-b) SUBSUMES DUTY(offeror, cons-b), M-CONTRACT(offeror, offeree, new-cons-b, cons-a)

There are two parties to a substituted contract: offeror and offeree. Typically the offeror in the substituted contract was the offeree in the original contract. This is represented by the precondition: M-CONTRACT(offeree, offeror, cons-a, cons-b). CONS-A and CONS-B are what each party was supposed to do in the original contract. NEW-CONS-B is what the offeror would do instead of CONS-B.

The offeree's acceptance of the offeror's promise to do NEW-CONS-B instead of CONS-B extinguishes the original duty to do CONS-B and creates a new duty to do NEW-CONS-B. This is indicated by the subsumption relationship between the original duty and the new one. SUBSUMES is one type of link we need to connect legal relationships. Others will be introduced as they arise.

Now let's compare M-SUBSTITUTED-CONTRACT with M-ACCORD-AND-SATISFACTION. Clearly there is some relationship between the two concepts (otherwise why would they be presented together in the case book?). How does this relationship manifest itself in our representation?



## M-ACCORD-AND-SATISFACTION

**Participants:** offeror, offeree

**Properties:** cons-a, cons-b, new-cons-b

**Scenes:** M-CONTRACT, M-OFFER, M-ACCEPT

**Preconditions:** M-CONTRACT(offeree, offeror, cons-a, cons-b), M-OFFER(offeror, offeree, DO new-cons-b), M-ACCEPT(offeree, offeror, offeror DO new-cons-b)

**Expectations:** DUTY(offeror, new-cons-b) SUSPENDS DUTY(offeror, cons-b), M-CONTRACT(offeror, offeree, new-cons-b, cons-a)

The primary difference between the two concepts as presented in their definitions is that in a substituted contract, the offeree accepts the offeror's promise to perform new-cons-b as fulfillment of offeror's duty to perform cons-b. In an accord, the offeree displays willingness to accept the performance of new-cons-b as fulfillment of the original duty to perform cons-b. In the first instance, the duty to perform new-cons-b subsumes the original duty. In the second, the duty to perform new-cons-b suspends the original duty.

### **Memory Organization**

Now that we have an initial structure for each schema, how are we going to index them into memory in such a way as to capture their similarities and differences? We employ a hierarchical organization in the style of (Kolodner, 1984). Before giving an example, we must add some detail to the representations of the previous section to provide the necessary material for creating indices.

The representation of the preconditions and expectations for M-SUBSTITUTED-CONTRACT and M-ACCORD-AND-SATISFACTION were somewhat oversimplified. The actual M-OFFER precondition for the former should be:

- M-OFFER(offeror, offeree, new-cons-b, cons-a')
- NEW-CONS-B is MTRANS(offeror, offeree, DO-ACT(x))
- CONS-A' is CONS-A plus DISCHARGES(MTRANS(offeror, offeree, DO-ACT(x)), DUTY(offeror, DO-ACT(y)))
- DO-ACT(y) was the object of the original CONS-B

The missing expectation is that:

- M-SUBSTITUTED-CONTRACT(offeror, offeree, new-cons-b, cons-a) --- SUBSUMES---> M-CONTRACT(offeree, offeror, cons-a, cons-b)

Schemas are distinguished by their participants, properties, preconditions, scenes, expectations and dimensions. For example, both M-SUBSTITUTED-CONTRACT and M-ACCORD-AND-SATISFACTION are contracts and are indexed under M-CONTRACT. We use their participants and properties to distinguish them from M-CONTRACT. However this is not adequate for distinguishing them from each other. They also share the same scenes. Examining their expectations we now find differences, the main one being that acceptance of the new offer suspends the previous duty in one instance and subsumes it in the other. When

two schemas which are indexed at the same level share information, it is often useful to build a new schema which contains the shared items and reindex the old schemas under the new schema. In addition to conserving memory space, the new schema provides storage for knowledge which applies to all schemas it indexes.

Non-adjacent schemas are also indexed using shared information. One reason is so that changes in one structure can be reflected in others along the lines described in (Schank, 1982). For example, all contracts, including M-SUBSTITUTED-CONTRACT and M-ACCORD-AND-SATISFACTION, depend upon the concept of acceptance.

