VOLUME 90 MARCH 1977 NUMBER 5

HARVARD LAW REVIEW

REFLECTIONS ON *TAXMAN*: AN EXPERIMENT IN ARTIFICIAL INTELLIGENCE AND LEGAL REASONING †

L. Thorne McCarty *

After introducing some basic techniques of semantic information processing developed in the field of artificial intelligence, Professor McCarty describes a computer program using such techniques which enables a computer to apply to certain fact situations concepts of the area of the taxation of corporate reorganizations. He considers both the present limitations of the program and ways in which it might feasibly be improved, and argues that if developed further by means of increasingly sophisticated techniques, it may prove of practical use to tax lawyers. The development of such computer models also has a dual theoretical purpose. It clarifies the structure of the area of law modeled, and the investigation of the limits of such models provides the foundation for a theory of the nature of legal concepts.

THIS Article introduces a new approach to some of the classical problems of legal reasoning: What is the structure of a legal concept? What is the process by which legal concepts are transformed and modified? What are the arguments that lawyers characteristically make with respect to legal concepts? These are the kinds of questions that the Analytical Jurists once treated as essential, and the Realists dismissed as irrelevant; the kinds of questions that Cardozo ascribed to the Method of Philosophy in the judicial process, as opposed to the Method of Sociology.[1]

* Assistant Professor of Law, State University of New York at Buffalo. B.A. 1966, Yale University; J.D. 1969, Harvard University. The work on this project was begun while the author was a Law and Computer Fellow at the Stanford Law School, 1971–1973, and was supported at that time by a grant from the IBM Corporation; additional work was undertaken during a summer workshop at the Stanford Law School in 1975, which was also supported by a grant from the IBM Corporation. Computer time for the project was provided by the Stanford Artificial Intelligence Laboratory. The Article in its present form owes a great deal to the suggestions and comments of a number of individuals, including especially Richard S. Bell, Bruce G. Buchanan, Marc Galanter, David G. Hays, Thomas E. Headrick, Mary Kay Kane, Jonathan King, Ejan Mackaay, Philip Slayton, Colin Tapper, and George Williams. None of these individuals is responsible, of course, for any errors of fact or interpretation that remain. Special thanks are due, above all, to Associate Dean Joseph E. Leininger of the Stanford Law School, whose support and encouragement over the past few years made this Article possible.

[1] B. CARDOZO, THE NATURE OF THE JUDICIAL PROCESS (1921).

But my approach to these problems will be anything but classical. I will describe a computer program I have written which models certain aspects of the conceptual structures which occur in a specific area of the law, the taxation of corporate reorganizations, as set forth in subchapter C of chapter 1 of the Internal Revenue Code of 1954.[2] The program, dubbed mnemonically TAXMAN, is capable of performing a very rudimentary form of "legal reasoning": Given a "description" of the "facts" of a corporate reorganization case, it can develop an "analysis" of these facts in terms of several legal "concepts." I will describe in considerable detail how this program operates, what its limitations are, and how it might be revised and rewritten to achieve a somewhat higher level of sophistication.[3] By looking carefully at both the adequacies and inadequacies of a particular formal model of a particular area of the law, I hope eventually to provide some insights into the structure and dynamics of legal concepts generally.[4]

The present work also has a practical purpose. Despite its limitations, the TAXMAN system provides a number of tech-

[2] I.R.C. §§ 354–356, 358, 361–362, 368.

[3] The TAXMAN program was written in 1972–1973 and was first discussed in a paper presented at a workshop on Computer Applications to Legal Research and Analysis, Stanford Law School, April 28–29, 1972. McCarty, Interim Reports on the TAXMAN Project: An Experiment in Artificial Intelligence and Legal Reasoning, in *Artificial Intelligence Techniques in Legal Problem Solving* (June 1, 1973) (mimeo, Stanford Law School). The most extensive analysis of this unpublished paper appears in P. SLAYTON, ELECTRONIC LEGAL RETRIEVAL 18–19, 25 (Canadian Department of Communications, 1974); P. Slayton, Radical Computer Use in Law 67–68 (June 1974) (Draft Report prepared for the Canadian Department of Communications). Additional references to TAXMAN appear in Boyd, *Law in Computers and Computers in Law: A Lawyer's View of the State of the Art*, 14 ARIZ. L. REV. 267, 286–87 n.107 (1972); Bauer-Bernet, *Effect of Information Science on the Formation and Drafting of Law*, 14 JURIMETRICS J. 235, 248 n.14 (1974); Popp & Schlink, *JUDITH, A Computer Program to Advise Lawyers in Reasoning a Case*, 15 JURIMETRICS J. 303, 313–14 (1975). The present Article, however, contains the first full-length description of the program.

[4] The most important precursor to the present article is Buchanan & Headrick, *Some Speculation About Artificial Intelligence and Legal Reasoning*, 23 STAN. L. REV. 40 (1970), which urged that serious research be undertaken to apply computer science methods to legal problem-solving, and speculated that this research "could lead both to a greater understanding of the legal reasoning process and to the design of machine methods for performing parts of it," *id.* at 41. A more recent project along these lines is Meldman, A Preliminary Study in Computer-Aided Legal Analysis, Project MAC Report No. TR-157 (1975) (Ph.D. Dissertation, Mass. Inst. of Technology), which presents a detailed design of a program to assist in the analysis of a simple assault and battery case, but reports no actual implementation. *See also* Stamper, *The LEGOL Project and Language*, in PROCEEDINGS, DATAFAIR SYMPOSIUM, 1973, at 263 (British Computer Soc'y, London); Note, *A Computer Method for Legal Drafting Using Propositional Logic*, 53 TEX. L. REV. 965 (1975); Popp & Schlink, *supra* note 3.

niques that should be useful in automating the more mundane aspects of legal research and analysis. The existing systems for computer-aided research and analysis in the law depend either on an exhaustive search through the full text of a body of legal materials and a retrieval of documents by key-words or combinations of key-words,[5] or on a sequence of preprogrammed questions and answers about a legal problem that is designed to terminate in a legal conclusion, in much the same style as a program for computer-aided instruction.[6] The TAXMAN system, with its computer representations of actual legal concepts, suggests a more structured, more flexible, potentially more powerful alternative, although it remains to be seen how practical and efficient this alternative will be.[7] At the very least, the prospects seem promising enough to justify a substantial amount of additional research.

But whatever its practical applications, the TAXMAN system provides, I claim, an important tool for the development of our theories about legal reasoning. A great many of the classical jurisprudential problems are tied to problems about the uses of abstract concepts in the regulation of human affairs. Thus Dworkin to a large extent bases his analysis of judicial discretion on his "logical distinction" between "legal rules" and "legal principles";[8] Hart and Fuller devote a portion of their debate on legal positivism to a dispute about the "core," the "penumbra," and the "open texture" of legal concepts;[9] Levi describes the law as the prime example of a "moving classification system," and sees in this description an explanation for many of the perplexing characteristics of the legal process.[10] In these and other writings there are many illuminating examples and many valuable insights about the structure and dynamics of legal concepts. But taken as a whole, the jurisprudential literature is notoriously imprecise: the conceptual structures themselves are only vaguely defined and vaguely distinguished from one another; the dynamics of conceptual change appear only as suggestive metaphors.

The TAXMAN system adds a strong dose of precision and rigor to these discussions of linguistic and conceptual problems. Its critical task is to clarify the concepts of corporate reorganization law in such a way that they can be represented in

[5] *See* pp. 888–90 & notes 123–26 *infra.*

[6] *See* pp. 890–91 *infra.*

[7] *See* p. 888 *infra.*

[8] Dworkin, *The Model of Rules*, 35 U. CHI. L. REV. 14, 22–40 (1967).

[9] Hart, *Positivism and the Separation of Law and Morals*, 71 HARV. L. REV. 593, 606–15 (1958); Fuller, *Positivism and Fidelity to Law — A Reply to Professor Hart*, 71 HARV. L. REV. 630, 661–69 (1958).

[10] E. LEVI, AN INTRODUCTION TO LEGAL REASONING 1–7 (1949).

computer programs. This requires a degree of explicitness about the structure of these concepts that has never previously been attempted. When we describe concepts in this way, we implicitly articulate theories about them; when we run the computer programs that embody these concepts, we test out the implications of our theories. Used in this fashion, the computer is the most powerful tool for expressing formal theories and spinning out their consequences that has ever been devised.

We can see this point most clearly, I think, if we examine briefly the contemporary traditions in linguistics and cognitive psychology, two areas with obvious connections to the present work.[11] Both disciplines, in opposition to their behaviorist predecessors, posit abstract mental structures in order to explain basic linguistic and psychological facts. In linguistics, the proposed mental structures are syntactic and semantic in nature, and they purport to organize and explain our intuitions about the grammaticality of various sentences. In cognitive psychology, where the schools of thought are more varied and diffuse, the proposed structures may be perceptual schemata, or memory mechanisms, or devices for acquiring and abstracting concepts. But what is especially striking about the abstract structural models that appear in these disciplines today is the fact that they are so closely tied to computational structures, that is, to the kinds of structures that are most naturally represented by computer programs. There are many examples in the recent literature: semantic theories formulated and tested within a computer system that conducts a simple dialogue in English; [12] theories of associative memory written in the form of programs and validated by experimental data from human subjects.[13] In these areas at least, the computer program has proven itself to be one of the ideal modes of expression for linguistic and psychological ideas.

So too with TAXMAN. But one must proceed with sophis-

[11] I assume that most lawyers are generally aware of the revolutions wrought by Chomsky and others in linguistics and by Piaget and others in cognitive psychology, since their ideas have by now worked their way into the popular literature. For accessible introductions, see, *e.g.*, N. CHOMSKY, LANGUAGE AND MIND (1968); J. PIAGET, THE ORIGINS OF INTELLIGENCE IN CHILDREN (1952); U. NEISSER, COGNITIVE PSYCHOLOGY (1968).

[12] *See, e.g.*, T. WINOGRAD, UNDERSTANDING NATURAL LANGUAGE (1972); Schank, *Conceptual Dependency: A Theory of Natural Language Understanding*, 3 COGNITIVE PSYCH. 552 (1972); Woods, *Transition Network Grammars for Natural Language Analysis*, 13 COMM. OF THE ASS'N FOR COMPUTING MACH. 591 (1970).

[13] *See, e.g.*, D.A. NORMAN, D. RUMELHART, & THE LNR RESEARCH GROUP, EXPLORATIONS IN COGNITION (1975); J.R. ANDERSON & G. BOWER, HUMAN ASSOCIATIVE MEMORY (1973); Quillian, *Semantic Memory*, in SEMANTIC INFORMATION PROCESSING 227 (M. Minsky ed. 1968).

tication about the uses and abuses of formal models. A formal model is necessarily a simplification of reality. It omits details, by design, which in many contexts might be crucial; and so, by design, it will always be inadequate in some respects. And yet the simplification inherent in a formal model is also the source of its power and utility: it will often lead us to insights that would otherwise be obscured, overwhelmed by the complexity of our data. The unexpected consequences of our formulations may reveal surprising truths or, just as often, the inadequacy of the formulations themselves.[14] These unacceptable conclusions must not be ignored but must be exploited systematically for the insights they can yield.

This is the strategy I have attempted to follow. This Article thus explores the positive role of the TAXMAN system, pushing it to its limits. The formal model here comes from a branch of computer science devoted to the study of what is called *artificial intelligence*.[15] The model is designed to represent linguistic and conceptual information in ways that are arguably similar to the

---

[14] Chomsky has stated this point succinctly and accurately in the preface to his earlier work:

> Precisely constructed models for linguistic structure can play an important role, both negative and positive, in the process of discovery itself. By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the linguistic data. More positively, a formalized theory may automatically provide solutions for many problems other than those for which it was explicitly designed. Obscure and intuition-bound notions can neither lead to absurd conclusions nor provide new and correct ones, and hence they fail to be useful in two important respects.

N. CHOMSKY, SYNTACTIC STRUCTURES 5 (1957).

[15] The field of artificial intelligence (often abbreviated "AI") is concerned with the problem of designing machines which exhibit some form of "intelligent" behavior. This includes machines which play chess, *see, e.g.*, Greenblatt, Eastlake, & Crocker, *The Greenblatt Chess Program*, in PROCEEDINGS OF THE FALL JOINT COMPUTER CONF., 1967, at 801 (Am. Fed'n of Information Processing Societies, Montvale, N.J.); prove mathematical theorems, *see, e.g.*, Bledsoe, Boyer, & Henneman, *Computer Proofs of Limit Theorems*, 3 ARTIFICIAL INTELLIGENCE 27 (1972); solve mass spectrometry problems in organic chemistry, *see, e.g.*, Buchanan, Sutherland, & Feigenbaum, *Rediscovering Some Problems of Artificial Intelligence in the Context of Organic Chemistry*, in 5 MACH. INTELLIGENCE 253 (B. Meltzer & D. Mitchie eds. 1970); and many others. The name "artificial intelligence" was chosen partly because of the provocative stance it takes on the question "Can a Machine Think?" that was raised in the early article by Turing, *Computing Machinery and Intelligence*, 59 MIND 433 (1950). On the unfortunate connotations of this choice of words, see H. SIMON, THE SCIENCES OF THE ARTIFICIAL 4 n.1 (1969). *See generally* COMPUTERS AND THOUGHT (E. Feigenbaum & J. Feldman eds. 1963); SEMANTIC INFORMATION PROCESSING (M. Minsky ed. 1968); REPRESENTATION AND MEANING (H. Simon & L. Siklossy eds. 1972). For a sharp critique, see J. WEIZENBAUM, COMPUTER POWER AND HUMAN REASON: FROM JUDGMENT TO CALCULATION (1976); H. DREYFUS, WHAT COMPUTERS CAN'T DO: A CRITICISM OF ARTIFICIAL REASON (1972).

representations of this kind of information in the human mind. The techniques used, sometimes referred to as techniques of *semantic information processing*,[16] are typical of the current applications of computer science to linguistics and cognitive psychology. The questions then are: How adequate is this paradigm? How adequately does it explain the familiar facts of legal reasoning? Specifically, how well does it handle the taxation of corporate reorganizations? It will become clear, I think, that the TAXMAN paradigm has considerable power but that it also has several serious deficiencies.[17]

The two Parts following this introduction are intended to provide the necessary background for anyone who knows nothing about either the legal or the technical foundations of the work. Part I thus presents an elementary introduction to the taxation of corporate reorganizations, and Part II presents an elementary introduction to the basic ideas of sematic information processing. With this background, Part III presents a description of the current implementation of TAXMAN, Part IV presents the actual performance of TAXMAN in analyzing a corporate reorganization case, and Part V outlines several extensions of the program which seem feasible within the present state of the art. A thorough exploration of these feasible extensions will help make clear the capabilities and limitations of the present approach.

I. Some Fundamentals of Corporate Reorganization Law

For an area of the law to be a suitable subject for a project like TAXMAN, it should possess both simplicity and complexity. The basic facts and concepts in the area should be simple enough for the application of the existing techniques of semantic information processing, allowing us to develop a working computer model

[16] *See, e.g.*, Semantic Information Processing (M. Minsky ed. 1968). I use the term here to indicate a subfield within the broader field of artificial intelligence, and to indicate also a general evolutionary trend in the field of artificial intelligence as a whole. As a subfield, semantic information processing is more closley tied to the work in linguistics and cognitive psychology, *supra* notes 12 and 13, than it is to the work on game-playing, theorem-proving, and problem-solving programs, *supra* note 15. But even within the game-playing programs themselves, the term indicates a shift in emphasis: the earlier programs were characterized primarily by their "heuristic search" techniques, while the later programs were increasingly characterized by their techniques for the "representation of knowledge." To see this shift in emphasis, compare the early articles collected in Computers and Thought (E. Feigenbaum & J. Feldman eds. 1968) with the articles collected in Representation and Meaning: Experiments with Information Processing Systems (H. Simon & L. Siklossy eds. 1972).

