1980 TOC Search

9

Representation of knowledge in a legal information retrieval system

Carole D. Hafner

9.1 Introduction

Recent interest in computer representation of knowledge has led to the development of 'expert' computer assistants for tasks such as medicaldiagnosis, technical instruction, and problem solving in restricted domains (Brown and Burton, 1975; Davis, Buchanan and Shortliffe, 1977; Goldstein and Roberts, 1977). Each of these systems is based on a semantic model of a specific subject area, along with some general methods for using subject area knowledge to understand and respond to users' requests. The emphasis of these systems is not on 'solving' problems by computer, but rather on helping a human problem solver organise and apply a complex body of knowledge. The research described here explores the use of subject area knowledge in an 'expert' document retrieval system. The goals of the research are to characterise the semantics of information retrieval requests, and to develop methods for representing and using subject area knowledge in computer retrieval systems.

The Legal Research System (LRS) is a knowledge-based computer retrieval system, intended to be used by lawyers and legal assistants to retrieve information about court decisions (cases) and laws passed by legislatures (statutes). The subject of its knowledge is Negotiable Instruments Law, an area of Commerical Law that deals with cheques and promissory notes (White and Summers, 1972; Speidel, Summers and White, 1974). The current implementation of the system (Hafner, 1978) has a database of about 200 statutes from the Uniform Commerical Code (American Law Institute, 1972) and 200 related cases. In LRS four kinds of knowledge about legal concepts and relationships are represented: functional knowledge, structural knowledge, semantic knowledge and factual knowledge.

In this chapter the motivation for including each kind of knowledge is discussed, the computer representation of each kind of knowledge is described and examples of the use of each kind of knowledge in LRS are presented. The next section gives a very brief overview of current legal retrieval systems, both manual and automated. Subsequent sections describe the representation of knowledge in LRS, and the use of this knowledge to understand and interpret user queries.

9.2 Legal information retrieval

Searching a large database is an important and time-consuming part of legal work. A lawyer is frequently faced with a complex situation that must be analysed to determine the rights and liabilities of the parties involved. In order to do this, statutes that apply to the situation and cases that dealt with similar situations must be located. The process of legal advocacy requires a conscientious lawyer to conduct an exhaustive search of the literature. Even if a favourable case or statute is found that is exactly 'on point', there may be other conflicting authorities. Even if there are not, the opposing attorneys will try to find some reason why the favourable case or statute should not be applied to the current situation, and will introduce alternative cases or statutes that would lead to a different result. A lawyer must try to out-manoeuvre the opposing side by anticipating the alternative lines of argument, finding the authorities that might be used to support them and preparing to argue against these authorities.

Many law firms hire legal assistants, who spend most of their time looking up cases and statutes. There are also legal research consulting firms that can be hired to investigate a particular problem. A number of bibliographic aids are available; without them, the task would be hopeless, since thousands of cases are recorded each year (National Association of Attorneys General, 1976). Legal encyclopedias such as *American Jurisprudence* (1967) provide textbook-style overviews of major legal topics, giving numerous citations of cases and statutes that illustrate each point. The *West Digest* system (1966) is a multi-volume collection of case summaries, organised according to a hierarchical network of 'key word' subject descriptors. *Shepard's Citations* (1977) contains cross-reference lists of cases; the name of each case is followed by a list of all subsequent cases in which the case was referenced. 'Shepardising' is a good way to find recent decisions on an issue and to avoid the embarrassment of citing a case that is no longer followed.

It is easy to see that the legal research database is an excellent candidate for an automated retrieval system, and there are several systems currently available (described in *WESTLAW: The Computerized Legal Research System*; *LEXIS: A Primer*, 1975; National Association of Attorneys General, 1976). The best-known is LEXIS (1975), a large centralised interactive system with specialised terminals and excellent display capabilities. LEXIS is a full text retrieval system; the user constructs a Boolean expression defining sets of words that must or must not appear in the text of the items retrieved. Constraints on how close together or how far apart pairs of words must be in the text can also be specified. LEXIS retrieves all items in the database that satisfy the user's request, and the text of these items can be viewed on-line.

LEXIS, like other keyword systems, provides reasonably easy access for the untrained user. However, like other key word systems, it suffers from a lack of expressive power. The user must translate a meaningful query into a Boolean function of word occurrence — a difficult, if not impossible, task in many cases. In the following sections some alternatives to the key word retrieval paradigm are suggested, and the additional expressive power that they provide is demonstrated.