What does it mean to ACCEPT an OFFER? If our understanding of acceptance changes then this will have an impact upon every contract-related concept in memory. Another reason is to facilitate cross-contextual retrieval. Suppose that there are two cases involving a contract to obtain an illegal object. It is quite possible that different charges were made and the courts, which can only decide the questions before them, produced different decisions. We need to be able to retrieve other situations based upon shared features. If the indices are not built when the case is initially stored, it will be difficult to find it without doing an exhaustive memory search.

### Refinement Using Illustrations

Once the schemas are positioned in memory, we need to test their effectiveness as indexing structures. Illustrations are positive examples provided in the case book to clarify the new concepts in the students' minds. Expectation failures while reading these examples usually indicate a fault in the initial representation and are handled quite differently from the expectation failures arising from unfamiliar situations. Consider the following illustrations:

1. A is under a duty to deliver a tractor to B on July 1. On June 1, A offers to deliver a bulldozer to B on July 1 if B will accept his promise in satisfaction of A's duty to deliver the tractor, and B accepts. The contract is a substituted contract. A's duty to deliver the tractor is discharged. If A does not deliver the bulldozer, B can enforce the duty to deliver it but not the original duty to deliver the tractor.
2. A owes B \$10,000. They make a contract under which A promises to deliver to B a specific machine within 30 days and B promises to accept it in satisfaction of the debt. The contract is an accord. A's debt is suspended and is discharged if A delivers the machine within 30 days.

Reading the positive example permits us to check the roles, preconditions, and expectations of our schema. Missing roles may be added. Unnecessary roles may be deleted. Violated preconditions and expectations must be modified to handle the example or removed. Typically the action taken in light of an expectation violation is much more definitive than usual because we have the extra information that this situation is supposed to be a positive instance.

To see how this works, let's look at the first example above. The offeror is B and the offeree is A. We have DUTY(b, a, ATRANS tractor). From this we can infer M-CONTRACT(a, b, DO-ACT(?x), ATRANS tractor)<sup>5</sup>. There is also M-OFFER(B, A, ATRANS bulldozer) and M-ACCEPT(A, B, M-OFFER). Finally we have an outcome which confirms the expectations associated with M-SUBSTITUTED-CONTRACT. There don't seem to be any violations so we can assume that our initial construction was not faulty.

---

<sup>5</sup> This raises an interesting point: Can we always infer a contract if there is a duty? Can there be a duty without a contract?

## Expanding and Refining Dimensions

Dimensions are the parameters of a case which can take on different values and produce a different outcome. Law students spend much of their time learning what the dimensions of legal concepts are and their ranges. Dimensions are modified by every new case the student reads as well as by expectation failures. Whenever the student reads a new case involving a familiar concept (at least one other case in memory) then the dimensions of that concept must be updated. If the case has a similar outcome, then the range of the dimensions is expanded to incorporate the values from the new case. If the outcome is different, then the ranges must be modified to exclude those particular values.

When a situation fails to conform to our expectations we must take action so that future similar failures are meaningful. The first time a particular expectation fails we index the situation using the failure and there is little else we can do. The next time that expectation is violated we can get to the previous failed episode and begin trying to explain the failure. This is the approach proposed in (Schank, 1982). Processing the failure may involve:

1. Modifying the dimensions to exclude the given situation
2. Modifying the scene ordering
3. Building a new schema which generalizes from other schemas
4. Building a new schema which specializes from other schemas
5. Creating a new schema to index the situation
6. Creating new dimensions

The first two repair mechanisms correspond to what Schank calls *learning the variables*. The law student needs to learn which aspects of a concept are static and which aspects are fixed. Once fixed aspects have been determined, the student is then free to use the next two repair methods to generalize and specialize the knowledge structures as needed. When these methods fail to explain a violation, we are dealing with a new concept and we need to build a new schema.