[17] In a future article I will discuss these deficiencies further and suggest some modifications. *See* pp. 892–93 *infra*.

in a reasonable amount of time. But at least some of these concepts should exhibit a richness and depth that cannot be captured in a trivial way, allowing us to push the current paradigm to its limits.

The taxation of corporate reorganizations is suitable in both respects. To convey briefly the basic ideas, I will use a simple hypothetical case.[18] Consider a New Jersey corporation which wishes to transfer its place of incorporation to Delaware. It chooses to do so by means of an "assets acquisition." It first sets up a new corporation in Delaware, then transfers all its assets to the Delaware corporation in exchange for all of the newly issued Delaware stock. The New Jersey stockholders then control a company which owns nothing but Delaware stock, and the New Jersey company operates the Delaware company as a wholly owned subsidiary. At this point, though, the old New Jersey company would probably be liquidated and the Delaware stock distributed pro rata to the New Jersey stockholders in exchange for their New Jersey stock.[19] This result would be fully equivalent to a direct reincorporation of the New Jersey company in Delaware.

If it were not for special provisions in the Internal Revenue Code, these transactions would produce a potential tax liability for both the stockholders and the corporation. The New Jersey stockholders would be taxed on the receipt of the Delaware stock, as on the receipt of any corporate distribution. If the Delaware stock were completely distributed and the New Jersey corporation completely liquidated, the stockholders would be taxed at capital gains rates on the appreciation of their investment, that is, on the difference between the market value of the Delaware shares received and the cost basis of their original shares in New Jersey.[20] In addition, the initial transfer of property to Delaware

[18] The hypothetical case that follows is based on United States v. Phellis, 257 U.S. 156 (1921), the earliest reorganization case. *Phellis* also appears in several subsequent sections of this Article. In using *Phellis* as a device for an introduction to corporate reorganization law, I have drawn heavily upon W. ANDREWS, FEDERAL INCOME TAXATION 126–58, 168–69, 757–844 (1969) [hereinafter cited as ANDREWS]; I have also drawn upon B. BITTKER & J. EUSTICE, FEDERAL INCOME TAXATION OF CORPORATIONS AND SHAREHOLDERS (3d ed. 1971) [hereinafter cited as BITTKER].

[19] Alternatively, and less likely, only a portion of the Delaware stock might be distributed in exchange for a portion of the New Jersey stock, while the New Jersey corporation itself remained in existence. The original stockholders would then control the Delaware company partly by direct ownership of its stock, and partly by indirect ownership through New Jersey.

[20] I.R.C. §§ 331(a)(1), 1001–1002, 1221–1222. The partial distribution in note 19 *supra* could result in even more burdensome tax consequences for the stockholders: unless qualifying as an "exchange" under § 302(b) or a "partial liquida-

would be viewed as a sale, and the New Jersey corporation would be taxed, at capital gains rates, on the value of the Delaware shares received less the cost basis of its transferred assets.

It could be argued, however, that these transactions have no economic significance beyond the change in the state of incorporation, and so should produce no tax consequences. In substance, the argument goes, there has been no sale or liquidation or distribution, but only the formal reshuffling of stock certificates: the stockholders retain the same interest in the same operating assets after the transactions as before.[21] The Internal Revenue Code adopts essentially this position in its provisions for tax-free corporate "reorganizations." Under section 354 of the Code, no gain or loss is recognized if "stock or securities in a corporation a party to a reorganization are, in pursuance of the plan of reorganization, exchanged solely for stock or securities" in a "corporation a party to the reorganization." Likewise, under section 361(a), no gain or loss is recognized if "a corporation a party to a reorganization exchanges property, in pursuance of the plan of reorganization, solely for stock or securities in another corporation a party to the reorganization." When a gain or loss goes unrecognized in a reorganization exchange, however, the basis of the property transferred is generally substituted for the basis of the property received,[22] thus making the unrecognized appreciation potentially taxable at a later date. The net effect of the recognition and basis provisions taken together, then, is not to eliminate the tax entirely, but to defer it until the reorganization related property is eventually sold in a nonreorganization exchange.

Since the recognition and basis provisions apply in full only to "reorganizations," that term is defined in elaborate detail in section 368(a). There are six types of reorganizations, known colloquially by the letter of the subsection in which they are defined. Type A, Type B, and Type C reorganizations are intended primarily for the combination of two separate corporations into

---

tion" under § 346, it would be treated as a "dividend" and included in ordinary gross income. I.R.C. §§ 301(a), (c), 302, 316–317, 331(a)(2), 346.

[21] This argument was originally made, and rejected, under a statute that contained no special reorganization provisions. *See* United States v. Phellis, 257 U.S. 156 (1921); Rockefeller v. United States, 257 U.S. 176 (1921); Cullinan v. Walker, 262 U.S. 134 (1923); Marr v. United States, 268 U.S. 536 (1925). *But see* Weiss v. Stearn, 265 U.S. 242 (1924).

[22] Under § 358, for example, the shareholders who receive stock or securities in the reorganization will carry their new certificates at the same basis as the old certificates they have given up, and the corporation which receives stock or securities in exchange for its property will carry the new certificates at the same basis as the transferred assets.

one, Type D reorganizations primarily for the division of one corporation into two, and Type E and Type F reorganizations for simple adjustments in capital structure, place of incorporation, etc., within a single ongoing corporation. Omitting some important detail, the six categories are: (A) "a statutory merger or consolidation"; [23] (B) "the acquisition by one corporation, in exchange solely for . . . its voting stock . . . , of stock of another corporation," if immediately thereafter the acquiring corporation has "control" of the acquired corporation; [24] (C) the acquisition by one corporation of "substantially all of the properties" of another corporation, again in exchange "solely for . . . voting stock" of the acquiring corporation; [25] (D) the transfer of assets by one corporation to another corporation of which the transferor immediately thereafter is in "control"; [26] (E) "a recapitalization"; [27] and (F) "a mere change in identity, form, or place of organization, however effected." [28] The concept of "control" is defined fairly mechanically as "the ownership of stock possessing at least 80 percent of the total combined voting power of all classes of stock entitled to vote and at least 80 percent of the total number of shares of all other classes of stock of the corporation." [29] The statute also delineates in detail the concept of "a party to a reorganization," [30] while the concept of a "plan of reorganization" is left conspicuously undefined.

Our New Jersey-Delaware hypothetical [31] clearly involves a Type F reorganization, but it fits the general pattern of two

---

[23] I.R.C. § 368(a)(1)(A). *See also* § 368(a)(2)(D) and (E), which provides additional rules for statutory mergers with a subsidiary corporation using stock of the subsidiary's parent, and which thus creates two hybrid reorganization patterns for "triangular mergers."

[24] I.R.C. § 368(a)(1)(B).

[25] I.R.C. § 368(a)(1)(C). *See also* § 368(a)(2)(B), which provides additional rules for the case in which the acquiring corporation exchanges "money or other property" in addition to voting stock.

[26] I.R.C. § 368(a)(1)(D). In order to qualify under § 368(a)(1)(D), however, the plan of reorganization must include a distribution of the transferee's stock or securities which qualifies under § 354, § 355, or § 356. But according to § 354(b)(1), § 354 does not apply to a Type D reorganization unless the transferee acquires "substantially all of the assets" of the transferor, and the transferor in turn distributes all of its properties (including those received in the reorganization) in what would effectively be a complete liquidation. Thus any putative Type D reorganization involving *less* than substantially all the assets of the transferor, or *less* than a complete liquidation of the transferor, would have to satisfy the requirements of § 355.

[27] I.R.C. § 368(a)(1)(E).

[28] I.R.C. § 368(a)(1)(F).

[29] I.R.C. § 368(c).

[30] I.R.C. § 368(b).

[31] *See* p. 843 *supra*.

other provisions as well. It could be viewed as an acquisition by the Delaware corporation of "substantially all of the properties" of the New Jersey corporation, a Type C reorganization. Or it could be viewed as a transfer of assets by the New Jersey corporation to the Delaware corporation, which New Jersey then "controls" by virtue of its ownership of all the newly issued Delaware stock, a Type D reorganization. In either case, if the overall transaction qualifies as a "reorganization," the receipt of Delaware stock by the New Jersey stockholders would be nontaxable under section 354, and the transfer of assets by the New Jersey corporation would be nontaxable under section 361.

Despite the apparent precision of these rules, the courts have intervened repeatedly throughout the history of the Code to deny a claimed reorganization status to transactions which have seemed inconsistent with their underlying purposes and assumptions. One early example is the doctrine of "continuity of interest" developed by the courts in a series of cases in the 1930's.[32] Prior to 1934, the predecessors of the Type B and Type C clauses lacked the requirement that a reorganization acquisition be made in exchange "solely for . . . voting stock" of the acquiring corporation and so applied, literally, to transactions which were outright sales. Suppose the New Jersey stockholders in our original hypothetical decide to sell out their company entirely to the DuPont corporation. They have New Jersey transfer all its assets to DuPont in exchange for DuPont's short-term notes, payable in two months; they then arrange the liquidation of New Jersey, receiving the notes as a liquidating dividend. Under the 1928 reorganization provisions these transactions would have qualified as a tax-free Type C reorganization,[33] with no tax to the New Jersey stockholders on receipt of the short-term securities, and no tax to the New Jersey corporation on the initial sale of its assets.

When a problem of this sort came to the Supreme Court in *Pinellas Ice & Cold Storage Co. v. Commissioner*,[34] however, the Court denied the claimed reorganization status. It held that a transaction in which the original owners retained no continuing interest in their transferred assets was, indeed, equivalent to a sale, and "to be within the exemption the seller must acquire an

[32] Cortland Specialty Co. v. Commissioner, 60 F.2d 937 (2d Cir. 1932), *cert. denied*, 288 U.S. 599 (1933); Pinellas Ice & Cold Storage Co. v. Commissioner, 287 U.S. 462 (1933); John A. Nelson Co. v. Helvering, 296 U.S. 374 (1935); Helvering v. Minnesota Tea Co., 296 U.S. 378 (1935); Helvering v. Watts, 296 U.S. 387 (1935); LeTulle v. Scofield, 308 U.S. 415 (1940).

[33] Revenue Act of 1928, ch. 852, § 112(i)(1)(A), 45 Stat. 791.

[34] 287 U.S. 462 (1933).

interest in the affairs of the purchasing company more definite than that incident to ownership of its short-term purchase-money notes."[35] Alternatively, the Court held that the transferred assets were not exchanged "solely for stock or securities in another corporation a party to the reorganization," under what is now section 361(a), since the notes "were not securities within the intendment of the act and were properly regarded as the equivalent of cash."[36] Today the transferor's continuity of interest in a Type B or Type C reorganization is assured by the "solely for . . . voting stock" requirement, but the term "stock or securities" is still defined only by judicial decision, and the "continuity of interest" doctrine remains important in other reorganization contexts.[37]

The celebrated case of *Gregory v. Helvering*[38] represents a more exotic example of a judicial refusal to apply the statutory language literally. Suppose our New Jersey corporation owns a block of General Motors stock which has appreciated substantially in value, and the New Jersey stockholders wish to sell the stock and realize a substantial profit. Since a direct distribution of the stock would be taxable at ordinary income rates, New Jersey embarks on the following plan of "reorganization." A new corporation is established in Delaware, and the General Motors stock is transferred to it in exchange for a controlling amount of newly issued Delaware stock, which is then distributed to the New Jersey stockholders. Under the 1928 statute, this transaction would have qualified as a Type D reorganization,[39] and neither the transfer of the General Motors stock nor the distribution of the Delaware stock would have been taxable. The Delaware corporation would then have been liquidated and the General Motors stock distributed to its stockholders. This liquidating distribution would normally have been taxable as a capital gain.[40] But while admitting that "the facts answer the dictionary definitions of each term used in the statutory definition,"[41] and that a taxpayer "may so arrange his affairs that his

[35] *Id.* at 470.

[36] *Id.* at 468–69.

[37] For example, it remains important for Type A and Type E reorganizations, where there is no statutory codification of a "continuity of interest" rule. *See* ANDREWS, *supra* note 18, at 810–15; BITTKER, *supra* note 18, ¶¶ 14.11, .12, .17.

[38] 293 U.S. 465 (1935), *aff'g* 69 F.2d 809 (2d Cir. 1934).

[39] Revenue Act of 1928, ch. 852, § 112(i)(1)(B), 45 Stat. 791. Under the current statute, though, the transaction would not satisfy the final clause of § 368(a)(1), *see* note 26 *supra*, which was intended to prevent a tax-free result in a case like *Gregory*.

[40] *See* I.R.C. § 331(a)(1).

[41] 69 F.2d 809, 810 (2d Cir. 1934).

taxes shall be as low as possible,"[42] the Second Circuit and the Supreme Court denied the capital gains treatment in *Gregory*. According to the Supreme Court, "the whole undertaking, though conducted according to the terms of subdivision (B), was in fact an elaborate and devious form of conveyance masquerading as a corporate reorganization, and nothing else."[43]

The complex manipulations of *Helvering v. Elkhorn Coal Co.*[44] provide a similar example. Paraphrasing the facts again in terms of our hypothetical, we assume that New Jersey wishes to transfer some, but not all, of its assets to DuPont in a tax-free exchange. First, New Jersey transfers the assets it wishes to retain to a newly created Delaware corporation in a Type D reorganization, distributing the Delaware stock; second, New Jersey transfers all its remaining assets to DuPont in exchange for voting stock, claiming a Type C reorganization; finally, Delaware acquires all the outstanding New Jersey stock from the New Jersey stockholders in exchange for its own voting stock, a Type B reorganization. When Delaware subsequently liquidates New Jersey, the transaction is complete: the Delaware corporation has now stepped completely into the former position of the New Jersey corporation, retaining only the desired assets plus the stock from DuPont. Once again, under the 1928 statute, the definitions of a Type D, a Type C, and a Type B reorganization would have been satisfied.[45] But the Fourth Circuit in *Elkhorn* refused to find a tax-free reorganization. The transfer of assets to the newly created corporation, it said, served only to

---

[42] *Id.*

[43] The Court continued:

> The rule which excludes from consideration the motive of tax avoidance is not pertinent to the situation, because the transaction upon its face lies outside the plain intent of the statute. To hold otherwise would be to exalt artifice above reality and to deprive the statutory provision in question of all serious purpose.

293 U.S. 465, 470 (1935). The principle developed in *Gregory* is known as the "business purpose" doctrine. It is restated in similar language in the Regulations under § 368: "A scheme, which involves an abrupt departure from normal reorganization procedure in connection with a transaction on which the imposition of tax is imminent, such as a mere device that puts on the form of a corporate reorganization as a disguise for concealing its real character, and the object and accomplishment of which is the consummation of a preconceived plan having no business or corporate purpose, is not a plan of reorganization," Treas. Reg. § 1.368–1(c) (1955).

[44] 95 F.2d 732 (4th Cir.), *cert. denied*, 305 U.S. 605 (1938).