9.3 Functional knowledge

When users of a retrieval system request information about a particular subject, they usually do not want everything that is known about the subject but only those items that fill a specific need. For example, a medical researcher might request one of several different kinds of information about penicillin: its chemical composition, its effect on various diseases, its recommended dosages or its potential for negative side-effects. The 'expert' retrieval system should know not only what concepts appear in a data item, but also what kinds of information the data item contains about those concepts.

For a particular application area, the kinds of information, or 'information functions', represented in a retrieval system should correspond to the information needs of its intended users. For example, in a medical retrieval system some information functions for pharmacological data might be: chemical composition, medical uses and physiological action. Each of these functions takes an 'argument', which is a description of a type of drug. Its 'value' is the set of data items that contain information of the type specified about the drug specified. Information functions can provide a retrieval system with a simple but important kind of subject area expertise.

In the Legal Research System each item in the database is indexed by a set of data descriptions, representing the system's knowledge about the item. A data description consists of: (1) an information function and (2) a description of a legal concept.

The information function tells what kind of information the item provides about the concept. In LRS there are eight information functions:

(1) (PLAINTIFF D) — The plaintiff of a case was a 'D'. D must describe a party — for example, the payee of a cheque.
(2) (DEFENDANT D) — The defendant of a case was a 'D'.
(3) (CAUSE-OF-ACTION D) — The legal basis of a case was 'D' — for example, negligence.
(4) (EXAMPLE D) — The fact situation of a case was an example of 'D' — for example, a forged promissory note.
(5) (HYPOTHETICAL EXAMPLE D) — A case or statute describes a hypothetical situation that is an example of 'D'.
(6) (CRITERIA D) — A case or statute defines criteria for a situation to be an example of 'D'.
(7) (LEGAL-EFFECT D) — A case or statute describes the legal consequences of 'D'.
(8) (RULE D) — A case ruled that the situation before the court was an example of 'D'.

These information functions correspond to the needs of legal researchers; important relationships that are difficult to describe in key word retrieval systems can be represented directly in LRS. The general nature of a lawsuit is identified by the PLAINTIFF, the DEFENDANT and the CAUSE-OF-ACTION. A lawyer can request cases where a depositary bank sued a drawee bank for unjustified delay in processing a forged cheque, and the system will find exactly those cases. Or the lawyer can ask for statutes that define the criteria for being the payee of a cheque. The system will retrieve those statutes, and not statutes that define the legal effects of being the payee of a cheque.

```
(PLAINTIFF (DRAWER CHECK))
(DEFENDANT (ISA (DRAWEE CHECK) BANK))
(CAUSE-OF-ACTION C-IMPROPER-DEDUCTION)
(EXAMPLE IMPROPER-DEDUCTION)
(EXAMPLE (DRAWER'S-SIGNATURE FORGED))
(EXAMPLE (NOT EXAMINED-PROMPTLY))
(LEGAL-EFFECT (DRAWER'S-SIGNATURE FORGED))
(LEGAL-EFFECT IMPROPER-DEDUCTION)
(LEGAL-EFFECT (DRAWER (NEGLIGENT (CONTRIBUTING-TO
                UNAUTHORIZED))))
(LEGAL-EFFECT (NOT REASONABLE-CARE-TAKEN))
(RULE C-IMPROPER-DEDUCTION)
(RULE ((NOT REASONABLE-CARE-TAKEN)
        (ISA (DRAWEE CHECK) BANK)))
(RULE (DRAWER ((NOT NEGLIGENT) (CONTRIBUTING-TO
        (DRAWER'S-SIGNATURE FORGED)))))
```

Figure 9.1 Data description for *Jackson v. First National Bank of Memphis Inc.*, 403 SW 2d 109 (1966)

The facts of a case (which are the basis of a legal precedent) are distinguished from hypothetical situations invented for explanatory purposes (which represent legal opinion, but are not the basis of a precedent) by the EXAMPLE and HYPOTHETICAL-EXAMPLE functions. A lawyer would probably try to find real examples first, and look for hypothetical examples only if there were not enough real examples or if the cases containing real examples were not favourable.

The legal issues debated in a case are distinguished from the facts that were not at issue by the RULE function. For example, there are a great many cases involving forged cheques where the forgery is not at issue — the question is: who should bear the loss caused by the forgery? Cases of this type have (EXAMPLE FORGED) descriptors but not (RULE FORGED) descriptors. In situations where forgery is an issue a lawyer can easily retrieve cases that dealt with that issue by requesting descriptors of the form (RULE FORGED).