## Briefing A Case

In (Llewellyn, 1960), Karl Llewellyn summarizes the importance of briefing cases:

Briefing should begin *at the earliest* with the second case of an assignment. Only *after* you have read the second case have you any idea what to do with the first. Briefing, I say again, is a problem of putting down what in the one case bears upon the problem stated by the other cases. Each brief should be in terms of *what this case adds to what I already know about* this subject. Hence at least two cases must be read before any can be intelligently briefed.

Our research is progressing to the point where we can start considering the processes involved in preparing a brief and how to model them. A brief must contain: the participants, what the dispute is about (who wants what), what the trial court decided (most cases in the case book are at the appellate level), what aspect of the trial court's behavior is being challenged, the judgement, the reasoning behind the judgement as stated in the judicial opinions, and the facts of the situation.

We can currently represent many of these components but have yet to work out all the reasoning processes involved. If we ignore the reasoning of the court in reaching its conclusion we can try to integrate the case into memory using just the facts, dispute, and result. In fact, STARE does something very similar to this now. The problem is that we need a better model of how memory reacts to new cases and we need methods for representing and incorporating the judicial opinions into the process. The opinions can be of great help in focussing on particular aspects of a case and deciding what is important to look for in future situations.

### 3. Conclusions

Building a system to understand and learn about Contract Law requires a dynamic memory organization which stores previous experiences and can be modified and reindexed as a result of processing new cases. The key to this flexible type of memory organization is a uniform representation for higher level concepts. Using schemas to represent concepts, dimensions, and episodes allows us to use the same mechanism to make generalizations and specializations at different conceptual levels. We are applying these ideas to extend the capabilities of our program, STARE, to briefing cases.

### References

- \_\_\_\_\_. (1981). *Restatement of the Law, Second: Contracts 2d*. American Law Institute Publishers, St. Paul, MN.
- Dyer, M.G. (1983). *Artificial Intelligence Series: In-Depth Understanding: A Computer Model of Integrated Processing For Narrative Comprehension*. MIT Press, Cambridge, MA.
- Goldman, S.R., Dyer, M.G., and Flowers, M. (1987). "Precedent-based Legal Reasoning and Knowledge Acquisition in Contract Law: A Process Model." In *Proceedings of the First International Conference on Artificial Intelligence and Law, ICAIL-87*. Boston, MA.
- Goldman, S.R. (1986). *A Process Model of Precedent-based Legal Reasoning and Knowledge Acquisition in Contract Law*. Masters Thesis. University of California, Los Angeles.
- Goldman, S.R., Dyer, M.G., and Flowers, M. (1985a). "Learning To Understand Contractual Situations." In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*. Los Angeles, CA.
- Goldman, S.R., Dyer, M.G., and Flowers, M. (1985b). "Representing Contractual Situations." In *Proceedings of the Second Law and Technology Conference*, Houston, TX.
- Hohfeld, W.N. (1913). "Some Fundamental Legal Conceptions As Applied in Judicial Reasoning." *Yale Law Journal*. 23:16-59.
- Hohfeld, W.N. (1917). "Fundamental Legal Conceptions As Applied in Judicial Reasoning." *Yale Law Journal*. 710-770.

- Kedar-Cabelli, S. (1984). *Analogy with Purpose in Legal Reasoning from Precedents: A Dissertation Proposal*. Technical Report LRP-TR-17, Laboratory for Computer Science, Rutgers University.
- Kolodner, J. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Lawrence Erlbaum Associates. Hillsdale, NJ.
- Kolodner, J., Simpson, R.L. Jr., Sycara, K. (1985). "A Process Model of Cased-Based Reasoning in Problem Solving." In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, (IJCAI-85)*. Los Angeles, CA.
- Llewellyn, K. (1960). *The Bramble Bush*. Oceana Publications, Inc. Dobbs Ferry, NY.
- McCarty, L.T. and Sridharan, N.S. (1981). *A Computational Theory of Legal Argument*. Technical Report LRP-TR-13, Laboratory for Computer Science Research, Rutgers University.
- Schank, R.C. (1982). *Dynamic Memory*. Cambridge. Cambridge University Press.
- Schank, R.C. & Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum.
- Schank, R.C. (1975). *Conceptual Information Processing*. North Holland. Amsterdam.