[45] Revenue Act of 1928, ch. 852, § 112(i)(A),(B), 45 Stat. 791. In this case, unlike the earlier ones, the current statutory definitions of a Type D, a Type C, and a Type B reorganization might be satisfied as well. In particular, the requirements of § 368 (a)(1)(D) and § 355 could be satisfied if both the assets transferred and the assets retained constituted an "active trade or business." *See* BITTKER, *supra* note 18, ¶ 14.52.

strip these assets away temporarily in preparation for the contemplated Type C reorganization. Since this first step was "a mere shifting of charters,"[46] the subsequent step had only the appearance of being a transfer of "substantially all of the properties" of the transferor. In reality, only a portion of the assets of the old company was transferred, and the transaction would therefore be taxable as a sale. This is "not made a different result because reached by following a devious path."[47]

It should now be clear why this area of the law is ideally suited for a project like TAXMAN. Superimposed on a manageable foundation of manageable complexity is another system of concepts as unruly as any that can be found in the law, with all the classical dilemmas of legal reasoning: contrasts between "form" and "substance," between statutory "rules" and judicially created "principles,"[48] between "legal formality" and "substantive rationality."[49] It is perhaps no accident that two of the more quotable passages on the problems of legal interpretation have arisen in corporate tax cases. Holmes' famous metaphor — "[a] word is not a crystal, transparent and unchanged, it is the skin of a living thought" — appeared first in *Towne v. Eisner*,[50] the earliest stock dividend case. And the simile of Learned Hand — "the meaning of a sentence may be more than that of the separate words, as a melody is more than the notes" — appeared first in *Helvering v. Gregory*.[51] In a sense, the ultimate task for the TAXMAN project is to translate these metaphors and similes into something more precise and concrete.

But that is indeed an ultimate task. The current version of TAXMAN is capable only of classifying a given case under the statutory rules as a Type B, a Type C, or a Type D reorganization.[52] An extended version of TAXMAN would be capable of covering a much broader expanse of corporate tax law: the full treatment of the parties to a reorganization, including both the nonrecognition and basis provisions; the treatment of corporate distributions outside the reorganization context; etc.[53] But the

---

[46] 95 F.2d 732, 734 (4th Cir.), *cert. denied*, 305 U.S. 605 (1938).

[47] *Id.* at 738 (on rehearing) (quoting Minnesota Tea Co. v. Helvering, 302 U.S. 609, 613 (1938) (emphasis omitted)). The principle developed here is generally referred to as the "step transaction" doctrine.

[48] *See, e.g.*, Dworkin, *The Model of Rules*, 35 U. CHI. L. REV. 14, 22–29 (1967).

[49] *See, e.g.*, Kennedy, *Form and Substance in Private Law Adjudication*, 89 HARV. L. REV. 1685 (1976); Kennedy, *Legal Formality*, 2 J. LEGAL STUD. 351, 354–60, 391–98 (1973).

[50] 245 U.S. 418, 425 (1918).

[51] 69 F.2d 809, 810–11 (2d Cir. 1934), *aff'd*, 293 U.S. 465 (1935).

[52] *See* pp. 876–81 *infra*.

[53] *See* pp. 882–84 & notes 103–111 *infra*.

concepts of "continuity of interest," "business purpose," and "step transaction," developed initially in *Pinellas*, *Gregory*, and *Elkhorn Coal*, may very well be beyond the reach of these straightforward extensions of the program. If so, these cases and their successors will provide the material we need to push the current paradigm to its limits and to suggest modifications.

II. Some Fundamentals of Semantic Information Processing

To understand what the TAXMAN system can and cannot accomplish, and why, we need to understand some of the existing techniques of semantic information processing. The problem is to develop models of conceptual structures which are at least rough approximations of the conceptual structures developed by the human mind. In this Part, I will show how these conceptual structures can be represented and stored in computer memory, how they can be searched for and retrieved from computer memory, and how they can be manipulated to simulate some rudimentary patterns of human reasoning and inference.

In the past several years, much of the work on this problem has coalesced into the development of a family of programming languages designed especially for these purposes, and thus one of the best ways to analyze the current ideas in this branch of computer science is to study one of these programming languages in detail. In the exposition that follows, I will use a simplified version of the programming language in which TAXMAN is actually written,[54] which is itself one of the earliest members of the family. Despite this simplification,[55] the version of the language I will use

[54] TAXMAN is written primarily in a language called Micro-PLANNER, which is a subset of a (not fully implemented) language called PLANNER. *See* T. Winograd, *supra* note 12, at 23–25, 108–17; G. Sussman, T. Winograd, & E. Charniak, Micro-PLANNER Reference Manual, AI Memo No. 203 (Mass. Inst. of Technology, 1970); C. Hewitt, Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot, AI Memo No. 251 (Ph.D. Dissertation, Mass. Inst. of Technology, 1972). Micro-PLANNER itself is a high-level language written in a lower-level language called LISP, which has been one of the principal tools for artificial intelligence research for over a decade. *See* J. McCarthy, P. Abrahams, D. Edwards, T. Hart, & M. Levin, LISP 1.5 Programmer's Manual (1962); C. Weissman, LISP 1.5 Primer (1967). Portions of TAXMAN are also written directly in LISP.

[55] The expert reader will note that many of the technical developments since Micro-PLANNER are omitted from this account or relegated to qualifying footnotes. This seems to me justified on pedagogical grounds. For a comparative survey of PLANNER and related languages (CONNIVER, QA4, QLISP, etc.), see Bobrow & Raphael, *New Programming Languages for Artificial Intelligence Research*, 6 Computing Surveys 153 (1974).

exhibits most of the salient features, and most of the outstanding problems, of the current paradigm of semantic information processing.

The concepts we are concerned about, of course, are those of corporate reorganization law. We must express certain fundamental propositions about corporations: that a particular corporation exists; that it has issued certain classes of stock; that certain individuals or other corporations own various shares of that stock; that the ownership of these shares of stock has been transferred from time to time. Propositions of this sort function as the relatively concrete, relatively primitive facts upon which reorganization law is constructed. We must also express a range of concepts which function within reorganization law at a much higher level of abstraction: "stock distribution," for example, or the concept of "control," or the legal concept of a "tax-free reorganization" itself. In fact, a major portion of the task here is the representation of these various levels of conceptual abstraction and the connection of one such level to another.

I will begin with the simplest representational problems. Given a corporation named "New Jersey," we can express the proposition "New Jersey is a corporation" by storing in computer memory the simple *list* of two *words*: (CORPORATION NEW-JERSEY). This list is stored exactly as written, both words spelled out in full, and it is indexed in such a way that by specifying either of the words CORPORATION or NEW-JERSEY we can find the location of the list in memory and retrieve it. The conventions for interpreting such a list are similar to those of formal symbolic logic: the first word is a *predicate*; the remaining words designate *objects* to which the predicate is applied. If we also want to express the proposition that "Delaware is a corporation," we could do so by storing a second list (CORPORATION DELAWARE). Here, CORPORATION would be the same one-place predicate as before, with the same interpretation, but applied now to an object named DELAWARE. The indexing system could now be used to retrieve either (CORPORATION NEW-JERSEY), by specifying NEW-JERSEY, or (CORPORATION DELAWARE), by specifying DELAWARE, or both (CORPORATION NEW-JERSEY) and (CORPORATION DELAWARE) together, by specifying the word CORPORATION.[56]

---

[56] A list can be composed not only by stringing together a sequence of words (or atoms) like CORPORATION and NEW-JERSEY and enclosing them inside a pair of parentheses, but also by stringing together and parenthesizing a sequence of *sublists*, like (LIST (CORPORATION NEW-JERSEY) (CORPORATION DELAWARE)), and this nesting of lists and sublists can continue to an arbitrary depth.

For propositions of slightly greater complexity, we can select a representational convention on largely pragmatic grounds. Suppose we want to represent the simple fact that New Jersey, a corporation, has issued stock. Since it may be necessary to say something further about the particular stock that New Jersey has issued, that it was common stock, that it was worth so many dollars per share, etc., it will be helpful to have a symbol in computer memory to which we can attach these additional propositions. We should therefore construct a new object, S1, say, and introduce a new predicate, STOCK, with which to assert that the new object is a class of stock. We thus store two lists to represent our original proposition: (ISSUE NEW-JERSEY S1) and (STOCK S1). Later, if we wish to say that the stock issued by New Jersey is worth so many dollars per share, the previously stored information will tell us that S1 is a stock and that S1 is issued by New Jersey, so we need only store in computer memory the additional fact that S1 is worth so many dollars per share. This procedure illustrates one of the basic conventions of TAXMAN: that we *nominalize* the concepts of corporate reorganization law as much as possible. Whenever it is convenient to create a new conceptual object, we will do so, setting up an internal symbol like S1 to stand for the postulated entity.

Consider now how to represent a stockholder's ownership interest in some particular class of stock. One possibility is to translate a proposition like "Phellis owns New Jersey stock" into the list (OWN PHELLIS S1), where S1 again names the stock issued by New Jersey. But a number of other stockholders also own portions of S1; Phellis himself only owns a certain number of shares of S1. So we do not say that Phellis owns S1 directly, but rather that he owns some other entity called P1 which is itself a "share-of" or a "piece-of" S1. Thus the proposition here would be stored as two lists: (OWN PHELLIS P1) and (PIECE-OF P1 S1). It is then possible to talk separately about what Phellis owns and what other stockholders own, and to attach to Phellis' interest another proposition describing how many shares it represents. Nominalization, the introduction of the new entity, has enabled us again to expand a complex concept to fit a particular need. Should the present formalism later prove inadequate in some other context, it could be expanded once more. In principle, at least, this procedure could be repeated to whatever degree of complexity and density is useful.

By now I have developed enough semantic conventions to represent a moderately complicated cluster of facts. Consider

For a discussion of these syntactic conventions, see C. WEISSMAN, *supra* note 54, at 5–24.

the proposition "Phellis owns 100 shares of common stock issued by New Jersey." In the present formalism this factual cluster could be stored as:

(CORPORATION NEW-JERSEY)
(ISSUE NEW-JERSEY S1)
(STOCK S1)
(COMMON S1)
(PIECE-OF P1 S1)
(NSHARES P1 100)
(OWN PHELLIS P1)

Notice how the objects in these lists are linked together in pairs by the predicates. PHELLIS and P1 are linked together by their appearance in the list with OWN; P1 and S1 are linked together by their appearance with PIECE-OF; NEW-JERSEY and S1 are linked together by ISSUE. Since these linkages can be represented in a diagram, as in Figure 1, the data structure here is generally referred to as a *semantic network*, a formalism that is often more convenient to understand and use than our initial disconnected lists. Both formalisms have an equivalent interpretation in the TAXMAN system, however. A list or network that is stored in the data base is taken to be "true." A list or network

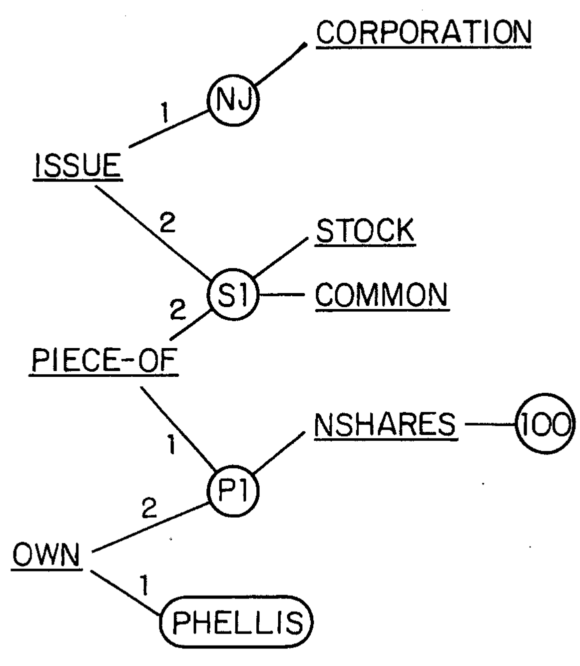

FIGURE 1

that is not stored in the data base, and thus not known to be true, is taken tentatively to be "false." The system can therefore be thought of as maintaining in its data base a "model" of the known "world" of corporations, stocks, stockholders, etc.

Let us now define some of the fundamental operations that the TAXMAN system ought to perform on the lists and networks in its data base in order to manipulate its "world model." I have already mentioned the simplest of these operations: at a minimum, we must be able to *store* a given list in memory, *retrieve* it from memory, and then, under some circumstances, *delete* it from memory again. The storage and deletion operations will be accomplished by the commands ASSERT and ERASE.[57] The command (ASSERT (CORPORATION NEW-JERSEY)) will call up a special program that stores and indexes the list (CORPORATION NEW-JERSEY), thus making it readily available for subsequent retrieval, or "true" in the "world" described by the data base. Similarly, the command (ERASE (CORPORATION NEW-JERSEY)) will remove the list (CORPORATION NEW-JERSEY) from memory, thus making it no longer available for retrieval, or "false" in that "world."

To retrieve a proposition from memory, a slightly more complicated GOAL command will be used. It is useful to distinguish several types of retrieval, depending on what information is to be retrieved and what information is available to guide the retrieval process. Thus, if we are considering the proposition (CORPORATION NEW-JERSEY) and want to know whether this proposition has already been asserted in the data base, the command (GOAL (CORPORATION NEW-JERSEY)) will provide the answer. The GOAL program will search the data base for the list (CORPORATION NEW-JERSEY) and, if successful, return T, otherwise NIL. Intuitively, the GOAL command tries to establish the "goal" of showing that the proposition "New Jersey is a corporation" is "true" in the given data base. This is the simplest form of retrieval: the straightforward verification that a particular assertion is present or absent in memory.

There are other retrieval possibilities, however. Suppose we wanted to find out whether there was an assertion in the data base about any corporation at all, whatever its name might be. We would then be asking whether a proposition of the form "x is a corporation," with "x" replaced by the name of a corporation, has been asserted in the data base, and this would be the case

[57] These ASSERT and ERASE commands (as well as some of the other operations in the discussion which follows, *e.g.*, GOAL, PROG, and THEOREM) are basic components of the Micro-PLANNER programming language.

if and only if a list of the form (CORPORATION X), with X replaced by the name of a corporation, appears in the data base. Accordingly, the command (GOAL (CORPORATION ?X))[58] would search the data base for a list whose first element matched CORPORATION, as before, but, if the match succeeded, the variable X would be replaced by the second element of the list, the name of the matching corporation. For example, if (CORPORATION X) is matched to (CORPORATION NEW-JERSEY), X would be replaced by the name NEW-JERSEY. We also need some mechanism to handle the possibility that propositions about two or more corporations may appear simultaneously in the data base. The original GOAL commands were designed to search for only one matching list, and thus if the data base contained both (CORPORATION NEW-JERSEY) and (CORPORATION DELAWARE), the program would return whichever one it happened upon first. As an alternative, we could use a command like (FIND ALL (CORPORATION ?X)), which would return a list of all possible matches, in this case the list ((CORPORATION NEW-JERSEY) (CORPORATION DELAWARE)).[59]

The preceding analysis is still relatively simple, applying only to single lists taken one at a time. I have defined a *data base*, in which we can formally represent concepts, and a set of *operations* on this data base, enabling us to add new data structures to it, modify old ones in it, and retrieve information generally about the state of affairs it represents. I have also introduced implicitly the idea of *pattern-matching*: a major feature of the GOAL commands was their ability to match a partially specified pattern in the GOAL statement against the fully specified patterns stored in the data base.

The next important step is to design a program that can store and retrieve a somewhat larger chunk of data structure, such as a network of related propositions, by matching patterns of greater structural complexity than the single lists which have so far appeared in the GOAL commands. Consider the problem of making a simple logical deduction from a set of propositions stored in the data base. Suppose the data base contains a representation of the network of Figure 1, and suppose we wish to establish the truth of the proposition "Phellis is a stockholder of New-Jersey."

---

[58] Throughout this Article the prefix "?" will be used to indicate that the letter to which it is attached is a variable. Thus "?X" in this example indicates that "X" functions as a variable in the GOAL command. Any letter or word without the prefix, *e.g.*, P1, S1, PHELLIS, or NEW-JERSEY, is a constant.

[59] These simple examples do not exhaust all the retrieval possibilities. *See* Bobrow & Raphael, *supra* note 55, at 164–66.

There is no single proposition in the network that expresses directly the relationship between a corporation and its stockholders, but the stockholder relationship is expressed indirectly by the sequence of propositions involving ISSUE, STOCK, PIECE-OF, and OWN. What is needed, therefore, is a command to search for all four of these related propositions in succession. Since this command will have some of the characteristics of an ordinary sequential computer program, it will be given the name PROG and required to have the following form: (PROG <list of variables> <sequence of operations>).[60] But the PROG command will also have some important differences from the ordinary computer program.