Figure 9.1 shows the description of a sample case from the LRS data base. *Jackson v. First National Bank of Memphis, Inc.* was a suit by the drawer of a cheque against the drawee bank, for improperly paying a cheque on which the drawer's signature was forged. If the plaintiff wins, the bank will be required to re-credit his account. The bank presented two arguments in its defence: first, that the plaintiff had been negligent and his negligence contributed to the forgery; and second, that the plaintiff had failed to examine his monthly statement promptly, and thus had delayed in notifying the bank of the forgery beyond the time where he could recover the money. The court held that the plaintiff could recover the improper deduction, that the bank did not exercise reasonable care in handling the cheque and that the plaintiff was not negligent.

9.4 Structural knowledge

The most important kind of knowledge in an information retrieval system resides in the subject descriptors, which link each data item to the subject area concepts the item is about. In the Legal Research System subject descriptors

```
SD ::= CD | (CD SD) | ("ISA" SD SD)

CD ::= Concept | ("NOT" Concept) | ("NOTHAVE" Concept)
```

Figure 9.2 Structured description language syntax

are coded in a formal language, in which complex descriptions can be built from basic subject area concepts. By using a structured language for subject descriptors we can represent relationships among concepts, as well as the concepts themselves.

For example, in a database of newspaper crime clippings, there might be a story about the child of a murderer, a story about the murderer of a child and a story about a child who is a murderer. A retrieval system that only allows subject descriptors such as 'child' and 'murderer' without providing a way to describe the relationships between them would be incapable of distinguishing these three stories. An actual example of this limitation was encountered during a LEXIS retrieval session*. The user wanted to find cases that determined the amount of a doctor's liability for negligence, when the negligence caused only part of a patient's injury. Most of the cases retrieved by LEXIS, unfortunately, dealt with the amount of injury that must be caused by a doctor's negligence in order for the doctor to have any liability. The concept 'amount of liability for injury' could not be distinguished from 'amount of injury for liability'.

In the Legal Research System subject descriptors are coded in a Structured Description Language (SDL), which is capable of expressing dependency relationships such as 'the amount of liability for injury' and identity relationships such as 'a child who is a murderer'. Although the syntax of SDL is simple (its grammar is shown in *Figure 9.2*), it can be used to build descriptions of arbitrary complexity.

Descriptions involving only one subject area concept are called concept descriptors (CD). A 'concept' is a single-word descriptor, such as INDORSEMENT†, representing a basic subject area concept. (NOT concept) and (NOTHAVE concept) represent negative instances of a concept; for example, (NOT INDORSEMENT) represents a signature that is not an endorsement and (NOTHAVE INDORSEMENT) represents a negotiable instrument that does not contain any endorsements.

Simple descriptions are combined into more complex expressions, as shown in the definition of a structured descriptor (SD). Expressions of the form (CD SD) represent dependency relationships such as (AMOUNT (LIABILITY (INJURY CAUSED-BY-NEGLIGENCE))). The concept contained in the first element (CD) is modified by the rest of the description. By use of the order of terms and the scope of parentheses, different complex concepts composed of the same basic concepts can be expressed. Expressions of the form (ISA SD SD) represent identity relationships, indicating a situation where two descriptions apply to the same entity. For example, (ISA (PARTY-NAMED (INDORSEMENT FORGED)) (OFFICER BANK)) describes a bank officer whose indorsement was forged.

* Peter Weidenbeck, University of Michigan Law School, personal communications.

† Descriptors are given in the American spelling as used by the system.

The following are some English descriptions of complex legal concepts, and their SDL equivalents:

(1) The holder of a cheque on which an endorsement was forged. (HOLDER (ISA CHECK (THEINST (INDORSEMENT FORGED))))
(2) The holder of a cheque whose endorsement was forged. (ISA (HOLDER CHECK) (PARTY-NAMED (INDORSEMENT FORGED)))
(3) A person who signed an endorsement in someone else's name. (ISA (SIGNER INDORSEMENT) ((NOT PARTY-NAMED) INDORSEMENT))

Although SDL expressions do not look very much like English, they perform some of the same functions as English descriptions. Structured modifiers can be used to simulate the action of adjectives and prepositional phrases. Negative instances of a concept can be specified, and combined with other concepts in complex ways. The 'ISA' construction does the work of a relative clause. Using these tools, researchers can describe the information they are looking for with some precision.

9.5 Semantic knowledge

The structures introduced in the previous sections are powerful descriptive tools; but they do not save the user from having to know (or guess) the exact terms used in the database to describe the items of interest — and having to specify all of them. This requirement (unavoidable in full text retrieval systems) makes the process of query construction difficult and tedious, and produces long queries with many 'or' clauses.

To illustrate this difficulty, we can consider the following relationships from Negotiable Instruments Law:

(A) An endorsement is a signature that appears on a negotiable instrument.
(B) A cheque is a type of draft, which is a type of negotiable instrument.
(C) A forged signature is a type of unauthorised signature.
(D) In order to be the 'holder' of a negotiable instrument, a party must be in possession of the instrument.

A researcher who wanted to retrieve information about a complex situation, such as 'an unauthorised signature of the party in possession of a draft', would have to specify the following additional descriptions (and many more) to ensure a complete answer:

(1) An unauthorised endorsement of the party in possession of a draft.
(2) An unauthorised endorsement of the party in possession of a cheque.
(3) A forged signature of the party in possession of a draft.
(4) A forged endorsement of the party in possession of a draft.
(5) An unauthorised signature of the holder of a draft.
(6) An unauthorised signature of the holder of a cheque.
(7) A forged signature of the holder of a draft.
etc.

In the Legal Research System a semantic network model of Negotiable Instruments Law is used to make simple inferences about the meanings of

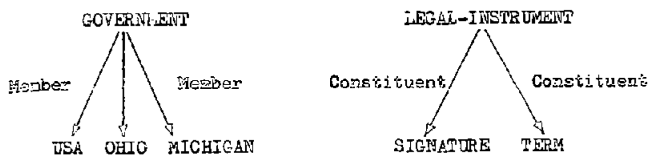


Figure 9.3 Set/member and constituent links

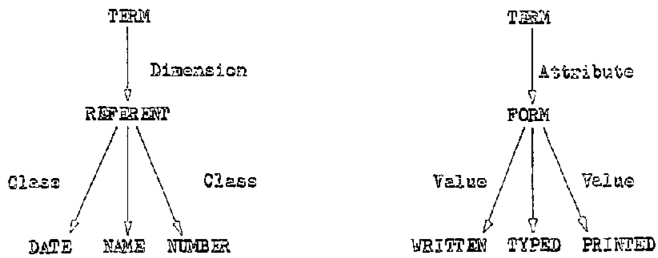


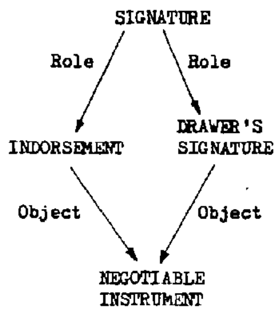
Figure 9.4 Superclass/sub-class and attribute links

queries. Legal relationships such as (A)–(D) above are encoded in the network, and are used to automatically retrieve data items with descriptions such as (1)–(7) in response to a query about 'an unauthorised signature of the party in possession of a draft'.

The semantic network model is a collection of *nodes* defining subject area concepts, such as 'holder', 'forged' and 'endorsement', connected by *links* representing a small set of semantic structures, such as classification and attribution. Starting from a few 'undefined' basic concepts, a complex network structure can be built up, in which the meaning of each node (or subject area concept) derives in a well-defined way from its position in the network. Semantic networks have been used by Quillian (1968) to represent the meanings of words; by Simmons (1973) and Schubert (1976) to represent the meanings of English sentences; and by Schank (1972) and Nash-Weber (1975) to represent the subject area knowledge required for natural-language processing.

There are six link types in the LRS semantic network: set/member links, constituent links, super-class/sub-class links, attribute links, role links and event-condition links. The network contains six basic concepts, which are not defined within the semantic model itself: PARTY, LEGAL-INSTRUMENT, LIABILITY (a legal obligation), LEGAL-ACTION (a case), ACCOUNT and AMOUNT-OF-MONEY. More than 200 other legal concepts are defined in terms of these six, through one or more link connections.

Set/member links connect set concepts, such as 'integer', to their individual members, such as '1' and '2'. Constituent links connect concepts that have constituents, such as 'book', to their component concepts, such as 'cover' and 'page'. *Figure 9.3* shows a set/member link and a constituent link from the Negotiable Instruments model. Super-class/sub-class links connect concepts that are sets, such as 'numbers', to concepts that are sub-sets, such as 'integers' and 'reals'. Attribute links define attributes of concepts, such as 'colour', and their values, such as 'red' and 'blue'. *Figure 9.4* shows a super-class/sub-class link and an attribute link from the Negotiable Instruments model.