To use the PROG command to establish the truth of the proposition "Phellis is a stockholder of New Jersey," we can write out the following expression:

```
(PROG  (S P)
   (GOAL  (ISSUE NEW-JERSEY ?S))
   (GOAL  (STOCK ?S))
   (GOAL  (PIECE-OF ?P ?S))
   (GOAL  (OWN PHELLIS ?P)))
```

Here the list (S P) is interpreted as a list of variables to be used in the computation, in the same way that X was used as a variable in the earlier example of a GOAL command. The list of four GOAL commands is then interpreted as a list of operations to be executed, one after another, so long at least as each GOAL command is successful in its search of the given network. Thus, the first thing the program does here is to execute the command (GOAL (ISSUE NEW-JERSEY ?S)). This command will succeed in the given network, and the variable S will be matched to the object S1. Next, the command (GOAL (STOCK ?S)) is executed: here, since S has already been replaced by S1, this second command amounts to a search for the constant list (STOCK S1), which also succeeds. The third GOAL command then becomes a search for a list to match (PIECE-OF ?P S1), where the original variable S has been replaced by the constant object S1. The search succeeds, and the variable P is matched to the object P1. Finally, the fourth GOAL command becomes a simple search for the list (OWN PHELLIS P1), which is also

[60] The angled brackets are used here to indicate in skeletal form the type of information which the command requires. The term "ordinary computer program" is used here to refer to the types of programming languages that most people are most familiar with: FORTRAN, BASIC, ALGOL, PL/I, etc. A subroutine in these languages contains a sequence of operations to be executed and a list of variables to be used while the subroutine is running. PROG is an "ordinary computer program" in this respect.

easily satisfied. The net result, therefore, is the successful execution of the original PROG command and so a successful search for the stockholder relationship.[61]

If the network were more complex, showing the New Jersey corporation with several stocks and bonds outstanding, and showing each of these stocks and bonds with several distinct stockholders and bondholders, the search might not proceed so simply. For example, it is possible that the first GOAL command would be satisfied by retrieving the list (ISSUE NEW-JERSEY S7) and thus assigning to the variable S the value S7. But it may then turn out that the object named S7 is not a stock at all, but rather a bond, and so the subsequent search for (STOCK S7) would fail. In such a case, the PROG command has the ability to *backtrack*: it would back up to the first GOAL command and search there for another match. If it then found the list (ISSUE NEW-JERSEY S1) on this second turn, it would retrieve it, reassign S to the new object S1, and proceed forward again to search for (STOCK S1). Continuing in this manner, the program would eventually winnow out all the other incorrect matches, the wrong STOCKs, the wrong PIECEs, etc., until it arrived at a choice of S and P that would enable it to execute successfully the fourth command: (GOAL (OWN PHELLIS ?P)). The net result, again, would be the successful execution of the original PROG command.

To sum up: We want the program in this example to determine whether there are two objects satisfying the four propositions that make up the "stockholder" relation. To do this it must match the two variables and satisfy the four commands simultaneously, an essentially *parallel* task, but it accomplishes this result by a trial-and-error search, a *sequential* procedure.[62] This ability of the PROG command to find objects satisfying a sequence of propositions can be exploited further. In English we often use a *definite*

[61] Since all four of the GOAL statements must be satisfied in order to satisfy the PROG command, they can be thought of as being implicitly connected by a logical AND. Micro-PLANNER also provides for the explicit use of the other logical connectives: OR, NOT, and IF . . . THEN. It is thus possible to represent in a PROG command an arbitrarily complex proposition.

[62] The implementation of these search procedures in Micro-PLANNER has been sharply criticized, however. *See* Sussman & McDermott, *From PLANNER to CONNIVER—A Genetic Approach*, 1972 PROCEEDINGS OF THE FALL JOINT COMPUTER CONF. 1171 (Am. Fed'n of Information Processing Societies, Montvale, N.J.). Unrestricted backtracking can be highly inefficient if the PROG command is complex and the data base is large, since each new variable to be matched increases exponentially the length of the search. As an alternative, some of the more recent languages give the programmer explicit control over the order in which the search is to be carried out. *See, e.g.*, Bobrow & Raphael, *supra* note 55, at 156–58, 162–64.

description, such as "the New Jersey stock owned by Phellis," to single out an object about which we want to say something, as in "The value of the New Jersey stock owned by Phellis is $5000." We can make this assertion in TAXMAN with the following PROG command:

```
(PROG (S P)
   (GOAL (ISSUE NEW-JERSEY ?S))
   (GOAL (STOCK ?S))
   (GOAL (PIECE-OF ?P ?S))
   (GOAL (OWN PHELLIS ?P))
   (ASSERT (VALUE ?P 5000)))
```

As in the previous example, this program will first use a pattern-matching search to find an object P such that "New Jersey has issued S, S is a stock, P is a piece-of S, and Phellis owns P." In doing so it will replace P by P1, its name for that piece of stock, and then interpret the final command to be: (ASSERT (VALUE P1 5000)). The program thus represents quite naturally the assertion that "the value of the New Jersey stock owned by Phellis is $5000."

A far more significant application of the PROG pattern-matching capability, however, lies in the construction of a new concept out of a set of component concepts already present in a data structure, a simple form of conceptual *abstraction*. We have already seen how to express the proposition "Phellis is a stockholder of New Jersey" in terms of the component concepts ISSUE, STOCK, PIECE-OF, and OWN, but we did this without developing a representation of the abstract concept of a STOCKHOLDER itself. What we need now is a way to search for the STOCKHOLDER relationship as a high-level GOAL without having to specify all the lower-level searches through the data structure. This can be done by incorporating the basic mechanisms of the PROG command inside a structure called a THEOREM and having the following form: (THEOREM ABSTRACT <list of variables> <conceptual pattern> <sequence of operations>).

For example, using the THEOREM formalism, the STOCKHOLDER relationship can be defined as follows:

```
(THEOREM ABSTRACT (O C S P)
   (STOCKHOLDER ?O ?C)
      (GOAL (ISSUE ?C ?S))
      (GOAL (STOCK ?S))
      (GOAL (PIECE-OF ?P ?S))
      (GOAL (OWN ?O ?P)))
```

Intuitively, the theorem states that the concept of a STOCKHOLDER is abstractly equivalent to a particular pattern of the concepts ISSUE, STOCK, PIECE-OF, and OWN. Further, since the theorem incorporates the GOAL mechanisms of the PROG command, it shows how the relationship (STOCKHOLDER ?O ?C) can be established in any given network, for any objects O and C, by the familiar PROG pattern-matching process, assigning variables temporarily, backtracking, etc. The only difference between the use of this theorem and the use of the PROG command in our original example is the fact that the stockholder variable, O, and the corporation variable, C, can now be assigned initially to any objects we choose.

Suppose we wish to establish, as before, that "Phellis is a stockholder of New Jersey," but we know that the stockholder relationship has already been defined and stored in memory along with the usual factual network. We could then execute the command (GOAL (STOCKHOLDER PHELLIS NEW-JERSEY) ABSTRACT). This command would first search the data base for a direct match with (STOCKHOLDER PHELLIS NEW-JERSEY), but the search would fail. Second, since the word ABSTRACT was present in the command, the program would search for an abstraction theorem with a final deduction which matched the list (STOCKHOLDER PHELLIS NEW-JERSEY), in this case retrieving the definition of STOCKHOLDER with its matching pattern (STOCKHOLDER ?O ?C). The program would then assign to the variables O and C the values PHELLIS and NEW-JERSEY, respectively, and execute the four GOAL commands of the theorem as an ordinary PROG. The net result would be a successful determination that "Phellis is a stockholder of New Jersey," as before.[63] The other variations in the GOAL commands would operate analogously.[64]

---

[63] The use of the GOAL command together with the ABSTRACT theorem, as described in the text, is often referred to as a process of "pattern-directed function invocation," and it is generally considered to be one of the more important innovations of the PLANNER family of languages. *See* Bobrow & Raphael, *supra* note 55, at 159, 166–69, 170. The example in the text also illustrates what is referred to as the "procedural representation of knowledge," since the STOCKHOLDER theorem there not only expresses an abstract equivalence between two conceptual structures, but also provides a concrete computer program for establishing that equivalence in a particular network. For a discussion of the arguments for and against these procedural representations, see Winograd, *Frame Representations and the Declarative/Procedural Controversy*, in REPRESENTATION AND UNDERSTANDING 185, 186–93 (D. Bobrow & A. Collins eds. 1975).

[64] For example, the command (GOAL (STOCKHOLDER ?O NEW-JERSEY) ABSTRACT) would apply the STOCKHOLDER definition with one variable assigned and one unassigned, and thus would eventually succeed by matching the unassigned variable O to one of New Jersey's stockholders, maybe Phellis, maybe

I have used the STOCKHOLDER definition in this example as a means of proceeding from the detailed level of the original network up to a less detailed, more abstract level expressing only a single general relationship between a corporation and its stockholders. But in many situations we may want to reverse the procedure. We may want to assert as a general proposition that "Phellis is a stockholder of New Jersey" without worrying about the detailed representation in terms of STOCKs, PIECEs, and OWNership relations, and yet have these concrete details available later for other purposes. In the TAXMAN system, these lower-level structures could be generated automatically by an *expansion* theorem:

```
(THEOREM  EXPAND (O C . . . )
   (STOCKHOLDER ?O ?C)
   <body of ASSERTions>)
```

which could be called into effect by a pattern-matching search from the command (ASSERT (STOCKHOLDER PHELLIS NEW-JERSEY) EXPAND).

I have been deliberately vague about the main body of this theorem, however, since the necessary details are more complex here than they were in the previous deductive use of the STOCKHOLDER definition. Basically, the program should store the four assertions, (ISSUE NEW-JERSEY ?S), (STOCK ?S), (PIECE-OF ?P ?S), and (OWN PHELLIS ?P), which constitute the definitional expansion of (STOCKHOLDER PHELLIS NEW-JERSEY), but since the abstract STOCKHOLDER concept suppresses all information about the values of the variables S and P, there may be more than one concrete expansion possible. The program must therefore examine the context of whatever network already exists and make the best strategic choices possible.[65] Since any decisions made here could be contradicted by

someone else. Similarly, the command (FIND ALL (STOCKHOLDER ?O NEW-JERSEY) ABSTRACT) would locate and return a list of all of New Jersey's stockholders.

[65] If there have been no assertions at all stored yet about New Jersey stock, it would be reasonable to create a class of stock for Phellis to own shares of. If the existence of one class of stock has already been asserted, it may be reasonable to list Phellis as one of the owners. If two or more classes of stock are outstanding, however, the assertion is clearly ambiguous, and it may be best simply to note the ambiguity explicitly in the network. For a further discussion of the problems of expanding a relatively abstract concept to a more concrete level in a particular context, see Rieger, *Conceptual Memory and Inference*, in CONCEPTUAL INFORMATION PROCESSING 157 (R. Schank ed. 1975); Wilks, *A Preferential, Pattern-Seeking Semantics for Natural Language Inference*, 6 ARTIFICIAL INTELLIGENCE 53 (1975).

later assertions, the program should be capable of backing up later on to correct its mistakes.

The important point, though, is the basic process of abstracting and expanding concepts. There is not much power in a system that stores and retrieves single lists and single propositions. But as soon as the system can represent the relationship between two levels of conceptual abstraction, its power increases substantially. It is then possible to query the system at the higher level with a (GOAL <conceptual pattern> ABSTRACT) command, and have the lower-level searches carried out automatically; or state a fact at a higher level with an (ASSERT <conceptual pattern> EXPAND) command, and have the lower-level representations generated automatically. This process of conceptual abstraction can be extended to more than one level. The (GOAL (STOCKHOLDER ?O ?C) ABSTRACT) statement might thus appear as a component part of some other definition, and so at least two levels of data base searches could then be carried out automatically. Continuing in this way, an abstraction hierarchy of any desired complexity could be developed. Or multiple abstraction hierarchies could be developed, extending upwards from a common data base in different directions and for different purposes.

Perhaps even more interesting is the possibility of expanding the conceptual hierarchy downwards below the level of the concepts that have so far been treated as primitive. For example, the STOCK predicate has been used in this discussion as a basic, unanalyzed term, but it could itself be represented as an abstraction built up from a description of the rules governing the relationship between a STOCKHOLDER and a CORPORATION. A basic philosophy of the TAXMAN system is to treat concepts as primitives only provisionally, and only for certain purposes, and to retain the option of expanding them further in other contexts. In the current version of TAXMAN, however, STOCK is treated throughout as primitive.

The reader familiar with modern symbolic logic will have noted that there are strong similarities between these techniques for semantic information processing and the formalisms of the classical predicate calculus. Indeed, the power of TAXMAN to *represent* concepts in the data base is equivalent to that of the higher-order predicate calculus.[66] But since the *procedures* in

[66] *See generally* T. WINOGRAD, *supra* note 12, at 108–12; J.R. ANDERSON & G. BOWER, *supra* note 13, at 155–71; Sandewall, *Conversion of Predicate-Calculus Axioms, Viewed as Non-Deterministic Programs, to Corresponding Deterministic Programs*, PROCEEDINGS OF THE 3RD INT'L JOINT CONF. ON ARTIFICIAL INTELLIGENCE 230 (1973). For a rough idea of how this equivalence works, note that the

TAXMAN for the manipulation of these concepts have the flexibility of a general programming language, they are much more powerful than the proof procedures of symbolic logic. A programming language can express heuristic rules to guide the search for relevant theorems and propositions,[67] for example, or it can simulate in a natural way a dynamic change in the data base.[68] More important, it can function efficiently in an environment of structured complexity: because of the convention of pragmatic nominalization in TAXMAN, there tends to be a proliferation of objects in the data base, and conceptual definitions tend to be tied closely to these objects; the most important operations tend to be searches up and down the abstraction hierarchy for objects and concepts matching specified sets of definitional patterns, with the search process tightly controlled by the abstraction hierarchy itself. The techniques of semantic information processing are well designed for these purposes, but the formalisms of the predicate calculus are not.[69] Nevertheless, in assessing the adequacies and inadequacies of the TAXMAN paradigm, it should be kept in mind, for whatever limitations it may entail, that despite the additional power and flexibility of a general programming language, the representation of concepts in the current version of TAXMAN is still abstractly equivalent to the representation of concepts in the classical higher-order predicate calculus.[70]

STOCKHOLDER program, pp. 858–59 *supra*, could be represented in the (first-order) predicate calculus as follows:

(∀O) (∀C) ((∃S) (∃P) ((ISSUE C S) & (STOCK S) & (PIECE-OF P S) & (OWN O P)) ⊃ (STOCKHOLDER O C)).

See, e.g., I. Copi, Symbolic Logic (1954).

[67] *See, e.g.*, Winograd, *supra note* 63, at 190–91. For an illustration in the TAXMAN system, although an imperfect one, see the discussion of the search procedures in the B-REORGANIZATION program, pp. 874–76 *infra*.

[68] *See, e.g.*, Winograd, *supra* note 63, at 189–90. For an illustration in the TAXMAN system, see the discussion of the TRANS assertion, pp. 866–67 *infra*.

[69] For a thorough and systematic discussion, see Minsky, *A Framework for Representing Knowledge*, in The Psychology of Computer Vision 211 (P. Winston ed. 1975). Minsky argues that the theories of the more "logic-oriented" workers in artificial intelligence and cognitive psychology have been "too minute, local, and unstructured to account — either practically or phenomenologically — for the effectiveness of common sense thought," *id.* at 211, and then proposes a theory based on some larger and more highly structured entities called "frames." For an illustration in the TAXMAN system of a complex structure with some of the characteristics of Minsky's frames, see the discussion of the DISTRIBUTE concept, pp. 868–70 *infra*.

[70] There have been a number of attempts to apply formal symbolic logic to the analysis of legal concepts. Much of this work has focused on the most abstract legal concepts, "right," "duty," "privilege," etc., drawing on the important early analysis of Hohfeld, *Some Fundamental Legal Conceptions as Applied in Judicial*

III. Implementations

In this Part, I will show how the ideas of semantic information processing can, after further elaboration, be implemented in a computer program for the analysis of an actual case involving the taxation of corporate reorganizations.

There is a similarity between the preceding two sections which suggests how this might be done. The discussion in Part II emphasized the idea of a conceptual hierarchy, using the simple STOCKHOLDER example to show how the relationship between two or more levels of conceptual abstraction could be represented in computer data structures. The discussion in Part I of the taxation of corporate reorganizations revealed a similar hierarchy of abstractions. Our New Jersey-Delaware hypothetical and its variations were presented first in lower-level "factual" descriptions: transfers of stocks and assets, stock distributions and stock redemptions, and so on. The case was then analyzed in terms of several higher-level "legal" concepts: the concept of "control,"[71] the various statutory concepts of a "reorganization,"[72] and the judicial concepts of "continuity of interest,"[73] "business purpose,"[74] and "step transaction."[75] When the New Jersey-Delaware hypothetical was said to "fit the general pattern" of both a Type C and a Type D reorganiza-

Reasoning, 23 Yale L.J. 16 (1913). *See, e.g.*, Allen, *Formalizing Hohfeldian Analysis to Clarify the Multiple Senses of 'Legal Right': A Powerful Lens for the Electronic Age*, 48 S. Cal. L. Rev. 428 (1974); Anderson, *The Logic of Hohfeldian Propositions*, 33 U. Pitt. L. Rev. 29 (1971); Anderson, *The Logic of Norms*, 1 Logique et Analyse 84 (1958). None of this work has paid much attention to the problems of connecting these higher-level abstractions to the lower levels of the conceptual hierarchy, which is the major concern of the TAXMAN project. Another line of work in the legal literature has the disadvantage of using only the propositional calculus, which lacks the full expressive power of the predicate calculus. *See, e.g.*, Allen, *Symbolic Logic: A Razor-Edged Tool for Drafting and Interpreting Legal Documents*, 66 Yale L.J. 833 (1957); Allen & Orechkoff, *Toward a More Systematic Drafting and Interpreting of the Internal Revenue Code: Expenses, Losses and Bad Debts*, 25 U. Chi. L. Rev. 1 (1957). Popp & Schlink, *JUDITH: A Computer Program to Advise Lawyers in Reasoning a Case*, 15 Jurimetrics J. 303 (1975), and Note, *A Computer Method for Legal Drafting Using Propositional Logic*, 53 Tex. L. Rev. 965 (1975), are also based on the propositional calculus, and thus subject to the same comments.

[71] I.R.C. § 368(c); p. 845 *supra*.

[72] I.R.C. § 368(a); *see* pp. 844–45 *supra*.

[73] *See* Pinellas Ice & Cold Storage Co. v. Commissioner, 287 U.S. 462 (1933); Cortland Specialty Co. v. Commissioner, 60 F.2d 937 (2d Cir. 1932), *cert. denied*, 288 U.S. 599 (1933). *See* pp. 846–47 *supra*.

[74] *See* Gregory v. Helvering, 293 U.S. 465 (1935), *aff'g* 69 F.2d 809 (2d Cir. 1934); pp. 847–48 *supra*.

[75] *See* Helvering v. Elkhorn Coal Co., 95 F.2d 732 (4th Cir.), *cert. denied*, 305 U.S. 605 (1938); pp. 848–49 *supra*.

tion,[76] then, the analysis was similar to that of the simple STOCKHOLDER program: a search through the abstraction hierarchy to match a conceptual structure at one level to one at another level.

The current version of TAXMAN takes advantage of the similarity to apply the semantic information processing treatment of conceptual hierarchies directly to corporate reorganization law. The TAXMAN system is capable of accepting a *description* of a corporate reorganization case at roughly the level of the statement of facts in an appellate opinion and then producing an *analysis* of the case at roughly the level of abstraction of the desired legal conclusion. In doing this, the program makes use of the *expansion* and *abstraction* mechanisms, respectively. In general, the descriptions are received at a slightly higher level of abstraction than the program requires, and the expansion mechanisms are used to rewrite these descriptions more concretely at a lower level. Once the descriptions are fully expanded, the abstraction mechanisms are used to apply to them, if possible, the higher-level legal concepts.

It would be a mistake to rely too heavily on this very general characterization of the TAXMAN system, however, for the performance of the system obviously depends on our ability to give a precise and programmable content to these mechanisms for the "description" and "analysis" of a corporate reorganization case. Unless the concepts of corporate reorganization law can, at least in principle, be incorporated into these description and analysis mechanisms, this general characterization is vacuous. It is therefore important to demonstrate that at least some of the concepts of corporate reorganization law *can* be represented within the current paradigm of semantic information processing, to show in some detail *how* they can be represented within this paradigm, and thus to show by inference how the existing program can be *extended* to handle problems of substantially greater complexity. The practical and theoretical limits of these extensions should then be evident.

The remainder of this Part is devoted primarily to these purposes. One of the claims I have made for the power of a programming language is that it can be used to modify a conceptual structure in various ways.[77] In Part III-A, I will show how the TAXMAN system makes use of this capability to represent an "event" as the modification of the description of a "state," and how these event descriptions can themselves be assembled into

---

[76] *See* pp. 845–46 *supra.*

[77] *See* p. 862 *supra.*

conceptual hierarchies to represent more complex events. In Part III-B, focusing on the concept of a "stock distribution," I will continue this analysis of complex event descriptions and also illustrate how the TAXMAN system can take advantage of repetition and redundancy in building an abstraction hierarchy. In Part III-C, finally, I will consider the implementation of the "reorganization" definition itself, showing how the pattern-matching process can be constructed and constrained, and indicating what effects these constraints might have.

A. *States and Events*

In order to talk about the facts of a corporate reorganization case, it is necessary to talk about corporate transactions, or corporate events, in addition to the static corporate structures of Part II. It is natural to think of an "event" of this sort as a transformation from one "state" to another. For example, the event described by the proposition "New Jersey issues a certain class of stock" could be viewed as a transformation from a state in which New Jersey has no such class of stock outstanding to a state in which it does. The TAXMAN system provides a simple way to implement these ideas in its mechanisms for *state descriptions* and *event descriptions*.

First, to capture the idea of a state description, the system allows the insertion of an optional extra variable at the end of a proposition to indicate that the proposition is true at a certain time: the proposition (ISSUE NEW-JERSEY S1 T4), for example, indicates that (ISSUE NEW-JERSEY S1) is true at time T4.[78] The network of all propositions in the data base that are true at a certain time can then be thought of as a description of the state of the "world" at that time.[79] With this data base available, the event descriptions play a dual role. As straightforward *representations* of events, the event descriptions have a hierarchical structure analogous to the structure of the state descriptions: certain simple events are taken as primitive, and more complex events are built up systematically from complex

[78] The same proposition without the final variable would indicate that (ISSUE NEW-JERSEY S1) was true at all times during the period being described.

[79] For reasons of computational efficiency, the data base contains only one network, but each proposition in the network has attached to it a separate list describing the time periods during which it is "true," and the ASSERT, ERASE, and GOAL commands have been modified to check these listings of time-dependent truth values whenever necessary. To implement this I have rewritten the standard Micro-PLANNER interpreter for the ASSERT, ERASE, and GOAL commands. Other languages provide a general CONTEXT mechanism which allows the user to maintain more automatically these multiple state descriptions. *See, e.g.*, Bobrow & Raphael, *supra* note 55, at 161.

patterns of these primitives. But event descriptions also function as *programs* or *procedures* that actually carry out the transformation from one state description to another, rewriting the computer data structures to correspond symbolically to the transformation of states. For example, if the New Jersey corporation has no stock outstanding at a particular time, and if a subsequent event description represents the issuance by New Jersey of a certain class of stock, the TAXMAN system would interpret this event description as a program and write out a modified network for the subsequent time period in which a proposition of the form (ISSUE NEW-JERSEY ?S <time>) would be asserted as true.[80] Technically, the event description would be expanded by a command of the form (ASSERT <event> EXPAND), with the expansion theorem producing as its expansion the description of the modified state.

Using these mechanisms it is possible to write out the *history* of a corporate transaction. We first write out an initial state description, then write out a list of event descriptions, arranged in chronological order, and finally expand the event descriptions in order, thus producing a full sequence of modified state descriptions. In this way the TAXMAN system can have available in its data base information not only about current states and events, but about all prior states and events as well, an essential requirement for the application of the higher-level concepts of corporate reorganization law.

To see in greater detail how the event descriptions operate, let us examine the class of events involving the transfer of ownership from one individual to another, beginning with the primitive concept TRANS. For present purposes, the simplest event in the system will take the form (TRANS ?T ?P ?O ?R <time>), which is interpreted to mean "T transfers P from O to R at a certain time." The variable T represents the transferor; P represents the property being transferred; O is the former owner; and R is the new recipient. Or, more mnemonically, the proposition could be written as (TRANS <transferor> <property> <owner> <recipient> <time>).[81]

---

[80] In addition, any proposition not explicitly modified by the event description would be carried over unchanged into the modified state description network. The event descriptions thus produce only "incremental" changes in the state descriptions. For a discussion of these conventions, see T. WINOGRAD, *supra* note 12, at 116–17. For a general discussion of event descriptions, see Bruce, *Case Systems for Natural Language*, 6 ARTIFICIAL INTELLIGENCE 327, 331–33 (1975).

[81] Notice that TRANS refers here only to a transfer of "ownership," not to a transfer of "possession," or a transfer of "location," or any other sense of the word "transfer." In the system of Schank and his collaborators there are three basic TRANS events: "abstract transfer," "physical transfer," and "mental trans-

By definition, an expansion theorem for the TRANS proposition requires the modification of the ownership relationships from a prior to a subsequent state, and the essential aspects of the TRANS theorem could thus be written as follows:

```
(THEOREM EXPAND (T P O R TA TB)
  (TRANS ?T ?P ?O ?R ?TB)
  (IF        (GOAL     (OWN ?O ?P ?TA))
     THEN (ERASE    (OWN ?O ?P ?TB))
              (ASSERT (OWN ?R ?P ?TB))))
```

Suppose we wish to assert and expand the proposition "PHELLIS transfers the property P1 to DELAWARE," given the prior state description network of Figure 1. The desired event could be represented by the command (ASSERT (TRANS PHELLIS P1 PHELLIS DELAWARE <time>) EXPAND). Recall from the discussion in Part II how to trace through the effects of such a command: the basic TRANS proposition would be added to the factual data base first; then the TRANS theorem above would be retrieved and the variables matched; then the main body of the theorem would be executed with O as PHELLIS, P as P1, R as DELAWARE, and with TA and TB representing, respectively, the prior and subsequent states. In the execution of the theorem, the IF . . . THEN statement would carry out the main substantive operation: the IF clause would check to see if PHELLIS owned P1 in the prior state; if so, as in the present case, the THEN clause would proceed to erase PHELLIS' ownership and assert in the subsequent state description that DELAWARE now owned P1 instead. In short, stripping away the technical details, the TRANS assertion does exactly what we would expect: it changes the proposition (OWN PHELLIS P1) into the proposition (OWN DELAWARE P1), and writes out the modification in the subsequent state description network.

Once the basic TRANS operation is defined, it becomes possible to assemble some of the more complicated structures involving the transfer of property. One pattern of TRANS propositions that is particularly useful in corporate reorganization cases involves the transfer of only a portion of an individual's interest in a piece of property. Suppose we wish to assert the proposition "Phellis transfers 20 shares of the property P1 to Delaware." Intuitively, we could imagine that Phellis' interest in the property P1 is "split" into two "pieces," with only one

fer." Using these and several more "primitive actions," Schank is able to represent a surprising number of English verbs. Schank, *Conceptual Dependency Theory*, in CONCEPTUAL INFORMATION PROCESSING 22, 40–67 (R. Schank ed. 1975).

of these pieces transferred to Delaware. Thus the proposition could be represented as follows:

```
(PROG (P)
    (ASSERT (NSHARES ?P 20))
    (ASSERT (SPLITPIECE ?P P1 <time>) EXPAND)
    (ASSERT (TRANS PHELLIS ?P PHELLIS DELA-
      WARE <time>) EXPAND))
```

In this example, SPLITPIECE is a special event description which creates a new piece of property, P, of size NSHARES = 20, and adjusts the ownership relationships in the networks so that Phellis owns both the original piece of stock, P1, reduced in size now by 20 shares, and the new piece, which can then be transferred. The final command, (ASSERT (TRANS PHELLIS ?P PHELLIS DELAWARE <time>) EXPAND), carries out the transfer of the 20-share piece of stock, as desired.

B. Description Mechanisms

To illustrate the construction of event descriptions of greater complexity than the TRANS assertion, I will analyze here the concept of a "stock distribution," a relatively abstract event, and show how the TAXMAN system can expand it downwards to a more concrete level.

Intuitively, a "distribution" of any kind consists of the splitting-up and piece-by-piece transfer of a certain "property" from its current "owner" to a certain "class of recipients." For example: "The New Jersey corporation distributes its Delaware common stock to its own common stockholders." Accordingly, a distribution could be represented at a relatively abstract level by a proposition of the form (DISTRIBUTE <transferor> <property> <owner> <class of recipients> <time>), where both the property transferred, *e.g.*, "the Delaware common stock owned by New Jersey," and the class of recipients, *e.g.*, "the common stockholders of New Jersey," could be represented by a definite description in the manner discussed earlier. At a more concrete level, a distribution could be represented by a sequence of particular transfers from one particular individual to another, or a sequence of particular TRANS assertions.

The TAXMAN system should be able to expand this concept from the first level down to the second. If P1 represents the total property to be distributed, then the distribution to any particular recipient, R1, could be represented by:

```
(ASSERT (SPLITPIECE ?P P1 <time>) EXPAND)
(ASSERT (TRANS <transferor> ?P <owner> R1
  <time>) EXPAND)
```

and the full distribution could be represented simply by repeating this partial transfer one time for each recipient. The expansion program for DISTRIBUTE, then, would simply examine the description of the class of recipients and, for each R that satisfied this description, generate an appropriate pair of SPLITPIECE-TRANS assertions, as above.[82] The result would be a list of concrete SPLITPIECE-TRANS events which could then be executed sequentially to generate a modified state description.

In a typical stock distribution, though, stock is transferred according to a formula, "two-shares-for-one," "pro rata," etc., which takes into account the proportionate ownership of the recipients in another security, and the TAXMAN system needs a mechanism to represent these concepts as well. At the concrete level of the SPLITPIECE-TRANS events we could use the NSHARES predicate to represent the number of shares of stock transferred to each recipient:

```
(ASSERT (NSHARES ?P <number>))
(ASSERT (SPLITPIECE ?P P1 <time>) EXPAND)
(ASSERT (TRANS <transferor> ?P <owner> R1
  <time>) EXPAND)
```

What we need, then, is a way to represent a proportionate stock distribution at a more abstract level, and a way to expand this abstraction into a sequence of concrete NSHARES-SPLITPIECE-TRANS assertions. To do this, the TAXMAN system takes advantage of the fact that we generally use a single description to define both the class of recipients of a stock distribution and the rule expressing the proportionality of the distribution. For example, if the recipients are defined as "the common stockholders of New Jersey," the distribution would typically be made in proportion to each recipient's ownership of "the common stock of New Jersey," or, more colloquially, the distribution would be made to the New Jersey common stockholders "with respect to their stock." Suppose we are describing an "n-shares-for-one" distribution. Then we could write out at the more abstract level

[82] The recipient class of "the common stockholders of New Jersey" would be represented, using the descriptions discussed in Part II, pp. 857–58 *supra*, by means of the pattern:

```
(GOAL (ISSUE NEW-JERSEY ?S))
(GOAL (STOCK ?S))
(GOAL (COMMON ?S))
(GOAL (PIECE-OF ?P ?S))
(GOAL (OWN ?R ?P))
```

The DISTRIBUTE program would simply match this pattern to the network; it would thereby FIND ALL the objects which match the variable R; and it would then write out a pair of SPLITPIECE-TRANS assertions for each object located.

a DISTRIBUTION-RULE that gives each recipient N shares of the distributed stock for each ONE share of the security with respect to which the distribution is made: (DISTRIBUTION-RULE <distribution> (N FOR ONE)). Since the expansion program for DISTRIBUTE examines the description of the recipient class to determine the names of the recipients in the SPLITPIECE-TRANS assertions, the expansion program for the DISTRIBUTION-RULE would simply examine this same description more thoroughly to determine how many shares of the defining security each recipient owned.[83] A simple arithmetical calculation would then provide the proper value of NSHARES.