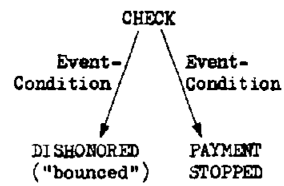


Figure 9.5 Role and event/condition links

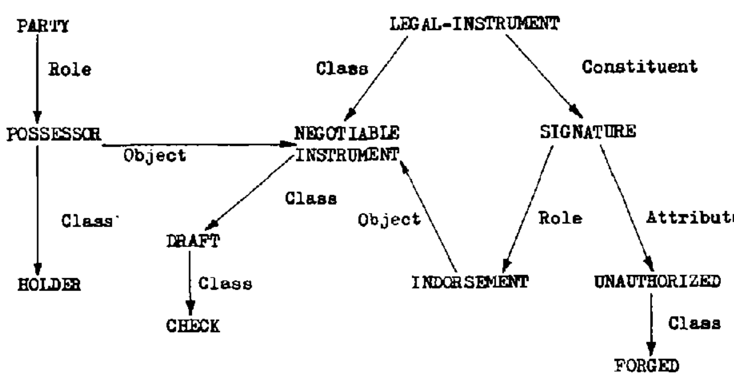


Figure 9.6 A portion of the LRS semantic network

Role and event-condition links are the most complex links in the network. A role link defines a concept such as 'author', which is a role taken by a person with respect to some book or paper. An event-condition link defines a concept such as 'married', which is an event that changes a person's status or condition. *Figure 9.5* shows role and event–condition links from the Negotiable Instruments model. *Figure 9.6* shows a combination of several link types in a larger portion of the network; this figure demonstrates the encoding of the information in statements (A)–(D) at the beginning of this section. The knowledge represented in the network is used by the Legal Research System to extend user queries to include terms that are implied but not mentioned by the user.

9.6 Descriptive deduction

Each item in the LRS database is indexed by a set of data descriptions, consisting of (1) an information function and (2) a subject descriptor. For example, the description (PLAINTIFF (PAYEE (THEINST FORGED))) describes a case whose plaintiff was the payee of a negotiable instrument containing a forged signature. The description (RULE (ISA

(INDORSEMENT NOTE) (NOT FORGED))) describes a case that decided that an indorsement on a promissory note had not been forged*.

Data descriptions of the same form are entered by the user as part of a search request, possibly connected by Boolean operators. For example, the query

FIND CASES (EXAMPLE DISHONORED) AND NOT (PLAINTIFF HOLDER)

requests cases that involved a dishonoured draft (for example, a bounced cheque) and whose plaintiff was not a holder. Each data description is handled as a separate request by the retrieval component of the system, and the items retrieved are placed in temporary sets for further processing†.

The main task of the LRS retrieval program is to decide when a subject descriptor D implies another subject descriptor D′, in the sense that a request for information about D implies a request for items with the description D′. This requires a procedure for making 'descriptive deductions'. Models of logical deduction define truth relationships among formulas — if we know that a formula F is true in some world, we can deduce that F′ is also true. A model of descriptive deduction defines applicability relationships among descriptions — if we know that a description D applies to some object, situation or event, we can deduce that D′ also applies. For example, from the description 'the holder of a cheque' we can deduce the description 'a party who is in possession of a draft'.

In an information retrieval system deduction operates in reverse: if the user asks for items with description D ('the defendant is a party in possession of a draft'), the system should locate items with descriptions D′ ('the defendant is the holder of a cheque'), such that D is deducible from D′. Thus, items with descriptions D′ 'satisfy' the user's request. In LRS this is done by computing the 'extension' of D.

Extension is defined as a relation among sets of Structured Description Language expressions: for any SDL expression D, the extension of D is the set of all SDL expressions D′ such that every situation described by D′ is also described by D. For example, (INDORSEMENT FORGED) is in the extension of UNAUTHORIZED, since every forged indorsement is an unauthorised signature. The definition of extension consists of two parts: the first, called qualification, is purely syntactic, while the second part uses knowledge encoded in the semantic network.

9.6.1 Qualification

The syntactic structures of SDL (shown in *Figure 9.2*) are used to combine simple concepts into complex descriptions of conceptual dependency and identity relationships. Dependency constructions (for example, (MURDERER CHILD) = 'a murderer of a child' and (CHILD MURDERER) = 'a child of a murderer') consist of a 'head' concept modified by restrictive qualifiers. Identity constructions (for example, (ISA CHILD

* In SDL a single attribute name (for example, FORGED) denotes an object that has the attribute (for example, for forged signature).

† Computing the answer to a Boolean query may require a series of unions, intersections and set exclusions to be performed on the sets of documents retrieved.

```
Let S represent any SDL expression.
For SDL expressions D and D', D' is a qualification of D if:

(a) D' = D.

(b) D' = (ISA S D) or (ISA D S).

(c) D = Concept, (NOT Concept), or (NOTHAVE Concept), and
    D' = (D S).

(d) D = (D1 S), and
    D' = (D1 S'), where S' is a qualification of S.

(e) D = (ISA D1 D2), and
    D' = (ISA D1' D2') or (ISA D2' D1'), where D1' and D2'
         are qualifications of D1 and D2, respectively.

(f) D' is a qualification of D", and D" is a qualification
    of D.
```

Figure 9.7 Qualification

MURDERER) = 'a child who is a murderer') consist of two head concepts that modify each other.

The qualification relation, shown in *Figure 9.7*, provides a way of recognising that one description is a more explicit variant of another description, so that a user's request for instances of a concept (for example, 'murderer') will result in the retrieval of its more explicit instances as well ('murderer of a child'). It is based on the idea that every SDL expression is a qualification of its head concept(s).

Figure 9.7 defines D′ to be a qualification of D if it consists of D plus zero or more modifiers. If D already contains modifiers, this principle is applied recursively (rules d and e in *Figure 9.7*). The definition says that (MURDERER CHILD) is a qualification of MURDERER (but not a qualification of CHILD), by rule c. (ISA MURDERER CHILD) is a qualification of both MURDERER and CHILD, by rule b. (MURDERER (CHILD POLITICIAN)) is a qualification of MURDERER, and a qualification of (MURDERER CHILD), by rules c and d.

9.6.2 Extension

The full definition of extension, shown in *Figure 9.8*, introduces expressions that do not contain any of the terms in the user's request but are related to it through the semantic network. For example, cheques are a sub-class of drafts, so CHECK is in the extension of DRAFT. TRANSFEREE (a party who acquired an instrument) has an attribute FOR-VALUE (indicating that the party gave something of value in exchange for the instrument); so FOR-VALUE is in the extension of TRANSFEREE. (NOT FOR-VALUE), indicating a transferee who did not give anything of value for an instrument, is also in the extension of TRANSFEREE.

The negations of roles, attributes and event-conditions are included in the extensions of the concepts that assume the roles, that have the attributes or that the event-conditions happen to. This follows from the presuppositions (Katz and Postal, 1964) of negative descriptions: to say that an object does not

```
For SDL expressions D and D', D' is in the extension of D if:

  (a) D' is a qualification of D.

  (b) D = Concept, and
      D' is a member, subclass, role, attribute, or
      event-condition of D.

  (c) D' = (NOT D"), and D" is a role, attribute, or
      event-condition of D.

  (d) D' = (NOTHAVE D"), and D" is a constituent of D,
      or a role or event-condition for which D is an object.

  (e) D = (D1 D2), and
      D' = (D1' D2'), where D1' and D2' are in the extensions
      of D1 and D2, respectively.

  (f) D = (ISA D1 D2)
      D' = (ISA D1' D2') or (ISA D2' D1'), for D1' and D2'
      in the extensions of D1 and D2, respectively.

  (g) D' is in the extension of D", and
      D" is in the extension of D.
```

Figure 9.8 Extension

have an attribute, or is not in a particular role or condition, presupposes that the object is of a type that could have the attribute or could be in the role or condition.

For example, in the semantic network model an attribute of NEGOTIABLE-INSTRUMENT is INCOMPLETE, describing an instrument that was signed while still incomplete. An event-condition of INCOMPLETE is COMPLETED, and an attribute of COMPLETED is ACCORDING-TO-AUTHORITY. ACCORDING-TO-AUTHORITY describes an instrument that was signed while incomplete but was completed by the holder in accordance with the original intention of the signer. If a user requests all cases that involved an incomplete draft — (DRAFT INCOMPLETE) — the system will locate cases involving an incomplete draft that was not completed — (DRAFT (NOT COMPLETED)) — as well as an incomplete draft that was completed — (DRAFT COMPLETED). It will also locate a cheque that was completed, but not according to the intention of the signer — (CHECK (NOT ACCORDING-TO-AUTHORITY)).

The negation of a sub-class does not have the same presupposition as the negations of roles and attributes. To say that an instrument is not a cheque does not necessarily imply that it is a draft, although cheques are a sub-class of drafts. Similarly, a party may fail to be a 'holder in due course' (a sub-class of holder) by failing to be a holder in the first place. We do not, however, say that an object fails to be forged because it is not a signature.