Finally, if the distribution rule is "pro rata," (DISTRIBUTION-RULE <distribution> (PRORATA)), the expansion program could proceed by calculating the number N in the preceding (N FOR ONE) rule, and then applying the (N FOR ONE) rule as before. To do this, it would have to examine both the description of the recipient class and the description of the property to be distributed, for it would be necessary to compute the total number of shares involved in each description. This calculation, however, would clearly be manageable.

Notice how complex this conceptual hierarchy has become in what is still a fairly simple example. The structural "core" of the concept, the sequence of pairs of SPLITPIECE-TRANS assertions, is "modified" by the insertion of the appropriate NSHARES propositions, all of which have been generated by the expansion of a DISTRIBUTION-RULE. The DISTRIBUTION-RULE, in turn, may have been generated by the expansion of a still more abstract concept, such as PRORATA, describing the type of distribution in question. Finally, when the structural core and the prescribed modifications have all been assembled, they are themselves expanded to produce in the data base a modified state description network. It is this compression into a higher-level description of complex lower-level information which gives the conceptual hierarchy its power and flexibility, and it is the existence of recurrent patterns in the concept of a stock distribution itself which enables us to structure and stratify the hierarchy in this way.

83 For example, when the DISTRIBUTE program matches the STOCKHOLDER pattern in note 82, *supra*, to the network to locate a recipient, R, it also locates the recipient's ownership interest, P, and so can easily determine the number of shares R owns. Of course, the recipient class might conceivably be defined as "the left-handed dentists in Atlantic City," in which case the search for the recipients might still succeed, given an appropriate data base, but the usual concept of a proportional stock distribution would be meaningless.

C. *Analysis Mechanisms*

I will now assume that the history of a corporate reorganization case has been stored in its expanded form in the data base, and turn to the mechanisms that can analyze the case in terms of the operative concepts of the Internal Revenue Code. The simplest example to start with is the concept of CONTROL. By statute, CONTROL is defined as:

> the ownership of stock possessing at least 80 percent of the total combined voting power of all classes of stock entitled to vote and at least 80 percent of the total number of shares of all other classes of stock of the corporation.[84]

It should be clear why this concept is relatively easy to program into the TAXMAN system: it is a concept applying only to a single state description, has an algebraic formula at its core, and is composed of essentially the same elements that entered into the analysis of the concept of STOCKHOLDER.

Suppose that (CONTROL ?X ?Y <time>) is used to represent the proposition that "X controls Y" at a certain time. A first step in the application of this concept would be to check whether X owns any stock of Y at all. This could be accomplished most easily with the command (GOAL (STOCKHOLDER ?X ?Y <time>) ABSTRACT). If the state description passes this test, the CONTROL program must then determine the total stock ownership of X and compare this with the total amount of stock issued by Y. The program must therefore make an exhaustive search to locate all the stock issued by Y, *i.e.*,

```
(GOAL (ISSUE ?Y ?S <time>))
(GOAL (STOCK ?S <time>))
```

and divide that stock into two classes depending on whether or not the proposition (VOTING ?S <time>) is true. Then, for each stock S, whether voting or nonvoting, the program would locate the total ownership interest of X, *i.e.*,

```
(GOAL (PIECE-OF ?P ?S <time>))
(GOAL (OWN ?X ?P <time>))
```

and calculate the total number of votes or the total number of shares, respectively, for each such P. Finally, sorting out the S's and the P's and the votes and the shares, the two separate calculations and comparisons of the "80 percent" figure would be performed in a purely mechanical fashion.

---

[84] I.R.C. § 368(c).

For a more complex example, consider the concept of a B-REORGANIZATION. By statute, a Type B reorganization is defined as:

> the acquisition by one corporation, in exchange solely for all or a part of its voting stock (or in exchange solely for all or a part of the voting stock of a corporation which is in control of the acquiring corporation), of stock of another corporation if, immediately after the acquisition, the acquiring corporation has control of such other corporation (whether or not such acquiring corporation had control immediately before the acquisition).[85]

The representation of this complex concept must, like the representation of the DISTRIBUTE concept, build on the several lower levels of abstraction in the structure that makes it up. An acquisition seems to have the basic structure of an "exchange" or mutual "transfer" of property. More exactly, since it may consist of several exchanges, we can represent it by an expression of the form (ACQUISITION <list <exchange>$_i$>), where the subscript *i* indicates that the list may consist of one, two, or several members. An exchange, in turn, has the basic structure of a pair of "transfers," or rather, since an exchange may involve the mutual transfer of several objects, a pair of lists of transfers: (EXCHANGE <list <trans>$_j$> <list <trans>$_j$>). Putting all this together, an acquisition could be represented as follows:

(ACQUISITION
    <list (EXCHANGE
        <list (TRANS ?T_1 ?P_1 ?O_1 ?R_1 <time>) . . . >
        <list (TRANS ?T_2 ?P_2 ?O_2 ?R_2 <time>) . . . >)
  . . . .>)

Consider now the problem of matching the ACQUISITION structure to the given network. Starting at the innermost level, the TRANS propositions would be matched with the command (GOAL (TRANS ?T ?P ?O ?R <time>) ABSTRACT). But the assignment of variables in this matching process cannot be entirely unconstrained. Since a pair of lists of TRANS propositions is intended to represent a mutual transfer, the recipient, R, of one transfer must be the same as the prior owner, O, of the other. To ensure this, the EXCHANGE program would require that within a single list of TRANS propositions, all the O's be identical and all the R's be identical; that the R_1's from the first member of a pair of lists be identical to the O_2's from the second member of the pair; and that the R_2's from the second be

[85] I.R.C. § 368(a)(1)(B).

identical to the O_1's from the first. Once the EXCHANGE structure is fully matched, then, the program obtains a single recipient, R_1 ($=O_2$), who has exchanged certain properties with another single recipient, R_2 ($=O_1$). By convention, we will take R_1, the recipient in the first list of TRANS propositions, to be the "acquirer" in the final ACQUISITION. If the ACQUISITION consists of multiple exchanges, so that there is more than one EXCHANGE to R_1, these exchanges would all be located and listed in the ACQUISITION structure; if not, the ACQUISITION structure would consist of only one EXCHANGE.

With the concept of an ACQUISITION represented in this way, it is now a relatively simple matter to represent the remainder of the B-REORGANIZATION definition: a B-REORGANIZATION is just the result of imposing a further set of constraints on the ACQUISITION structure. One constraint is the statutory requirement that the acquisition be "by one corporation," which means simply that R_1, the acquirer located by the ACQUISITION match, must be a corporation, or: (GOAL (CORPORATION R1)). Another is the requirement that the acquisition be "of stock of another corporation," which means that each P_1 which is transferred to R_1 in the first list of TRANS propositions should be a PIECE-OF a STOCK which is ISSUEd by some other CORPORATION. A more crucial constraint is that the acquisition must be:

> in exchange solely for all or a part of [the acquiring corporation's] voting stock (or in exchange solely for all or a part of the voting stock of a corporation which is in control of the acquiring corporation) . . . .

To apply this test, the B-REORGANIZATION program must locate all the P_2's in the second list of TRANS propositions, *i.e.*, all the properties transferred by R_1 ($=O_2$) "in exchange for" the P_1's, and determine that these P_2's are either (a) each a PIECE-OF a VOTING STOCK which is ISSUEd by R_1, or (b) each a PIECE-OF a VOTING STOCK which is ISSUEd by another CORPORATION in CONTROL of R_1. The machinery necessary to implement this "solely for voting stock" requirement is already available and can simply be arranged in the appropriate patterns. Finally, consider the constraint that "immediately after the acquisition, the acquiring corporation has control of the acquired corporation." Again, the components of this test are all in place: the "time" of the acquisition has been recorded in the lists of TRANS propositions; the "acquiring corporation," R_1, has been previously identified, and the "acquired corporation" has been located in the course of the search for the issuer of the stock with piece P_1. We now need only add the specification: (GOAL (CONTROL <acquiring corporation> <acquired corporation> <time>) ABSTRACT).

For expository purposes I have been describing in the preceding paragraphs a particular sequence of operations: first the matching of the ACQUISITION concept to the network, then the imposition of the B-REORGANIZATION constraints. But in many situations this would be a highly inefficient procedure. For example, it would be highly inefficient for the program to locate first a large number of ACQUISITION structures and only then discover that these were not acquisitions of STOCK, or not even acquisitions by a CORPORATION. Instead, ideally, the B-REORGANIZATION constraints should be checked at the same time that the ACQUISITION structure is being matched. The current version of TAXMAN takes a step in this direction, in fact, by specifying a particular order in which the B-REORGANIZATION and ACQUISITION searches are to be performed. The program first searches for a relationship of CONTROL between two corporations and locates the earliest time at which this control exists; second, it searches backwards in time to locate the transfers of stock, if any, to the controlling corporation; third, for each such transfer of stock to the controlling corporation, it locates the transfers made in exchange and determines whether or not these are all solely for voting stock. If any of these initial decisions leads to a dead end, the program backs up to the appropriate point and starts the search over again from there.[86]

There are various substantive legal questions reflected in this choice of a search procedure. The problems of "creeping acquisitions," [87] for example, are reflected in the structural ambiguities

[86] This procedure appears to be reasonably efficient in the great majority of situations we would expect to encounter. A more sophisticated version of the program, beyond the current capacity of TAXMAN, would be able to alter its search procedures as it goes along, depending on the specific information it is searching for and the specific knowledge it has already obtained.

[87] Prior to 1954, the Type B reorganization was defined as "the acquisition by one corporation . . . of at least 80 per centum of the voting stock and at least 80 per centum of the total number of shares of all other classes of stock of another corporation," Int. Rev. Code of 1939, ch. 1, § 112(g)(1)(B), 53 Stat. 40, and this language seemed to require that the entire 80 percent controlling interest in the acquired corporation be purchased in a single solely-for-voting-stock transaction, thus precluding the use of a Type B reorganization by a corporation that already owned more than 20 percent of the acquired corporation's shares. *See* Lutkins v. United States, 312 F.2d 803 (Ct. Cl.), *cert. denied*, 375 U.S. 825 (1963); Robert A. Pulfer, 43 B.T.A. 677 (1941), *aff'd per curiam*, 128 F.2d 742 (6th Cir. 1942). *But see* Charles A. Dana, 36 B.T.A. 97 (1937), *aff'd*, 103 F.2d 359 (3d Cir. 1939). The 1954 statute was intended to reverse this result by specifying only that the acquiring corporation have control "immediately after the ac-

of the B-REORGANIZATION concept as it has been described so far: first, the ACQUISITION structure is an open-ended list of EXCHANGE relationships, and it is unclear how far back into the past or forward into the future this list should be extended; second, the CONTROL relationship can exist over an extended period of time in the history of a corporate transaction, and the process of obtaining control can sometimes be a lengthy one. When we specify the order and scope of the search procedures for the ACQUISITION and CONTROL concepts, therefore, we determine which exchanges will be included in a qualifying B-REORGANIZATION. Actually, the current TAXMAN system does a rather poor job here, since it fails to embody accurately any of the current rules on these issues.[88] The important point, though, is that the current formalism reveals all the possibilities quite clearly, and thus provides a means of examining in concrete detail the possible solutions. A feasibly extended version of the TAXMAN system could contain further rules,

quisition . . . (whether or not such acquiring corporation had control immediately before the acquisition)." There were additional problems under the 1954 statute, however. Would a prior acquisition of stock for cash disqualify from reorganization treatment a subsequent acquisition made solely for voting stock? Conversely, would a prior acquisition of stock in exchange for voting stock qualify as part of a subsequent tax-free reorganization? The Treasury Regulations would ignore the prior acquisition for cash if it was clearly separate and occurred well before, *e.g.*, 16 years before, the claimed reorganization, and would include the prior acquisition for voting stock if it occurrred "in a series of transactions taking place over a relatively short period of time such as 12 months." Treas. Reg. § 1.368-2(c) (1960). *See generally* BITTKER, *supra* note 18, ¶ 14.33 to .34; Kanter, *Cash in a "B" Reorganization: Effect of Cash Purchases on "Creeping" Reorganization*, 19 TAX L. REV. 441 (1964); MacLean, *"Creeping Acquisitions,"* 21 TAX L. REV. 345 (1966).

[88] Most significantly, the current TAXMAN system contains no explicit provision for decomposing a sequence of exchanges into two or more separate acquisitions. Instead, the treatment of "creeping acquisitions" is left by default to the somewhat haphazard interaction of the ACQUISITION and CONTROL structures, according to the search procedure described in the text, pp. 871–74 *supra*. At first glance, this procedure might appear to implement the 1939 rather than the 1954 statute, *see* note 87 *supra*, since it seems to search only for acquisitions which precede the time of earliest control. But after first searching backwards from the earliest control relationship, the program can then search forward for any later control relationships which are themselves preceded by prior acquisitions, and in this way locate an entire sequence of qualifying Type B reorganizations, if such a sequence exists. This forward search will continue until the control relationship no longer exists, or until an exchange is made which is not solely for voting stock, a single mechanism for terminating the future extent of the ACQUISITION structure. It should be evident from this brief discussion that the problems of "creeping acquisitions" were not carefully thought out at the time the current TAXMAN system was designed, but it is not implausible to suggest that the problems of interpretation under the 1939 and the 1954 statutes have a similar origin: the unanticipated consequences of choosing a particular mode of expression for the "acquisition" of "control."

expressed in higher-level concepts, about "creeping acquisitions." [89]

IV. Results

Let us now observe the TAXMAN system as it describes and analyzes an actual case involving the taxation of corporate reorganizations.[90] TAXMAN is an on-line real-time computer program, and the description mechanisms and analysis mechanisms correspond quite literally to the operations that can be performed by a user of the program typing at a terminal. The user types out a description of the case in the programming language we have been working with throughout,[91] starting with an initial state description and proceeding with a sequence of event descriptions.[92] The user then interrogates the system with the various GOAL commands to determine whether or not — and why or why not — the case fits the pattern of a Type B, a Type C, or a Type D reorganization.

I will use as an illustration *United States v. Phellis*,[93] one of the earliest reorganization cases, and the case upon which our original New Jersey-Delaware hypothetical was based. Since the original *Phellis* case was decided under a statute which con-

---

[89] At a minimum, these rules could incorporate the simple Treasury Regulation, *see* note 87 *supra*, to distinguish a single acquisition from two or more separate acquisitions.