9.7 Factual knowledge

The LRS semantic neetwork structures (described in Section 9.5) represent knowledge that is definitional and therefore always true, like the knowledge that a bachelor is unmarried. The network cannot represent knowledge about factual relationships, from which more complex inferences may be made. For example, the network 'knows' that:

(a) PARTY-NAMED is a role taken by a PARTY with respect to a SIGNATURE.
(b) EFFECTIVE is an attribute of a SIGNATURE.
(c) FORGED is a sub-class of UNAUTHORIZED, which is an attribute of a SIGNATURE.
(d) RATIFIED is an event-condition of a SIGNATURE, whose agent is a PARTY.

But it does not know that:

(e) An unauthorised signature is ineffective unless it is ratified by the party named.

This limitation of the semantic model leads to a corresponding limitation in the system's deductive capability. In response to a request for items involving ineffective endorsements, the system cannot automatically retrieve items involving forged endorsements, excluding those that were ratified.

Legal rules such as (e) are not *necessarily* true (in the logical sense); they may vary from time to time, from place to place, and from case to case. Even for legal rules that are firmly established, there may be occasional exceptions. A semantic network model that could handle this kind of knowledge would have to include:

(1) Network structures to represent factual relationships such as 'implies' and 'unless'.
(2) A mechanism for handling uncertain and contradictory information.

In the Legal Research System factual knowledge about legal rules is represented, not in the semantic network but in the database. Along with each ruling of a case (represented as a RULE descriptor) is a description of the legal reasons given for the ruling. These legal 'facts' are encoded as modifiers of the RULE descriptors, in the following way:

(RULE D (REF D')...)
(RULE D (EREF D')...)

A RULE descriptor can contain zero or more REF or EREF modifiers following the SDL expression (D) that represents the result of the ruling. A REF modifier (REF D') says that the court decided D because D' was true. An EREF modifier (ERER D') says that the court decided D even though D' was true. For example, if a case decided that a signature was not effective because it was forged, the RULE descriptor would be:

(RULE (NOT EFFECTIVE) (REF FORGED))

If a case decided that a signature was effective even though it was forged, ebcause it was ratified, the descriptor would be:

(RULE EFFECTIVE (EREF FORGED) (REF RATIFIED))

A user of LRS who wanted to know about all possible circumstances in which a forged endorsement might be effective (even though forged) could ask for

```
(RULE C-IMPROPER-DEDUCTION
      (REF IMPROPER-DEDUCTION)
      (REF ((NOT REASONABLE-CARE-TAKEN)
              (ISA (DRAWEE CHECK) BANK)))
      (EREF (NOT EXAMINED-PROMPTLY)))
(RULE ((NOT REASONABLE-CARE-TAKEN)
       (ISA (DRAWEE CHECK) BANK)))
(RULE (DRAWER ((NOT NEGLIGENT) (CONTRIBUTING-TO
       (DRAWER'S-SIGNATURE FORGED)))))
```

Figure 9.9 RULE descriptors for *Jackson v. First National Bank of Memphis Inc.*, 403 SW 2d 109 (1966)

cases with a (RULE EFFECTIVE) descriptor containing an (EREF FORGED) modifier.

The use of modifiers in the LRS database is illustrated in *Figure 9.9*, which shows the complete set of RULE descriptors for *Jackson v. First National Bank of Memphis, Inc.* (These descriptors were shown in *Figure 9.1* without modifiers.) The first RULE descriptor says that the plaintiff would be able to recover the improper deduction even though he had not examined his statement promptly, because the bank did not exercise reasonable care in handling the cheuqe.

Encoding knowledge about legal rules in this way has several advantages and one major disadvantage. The advantages are:

(1) It enhances the retrieval capabilities of the system. The reasons for a court decision are just as important as the results of the decision. This approach provides a way of searching the database for a particular line of legal reasoning.
(2) It finesses the problem of representing inconsistent and contradictory results. Even a case that is self-contradicting can be handled easily by using several RULE descriptors.
(3) It represents factual knowledge in a way that is independent of any particular inference system. Various methods of representing legal rules and performing legal inference could be tried out without its being necessary to restructure the database.

The disadvantage is: knowledge represented in this way is diffused throughout the database and is not in a form in which it can be used directly for making inferences.

The problem of representing factual knowledge is not solved by attaching rules to individual data items; however, this kind of representation is a necessary component of an intelligent retrieval system, providing the 'raw data' from which a model of factual knowledge must be built.