[90] The TAXMAN system was programmed for and run on the PDP-10 computer at the Stanford Artificial Intelligence Laboratory. *See* notes * and 3, *supra*. The example of a corporate reorganization case that follows occupied a core image of approximately 40,000 words on that machine, of which approximately 20,000 words were available for the assertions and theorems necessary to describe and analyze the case. The abstraction and expansion theorems were expressed in approximately 450 lines of code in Micro-PLANNER and LISP, *see* note 54 *supra*; and the case description itself required slightly over 200 lines of code in addition. The illustrations in the text generally took less than 30 seconds to process, when the time-sharing system was lightly loaded. The Micro-PLANNER programming language is known to be slow and inefficient, however, *see, e.g.*, Sussman & McDermott, *supra* note 62, and a moderate increase in the complexity of the TAXMAN system would have produced a substantial degradation of performance. Complete listings of the program and a sample terminal session are available from the author upon request.

[91] *See* notes 54–55 *supra*. It is tedious to write case descriptions in the Micro-PLANNER programming language, and until this requirement can be relaxed a system like TAXMAN will be accessible only to a small number of specialists. For a discussion of the possibilities of incorporating a natural language capability into a system like TAXMAN, see pp. 886–88 *infra*.

[92] *See* p. 866 *supra*. The requirement that these event descriptions be written out in chronological order is a serious restriction. For a discussion of the possibilities of relaxing this requirement, see pp. 884–85 & notes 112–114 *infra*.

[93] 257 U.S. 156 (1921).

tained no provisions for a tax-free corporate reorganization, the TAXMAN system will make no attempt here to reproduce the original decision, but will instead attempt to analyze the case as it would be decided under the present Code. I have already pointed out some of the ways in which the TAXMAN analysis is imperfect and incomplete,[94] and these will be confirmed by the illustration; but the illustration should also suggest the potential power of the approach.

The following is a statement of the relevant facts in *United States v. Phellis*:

> Mr. Justice Pitney delivered the opinion of the court. The court below sustained the claim of C. W. Phellis for a refund of certain moneys paid by him under protest in discharge of an additional tax assessed against him for the year 1915, based upon alleged income equivalent to the market value of 500 shares of stock of a Delaware corporation called E. I. du Pont de Nemours & Company, received by him as a dividend upon his 250 shares of stock of the E. I. Du Pont de Nemours Powder Company, a New Jersey corporation. The United States appeals.
>
> From the findings of the Court of Claims, read in connection with claimant's petition, the following essential facts appear. In and prior to September, 1915, the New Jersey company had been engaged for many years in the business of manufacturing and selling explosives. Its funded debt and its capital stock at par values were as follows:

5% mortgage bonds	$ 1,230,000
4½% 30-year bonds	14,166,000
Preferred stock ($100 shares)	16,068,600
Common stock ($100 shares)	29,427,100
Total	$60,891,700

> . . . In that month a reorganization and financial adjustment of the business was resolved upon and carried into effect with the assent of a sufficient proportion of the stockholders, in which a new corporation was formed under the laws of Delaware with an authorized capital stock of $240,000,000 to consist in part of debenture stock bearing 6 per cent. cumulative dividends, in part of common stock; and to this new corporation all the assets and good-will of the New Jersey company were transferred as an entirety and as a going concern, as of October 1, 1915, at a valuation of $120,000,000, the new company assuming all the obligations of the old except its capital stock and funded debt. In payment of the consideration, the old company retained $1,484,100 in cash to be used in redemption of its outstanding 5% mortgage bonds, and received $59,661,700 par value in debenture stock of the new company (of which

[94] *See* pp. 875–76 & notes 88–89 *supra*.

> $30,234,600 was to be used in taking up, share for share and dollar for dollar, the preferred stock of the old company and redeeming its 30-year bonds), and $58,854,200 par value of the common stock of the new company which was to be and was immediately distributed among the common stockholders of the old company as a dividend, paying them two shares of the new stock for each share they held in the old company. . . . Each holder of the New Jersey company's common stock (including claimant), retained his old stock . . . and the New Jersey corporation retained in its treasury 6 per cent. debenture stock of the Delaware corporation equivalent to the par value of its own outstanding common stock. . . . After the reorganization and the distribution of the stock of the Delaware corporation, the New Jersey corporation continued as a going concern, and still exists but, except for the redemption of its outstanding bonds, the exchange of debenture stock for its preferred stock, and the holding of debenture stock to an amount equivalent to its own outstanding common and the collection and disposition of dividends thereon, it has done no business. It is not, however, in process of liquidation. It has received as income upon the Delaware company's debenture stock held by it, dividends to the amount of 6% per annum, which it has paid out to its own stockholders including the claimant. . . . The fair market value of the stock of the Delaware corporation distributed as aforesaid was on October 1, 1915, $347.50 per share. The Commissioner of Internal Revenue held that the 500 shares of Delaware company stock acquired by claimant in the distribution was income of the value of $347.50 per share and assessed the additional tax accordingly. . . .[95]

Some of these facts could not be represented at all in the current TAXMAN system. For example, it is not possible to express the fact that the New Jersey company "had been engaged for many years in the business of manufacturing and selling explosives," or that the plan of reorganization was "carried into effect with the assent of a sufficient proportion of the stockholders." But it should also be clear that the operative facts of the case could indeed be represented in TAXMAN, using the basic description mechanisms we have examined so far. For example, the initial state description consists of a New Jersey corporation with common and preferred stock and two classes of bonds outstanding, and one identified stockholder, Phellis, who owns 250 shares of the common stock. This is only slightly more complicated than the simple example discussed earlier in the text. The several event descriptions in the case are also only slight variations of our earlier examples.[96] When the *Phellis*

[95] 257 U.S. at 165–68.

[96] There is a transfer of "all the assets and good will of the New Jersey com-

case is written out in full in the TAXMAN system and expanded in full, it produces only a moderately complex network.[97]

Assuming now that the full description is expanded and stored in the data base, let us look at the process of applying the analysis mechanisms to it. If we query the system first with (GOAL (B-REORGANIZATION ?A ?C ?T) ABSTRACT), where A is taken to be the acquiring corporation, C is taken to be the acquired corporation, and T is taken to be the time at which the reorganization is complete, the system will search for a Type B reorganization, at any time, with either Delaware or New Jersey as the acquiring corporation. The search for (B-REORGANIZATION DELAWARE NEW-JERSEY ?T) will fail, since Delaware never acquires any stock of New Jersey. In terms of the order of search described earlier, the failure occurs in searching for (CONTROL DELAWARE NEW-JERSEY ?T); and within the CONTROL theorem, the failure occurs in searching for (STOCKHOLDER DELAWARE NEW-JERSEY ?T). The search for (B-REORGANIZATION NEW-JERSEY DELAWARE ?T) will also fail, but for a more complex reason. Here the system finds (CONTROL NEW-JERSEY DELAWARE PHE32), and then finds the transfers of stock that have preceded this establishment of control. The result is an acquisition structure as follows:

```
(EXCHANGE
    ((TRANS DELAWARE PHE29 DELAWARE NEW-
      JERSEY PHE30)
    (TRANS DELAWARE PHE31 DELAWARE NEW-
      JERSEY PHE32))
    ((TRANS NEW-JERSEY PHE26 NEW-JERSEY
      DELAWARE PHE28)))
```

PHE29 and PHE31 here represent the debenture stock and the common stock, respectively, that were newly issued by Delaware

pany . . . at a valuation of $120,000,000"; the fact that the New Jersey company receives "$59,661,700 par value in debenture stock of the [Delaware] company . . . and $58,854,200 par value of the common stock of the [Delaware] company"; a redemption of the 5 percent mortgage bonds with cash; and a redemption of the preferred stock and the 30-year bonds with debenture stock. Finally, there is the precise example used earlier as an illustration of the description mechanisms of TAXMAN: the fact that "$58,854,200 par value of the common stock of the [Delaware] company . . . [is] immediately distributed among the common stockholders of the [New Jersey] company as a dividend, paying them two shares of the [Delaware] stock for each share they held in the [New Jersey] company."

[97] There is a list of 14 higher-level event descriptions, and there are approximately 50 internally generated symbols, all with the mnemonic prefix PHE, which represent the new classes of stocks and bonds, the ownership interests that are transferred, etc.

and then transferred to New Jersey. Curiously enough, the system has succeeded in matching a portion of the B-REORGANIZATION pattern to the description of the *Phellis* case: it has found an "acquisition" of Delaware stock by New Jersey, which is followed by a situation in which New Jersey "controls" Delaware. But PHE26 then turns out to represent the assets of Delaware, and thus in checking the "solely for voting stock" requirement of the Type B reorganization, the system discovers that the transfer is not even made in exchange for "stock," let alone "voting stock." Hence the search to match the full pattern of the B-REORGANIZATION concept ultimately fails, as it should.

Suppose we now query the system with (GOAL (C-REORGANIZATION ?A ?C ?T) ABSTRACT).[98] When the C-REORGANIZATION pattern is matched to the description of the *Phellis* case, the system finds the acquisition structure

```
(EXCHANGE
    ((TRANS NEW-JERSEY PHE26 NEW-JERSEY DEL-
      AWARE PHE28))
    ((TRANS DELAWARE PHE29 DELAWARE NEW-
      JERSEY PHE30)
     (TRANS DELAWARE PHE31 DELAWARE NEW-
      JERSEY PHE32)))
```

and proceeds to apply the additional constraints of the statutory definition. It checks first to determine that PHE26 is "substantially all of the properties" of the New Jersey corporation, which it is. Then it examines what is exchanged for these properties: PHE29, which turns out to be the Delaware debenture stock, and PHE31, which turns out to be the Delaware common stock. If both of these are voting stocks,[99] the C-REORGANIZATION program succeeds, and returns: (C-REORGANIZATION DELAWARE NEW-JERSEY PHE28).

The concept of a Type D reorganization [100] cannot presently

---

[98] A Type C reorganization is defined as "the acquisition by one corporation, in exchange solely for all or part of its voting stock (or in exchange solely for all or part of the voting stock of a corporation which is in control of the acquiring corporation), of substantially all of the properties of another corporation . . . ." I.R.C. § 368(a)(1)(C).

[99] Although the report of *Phellis* contains no mention of the voting rights of these securities, it is quite likely that the debenture stock is nonvoting, thus preventing the transaction from qualifying as a Type C reorganization under the current statute. For purposes of the illustration, however, I have stipulated that both of these stocks are voting stocks. A more realistic approach would require the expansion of the concepts of "common stock" and "debenture stock" to a more concrete level, *see* pp. 883–84 & note 107 *infra*.

[100] The definition is "a transfer by a corporation of all or a part of its assets to

be fully implemented in the TAXMAN system, since there are no facilities for representing the distributions that would qualify under sections 354, 355, or 356.[101] Nevertheless, we can implement the structural core of the remainder of the concept: the transfer of property to a corporation which the transferor then controls. When this core concept is matched against the description of the *Phellis* case, the system finds:

((TRANS NEW-JERSEY PHE26 NEW-JERSEY DELAWARE PHE28))

and

((CONTROL NEW-JERSEY DELAWARE PHE32)
(CONTROL NEW-JERSEY DELAWARE PHE33)
(CONTROL NEW-JERSEY DELAWARE PHE36))

Restricted to this partial fragment of the concept, then, the system determines that a Type D reorganization has occurred, and returns: (D-REORGANIZATION NEW-JERSEY DELAWARE PHE28).

This example illustrates the fact that the various reorganization patterns can be matched up to a case description network in multiple ways.[102] The existence of these partial matches and multiple matches suggests some of the ways in which the concepts of corporate reorganization law can be manipulated in the planning of a corporate transaction, and some of the ways in which ambiguities can arise in the analysis of a corporate reorganization case. It is potentially a very powerful feature of the TAXMAN system that it can identify and display these partial and multiple matches automatically.

V. Feasible Extensions

One of the main purposes behind the work on TAXMAN was the expectation that some of the techniques developed here could eventually prove useful in automating the more mundane

another corporation if immediately after the transfer the transferor, or one or more of its shareholders (including persons who were shareholders immediately before the transfer), or any combination thereof, is in control of the corporation to which the assets are transferred; but only if, in pursuance of the plan, stock or securities of the corporation to which the assets are transferred are distributed in a transaction which qualifies under section 354, 355, or 356." I.R.C. § 368(a)(1)(D).

[101] *See* p. 845 & notes 26–30 *supra.*

[102] The same three TRANS propositions appear in both the C-REORGANIZATION pattern and the partial B-REORGANIZATION pattern, but with their order within the EXCHANGE structure reversed in the two situations. In the partial B-REORGANIZATION pattern, the CONTROL structure appears in association with the acquisition of stock; whereas in the D-REORGANIZATION pattern, the same CONTROL structure appears in association with the transfer of assets.

aspects of legal research and analysis. Clearly the TAXMAN system in its present form does not yet provide a very useful tool for the practicing tax lawyer. The coverage of the system is much too limited, and even with a broader coverage of the tax law, the necessity of writing state descriptions and event descriptions in sequential order, and doing so in an obscure programming language, would render the system inaccessible to most potential users. Nevertheless, viewed as a simple pilot program rather than a full-scale applications program, the TAXMAN system does suggest a number of extensions which should eventually prove to be of practical significance. In this Part, I will describe some of these extensions and sketch briefly the directions in which future work in this area might proceed. Then, assuming that these extensions are feasible, I will outline how a more sophisticated version of the TAXMAN program might function within a system for computer-aided legal research, and compare such a system with the legal research systems on the market today.

The coverage of the system could be extended in many instances simply by a straightforward application of the basic techniques analyzed throughout this Article. One important area for further work would be the treatment of distributions and exchanges pursuant to a statutory reorganization. The major defect of the D-REORGANIZATION concept discussed in Part IV resulted from the inability of the current TAXMAN system to analyze the distributions that would qualify under sections 354, 355, or 356. An extended version of the system should be able to represent at least part of these nonrecognition rules [103] and at

[103] *See* note 26 *supra.* The basic structure of the nonrecognition rule could be represented, certainly, since it has some similarities to the basic conceptual structures we have already examined in Part III, *supra.* For example, the general rule of § 354(a)(1) reads: "No gain or loss shall be recognized if stock or securities in a corporation a party to a reorganization are, in pursuance of the plan of reorganization, exchanged solely for stock or securities in such corporation or in another corporation a party to the reorganization." Here, the core of the concept is a simple exchange structure, which would cause very few difficulties for the TAXMAN system, and the more important problems would then arise in representing the concepts that are attached to this conceptual core. *See generally* BITTKER, *supra* note 18, ¶¶ 14.30–.34. The definition of a "party to the reorganization" in § 368(b) would not be too difficult to implement, since it is built up out of a number of definite descriptions extracted from the components of the reorganization concept itself. *See* BITTKER, *supra* note 18, ¶ 14.32(1); Treas. Reg. § 1.368–2(f) (1976). *But see* Groman v. Commissioner, 302 U.S. 82 (1937); Helvering v. Bashford, 302 U.S. 454 (1938). The concepts of "stock" and "securities" are not defined in the statute, but it is possible to articulate some of the rules governing the application of these concepts, as these have emerged from the cases. *See* BITTKER, *supra* note 18, ¶ 14.31; Treas. Reg. § 1.354–1(e) (1960); cases cited note 32 *supra*; Neville Coke & Chem. Co. v. Commissioner, 148 F.2d 599 (3d Cir.), *cert. denied*, 326

least part of the associated basis rules.[104] More ambitiously, an extended version of the system should be able to represent some of the rules for distributions generally, even outside the reorganization context.[105] An extension of the analysis mechanisms in these directions would require a comparable extension of the description mechanisms: the system would have to expand the concepts of stocks and bonds, as suggested earlier,[106] and provide a formalism for describing various hybrid securities; [107] it would

---

U.S. 726 (1945); Carlberg v. United States, 281 F.2d 507 (8th Cir. 1960). Not all the components of the general nonrecognition rule could be handled in a straightforward fashion, however. The concept of an exchange being made "in pursuance of the plan of reorganization" is not well defined at all, and it is further complicated by its involvement in the problems of "creeping acquisitions," *see* pp. 873–76 & notes 87–88 *supra*, "business purpose," *see* pp. 843–48 *supra*, and "step transactions," *see* pp. 848–49 *supra*. *See* BITTKER, *supra* note 25, ¶¶ 14.32(1), .50, .51; Manning, *"In Pursuance of the Plan of Reorganization": The Scope of the Reorganization Provisions of the Internal Revenue Code*, 72 HARV. L. REV. 881 (1959); Treas. Reg. §§ 1.368–1(c) (1960), –2(g) (1976), –3(a) (1962).