9.8 Conclusions

The model of knowledge described in this chapter, while general in its applicability to different subject areas, is restricted in the kind of knowledge it represents. It is a 'classificational' model, based on the conceptual structures used to classify objects and situations and the linguistic structures used to describe them. The inclusion of roles and event-conditions reflects the fact that objects are classified according to their situational as well as their inherent characteristics. (This is particularly obvious in dealing with legal concepts.)

No attempt has been made to represent a complete set of semantic structures. On the contrary, the goal was to fins a small set of structures that would 'buy' a lot of descriptive power in an information retrieval context, but still be simple and compact enough to be the basis of a manageable, understandable computer system. This goal has, to a great extent, been achieved. By limiting the scope of the system's knowledge, it has been possible to model a complex domain of practical significance*, and to implement algorithms that 'understand' this domain, in a limited sense. (This is the less frequently explored side of the trade-off between depth of representation and breadth of subject matter that characterises most AI research today.)

The research described in this report is only the first step towards a formal characterisation of the relationship between knowledge representation and information retrieval. Certainly, a great deal more needs to be done before any conclusions about the practical value of this approach can be drawn. I hope that the work reported here will stimulate others to tackle this problem.

Acknowledgements

This research was supported by National Science Foundation grants DCR71-02038 and MCS77-00880.

References

American Jurisprudence (1962). 2nd edn, Bancroft-Whitney Co. and The Lawyers Co-operative Publishing Company, San Francisco

AMERICAN LAW INSTITUTE (1972). *Uniform Commercial Code*, American Law Institute and National Conference of Commissioners on Uniform State Laws, Philadelphia

BROWN, J. S. and BURTON, R. (1975). 'Multiple representations of knowledge for tutorial reasoning', in BOBROW, D. and COLLINS, A. (Eds), *Representation and Understanding: Studies in Cognitive Science*, pp. 311–349, Academic Press, New York

DAVIS, R., BUCHANAN, B. and SHORTLIFFE, E. (1977). 'Production rules as a representation for a knowledge-based consultation program', *Artificial Intelligence*, **8**, 15–45

GOLDSTEIN, I. and ROBERTS, R. (1977). 'NUDGE: a knowledge-based scheduling program', in *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 257–263, Cambridge, Mass.

HAFNER, C. (1978). *An Information Retrieval System Based on a Computer Model of Legal Knowledge*, Ph.D. Thesis, University of Michigan

KATZ, J. and POSTAL, P. (1964). *An Integrated Theory of Linguistic Descriptions*, MIT Press, Cambridge, Mass.

* The LRS database contains all of the statutes from Articles 3 and 4 of the Uniform Commercial Code, and all of the cases cited in the section on Negotiable Instruments of a standard textbook on Commerical Law (White and Summers, 1972).

LEXIS: A Primer (1975). Mead Data Central, New York.

NASH-WEBER, B. (1975). 'The role of semantics in automatic speech understanding', in BOBROW, D. and COLLINS, A. (Eds), *Representation and Understanding: Studies in Cognitive Science*, pp. 351–382, Academic Press, New York.

National Association of Attorneys General (1976). *Computerized Research in the Law*, Raleigh, N.C.

QUILLIAN, M. R. (1968). 'Semantic memory', in MINSKY, M. (Ed.) *Semantic Information Processing*, pp. 227–270, MIT Press, Cambridge, Mass.

SCHANK, R. C. (1972). 'Conceptual dependency: a theory of natural language understanding', *Cognitive Psychology*, 3, 552–631

SCHUBERT, L. K. (1976). 'Extending the expressive power of semantic networks', *Artificial Intelligence*, **7**, 163–198

Shepards United States Citations (1977). Shepards, Colorado Springs

SIMMONS, R. (1973). 'Semantic networks: their computational use for understanding English', in SCHANK, R. C. and COLBY, K. M. (Eds) *Computer Models of Thought and Language*, pp. 63–113, Freeman, San Francisco

SPEIDEL, R., SUMMERS, R. and WHITE, J. (1974). *Teaching Materials on Commerical and Consumer Law*, 2nd edn, West Publishing, St. Paul, Minn.

West's California Digest (1966). West Publishing, St. Paul, Minn.

WESTLAW: The Computerized Legal Research System (n.d.). West Publishing, St. Paul, Minn.

WHITE, J. and SUMMERS, R. (1972). *Handbook of the Law under the Uniform Commercial Code*, West Publishing, St. Paul, Minn.