[104] *See* p. 844 *supra*. The general rule of § 358(a) is quite mechanical: "the basis of the [nonrecognition] property . . . received [is] the same as that of the property exchanged, decreased by the fair market value of any other property [and] any money received by the taxpayer, . . . and increased by . . . the amount of gain to the taxpayer which was recognized on such exchange . . . ." Thus, assuming that that nonrecognition rules could be represented well enough to determine "the amount of gain . . . recognized," the representation of the basis rules would be a simple matter. Further rules prescribing the allocation of basis when several different stocks and securities are exchanged could also be represented quite easily. *See* Treas. Reg. § 1.358–2 (1960).

[105] *See* pp. 843–44 *supra*. The rules for corporate distributions have some of the same characteristics that were observed in note 103 *supra*: some, employing relatively precise tests, could be handled in the TAXMAN system in a straightforward fashion, but others, involving vague or ill-defined concepts, might require a fundamental modification of the TAXMAN paradigm. For example, the rules for identifying "stock redemptions" include the simple mechanical tests of a "substantially disproportionate" distribution, § 302(b)(2), as well as the vague standard of a redemption "not essentially equivalent to a dividend," § 302(b)(1). The new stock dividend rules of § 305 specify certain distributions which are taxable, § 305(b)(3)–(5), but also require an inquiry into whether or not a distribution "has the result of the receipt of property by some shareholders, and an increase in the proportionate interests of other shareholders," § 305(b)(2). The general dividend rule, § 316(a), requires a determination of the corporation's "earnings and profits," which can be computed in most instances from the complex rule of § 312 and Treas. Reg. § 1.312 (1960), but which is nowhere given a comprehensive statutory definition. *See generally* BITTKER, *supra* note 18, ¶¶ 7.01–.04, 7.60–.63, 9.20–.24.

[106] *See* p. 861 *supra*.

[107] This would be necessary, for example, to implement the rules governing the exchange of "stock or securities" under § 354(a), *see* note 103 *supra*, and to implement the rules governing "stock dividends" in § 305, *see* note 105 *supra*. The general approach here would be to develop descriptions of the possible components of a security interest: the voting rights, the priorities upon liquidation, the dividend provisions, etc. These components could then be combined in various combinations

also have to provide a better formalism for corporate accounting[108] and for some of the economic consequences of the transactions.[109] But once this expanded factual network is developed, a good many of the rules and concepts governing corporate distributions generally, and reorganization exchanges specifically, could be represented in abstraction theorems not dissimilar to those of the current version of TAXMAN.[110] These extensions would not be trivial, of course, but they would not seem to require any major conceptual breakthroughs.[111]

A number of techniques are also available to overcome the rigidities of the current system. I have mentioned some of the difficulties that arise from the requirement that event descriptions be written out in a strict sequential order.[112] An extended version of the TAXMAN system should have built into it a more sophisticated model of temporal relationships and a method of manipulating these relationships by means of higher-level descriptions.[113] More generally, an extended version of TAXMAN

---

to produce, *e.g.*, "convertible subordinated debentures," § 279(b), "hermaphrodite convertible preferred stock," Rev. Rul. 70-108, 1970-1 C.B. 78, or whatever.

[108] This would be necessary to implement any of the rules concerning "earnings and profits" of §§ 312 and 316(a), *see* note 105 *supra*, and in a number of other contexts.

[109] A very simple economic model would suffice to represent the concept of "gain" or "loss" and to implement the basis rules of § 358, *see* note 104 supra. More complex models would be required to represent the economic differences between the ownership of various classes of securities, as in note 107 *supra*. Finally, for a fuller understanding of the "purposes" and "motivations" behind the various reorganization patterns, and an understanding of the divergences between "form" and "substance," it would be necessary for the TAXMAN system to have available a means of describing both the tax and the nontax benefits to the participants in any corporate transaction. The development of these economic models would not be a simple task, but once developed they would play an important role in an extended version of TAXMAN.

[110] The proposals in notes 103-105 *supra* would, in effect, treat the DISTRIBUTE concept as an analysis mechanism and implement it as an abstraction theorem; whereas the previous discussion of corporate distributions, pp. 868-70 *supra*, treated the DISTRIBUTE concept as a description mechanism and implemented it as an expansion theorem.

[111] There are two caveats to this statement. The first has already been indicated in notes 103-105 *supra*: some of the concepts and rules of corporate reorganization law may elude the current methods of semantic information processing. The second caveat is that although it may be possible in principle to describe the factual situations and conceptual structures that occur in this area of the law, the programming effort could conceivably become bogged down in the complexity of the details. My own judgment, however, is that a large portion of corporate reorganization law could be incorporated into an extended version of TAXMAN before either of these limits is reached.

[112] *See* note 92 *supra*.

[113] The current version of TAXMAN represents "time" as a simple linear list. The natural generalization would use an unordered set of "time points," which could

should permit a more flexible use of description and analysis mechanisms of all kinds.[114] A still more radical variation would shift the operation of the entire system from an "analysis" mode to a "planning" mode. The current system is designed to produce a legal analysis of a given sequence of events. In a planning mode, by contrast, the system would construct a sequence of events so as to satisfy a given legal analysis.[115] Although this is not a trivial problem, it is one that has been solved in other contexts.[116] In

then be given a partial ordering by relationships such as "before" and "after"; and a corresponding set of "time intervals," which could be connected to each other by expressions such as "*A* occurs during *B*," and "*B* overlaps with *C*." It should be possible in this system to express incomplete knowledge, or even to express inconsistent knowledge, about these temporal relationships. For an example, see Bruce, *A Model for Temporal References and Its Application in a Question Answering Program*, 3 ARTIFICIAL INTELLIGENCE 1 (1975).

[114] One rigidity of the current system is the requirement that a case be described in full, and then expanded in full, before any analysis mechanisms can be applied. A more flexible system would accept a partial description, produce only a partial expansion, and attempt with this to do a partial analysis. The analysis mechanisms would then generate, if necessary, an additional expansion, or request from the user an additional piece of the description.

The problem here could be characterized as a *store vs. recompute* decision: Is it preferable to expand all factual descriptions at once, and thus "store" in computer memory everything that can be said about a given situation? Or is it preferable to leave the descriptions in an unexpanded form, as they are received, and then "recompute" the lower-level information every time an analysis mechanism actually needs it? Any response to this issue, however, requires a response to a large number of other issues about the efficient representation of conceptual information in computer data structures, and this leads quickly to most of the currently active areas of research in semantic information processing. For a discussion of the "design issues" that must be faced in constructing any representation of conceptual knowledge, see Moore & Newell, *How Can MERLIN Understand?*, in KNOWLEDGE AND COGNITION 201 (L. Gregg ed. 1974); Bobrow, *Dimensions of Representation*, in REPRESENTATION AND UNDERSTANDING 1 (D. Bobrow & A. Collins eds. 1975). Some of these issues have been briefly mentioned earlier in this Article. *See, e.g.*, the discussion of control structures, note 62 *supra*; abstraction theorems and procedural representations, note 63 *supra*; expansion theorems, note 65 *supra*; and references cited therein.

[115] These two models are of course not entirely distinct, and each can appear as a component of the other. Obviously the planning mode requires the use of the analysis mode to determine the legal consequences of various alternative plans, but the analysis mode may often require the use of the planning mode as well. In the corporate tax area, for example, the arguments concerning the proper legal characterization of a given transaction may be strongly influenced by a consideration of the alternative ways in which the transaction could have been carried out. This can be seen in many of the cases involving the "business purpose" doctrine, *see* pp. 847–48 *supra*, and the "step transaction" doctrine, *see* pp. 848–49 *supra*.

[116] A close analogy can be seen in the problem of chemical synthesis, where computer programs have been written to construct a sequence of chemical reactions from commercially available compounds that will produce a desired organic molecule, such as vitamin A. *See* Sridharan, *Search Strategies for the Task of Organic Chemical Synthesis*, PROCEEDINGS OF THE 3RD INT'L JOINT CONF. ON ARTIFICIAL

general, if the conceptual structures in a problem domain are reasonably well understood and reasonably well defined, it is a manageable task to devise a computer program to search heuristically for an appropriate "plan." [117]

Finally, let us look briefly at the problem of natural language understanding. The important point here is that the type of programming language in which TAXMAN is written is intended to function as part of a larger system for natural language processing, and thus it would be possible, though not at all easy, to write programs that could translate automatically from the description and analysis mechanisms of TAXMAN into a simplified form of English, and vice versa. The best example of a program of this sort is found in the work of Winograd, cited earlier.[118] Winograd's system conducts a dialogue in English about a simple world of blocks — small yellow cubes, large red boxes, tall green pyramids, etc. — which can be moved around and stacked on a table top. The system has three main components which interact with each other: a component for *inference*, which maintains the basic model of the block world, and is written in the same language as TAXMAN; [119] a component for *syntax*, which contains a basic grammar of English; [120] and

---

INTELLIGENCE 95 (1973). Other examples include constructing proofs of simple mathematical theorems, *see, e.g.*, Bledsoe, Boyer, & Henneman, *supra* note 15, and building towers and other structures in a world of children's blocks, *see, e.g.*, Fahlman, *A Planning System for Robot Construction Tasks*, 5 ARTIFICIAL INTELLIGENCE 1, 26–29 (1974).

[117] In TAXMAN, for example, it should be possible to reproduce some of the formal manipulations that were seen in Helvering v. Elkhorn Coal Co., 95 F.2d 732 (4th Cir.), *cert. denied*, 305 U.S. 605 (1938); *see* pp. 848–49 *supra*. Given the initial state description in that case, and a description of the desired end result, an extended version of the TAXMAN system should be able to notice that the Type C reorganization almost works, except for the problem of "substantially all properties," and then with this information generate the sequence of Type D, Type C, and Type B reorganizations that would be, formally, tax-free. This proposal for generating a "plan" would, in effect, treat the REORGANIZATION concept as a description mechanism, and implement it as an expansion theorem; whereas the previous discussion of corporate reorganization, *see* pp. 872–76 *supra*, treated the REORGANIZATION concept as an analysis mechanism, and implemented it as an abstraction theorem. The planning mode can thus be viewed as a process of positing, hypothetically, a higher-level description that we would like to have applied to our case, and then generating the lower-level expansions which would make that hypothetical description come true, consistent with the other assertions already present in the data base. On the dual representation of such concepts, compare note 110 *supra*.

[118] *See* notes 12, 63, 66–68, 80 *supra*.

[119] Winograd, *supra* note 12, at 23–27, 108–26.

[120] *Id.* at 16–23, 46–108. The syntactic component is based on Halliday, *Notes on Transitivity and Theme in English*, 3 J. LINGUISTICS 37 (1967), 4 J. LINGUISTICS 179 (1968); Halliday, *Functional Diversity in Language as Seen from a Considera-*

a component for *semantics*, which mediates between the other two components.[121] Winograd's system operates by translating an English sentence, usually a question or a command, into the internal representation of the block world model, generating an appropriate response within this model, and then translating this response back into English.[122] In the long run, a program of this

tion of Modality and Mood in English, 6 FOUNDATIONS OF LANGUAGE 322 (1970).

[121] Winograd, *supra* note 12, at 28–34, 126–69. Winograd provides the following introductory account of the role of the semantic component in his system: "There has never been a clear definition of what the field of 'semantics' should cover, but attempts to program computers to understand natural language have clarified what a semantic theory has to do, and how it must connect with the syntactic and logical aspects of language. In practical terms, we need a transducer that can work with a syntactic analyzer, and produce data which is acceptable to a logical deductive system. Given a syntactic parser with a grammar of English, and a deductive system with a base of knowledge about particular subjects, the role of semantics is to fill the gap between them." *Id.* at 28. In many ways the semantic component is the key to the success of Winograd's system, and the reader is advised to consult the complete work for an understanding of how it operates.

[122] The following dialogue is typical. The system has been requested earlier to "stack up both of the red blocks and either a green cube or a pyramid," and it has done so, in the process removing a small pyramid from the top of a red cube so that it could use that cube to complete the stack as specified. The following exchange then occurs (with queries to the system given in lower case, and responses in upper case):

> Had you touched any pyramid before you put the green one on the little cube?
>
> YES, THE GREEN ONE.
>
> When did you pick it up?
>
> WHILE I WAS STACKING UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.
>
> Why?
>
> TO GET RID OF IT.
>
> Why did you do that?
>
> TO CLEAR OFF THE RED CUBE.
>
> Why did you clear off that cube?
>
> TO PUT IT ON A LARGE GREEN CUBE.
>
> Why did you do that?
>
> TO STACK UP THE RED CUBE, A LARGE RED BLOCK AND A LARGE GREEN CUBE.
>
> Why did you do that?
>
> BECAUSE YOU ASKED ME TO.
>
> How did you do it?
>
> BY PUTTING A LARGE RED BLOCK ON THE TABLE; THEN LETTING GO OF IT; THEN PUTTING A LARGE GREEN CUBE ON IT; THEN LETTING GO OF THAT CUBE; THEN PUTTING THE RED CUBE ON THAT CUBE; THEN LETTING GO OF THAT CUBE.
>
> How many objects did you touch while you were doing it?
>
> FOUR OF THEM.
>
> What did the red cube support before you started to clean it off?
>
> THE GREEN PYRAMID.
>
> There were five blocks to the left of the box then.

sort could be developed for use with the TAXMAN system; but in the short run, the attempt would probably by stymied by the problem of complexity. Even in its very restricted current version, the TAXMAN system is more complex than the block world model, mainly because of the complexity of the higher-level concepts of "distributions" and "reorganizations." A future version of TAXMAN, extended in coverage and flexibility as suggested above, would greatly exceed the capacity of the current programs for the syntactic and semantic components of natural language. I consider it more important, then, to develop the other feasible extensions first, and to restrict communications with the computer for the time being to a high-level programming language, as in the current system. It should later be possible to return successfully to the design of a natural language processor for corporate reorganization law.

How might the extended version of TAXMAN described in the preceding paragraphs function within a system for computer-aided legal research? The basic idea would be to develop and maintain a data base consisting of (1) a large number of factual descriptions, taken from reported cases, revenue rulings, treasury regulation hypothticals, etc.; (2) a wide variety of legal concepts and legal rules, taken from the Code, the Regulations, and other sources; and (3) for each factual description in the data base, the set of all possible legal characterizations of these facts in terms of the stored concepts and rules. If the system is sophisticated enough, it should be possible to store and analyze each factual description in the same way that the case of *United States v. Phellis* was stored and analyzed, but with greater descriptive flexibility, and with some form of natural language input. The system could then be used in several ways. It could be used simply as a device to retrieve the factual situations which match certain patterns of interest, or which satisfy certain fragments of a legal concept or a legal rule. It could be used to develop a suggested analysis of a new case: the user could describe the new case to the system and then generate a range of applicable higher-level concepts. Finally, the system could be used in a planning mode: the user could describe an initial situation and a desired end result and then generate a number of possible transaction patterns with the desired tax consequences.

It may be instructive to compare this imagined system with the systems for computer-aided legal research and analysis which are currently available. First, there are a number of systems in

---

NO, ONLY FOUR OF THEM: THE RED CUBE, TWO LARGE GREEN CUBES AND A LARGE RED BLOCK. . . .

Id. at 13–14.

operation today which maintain a data base of statutes and decided cases and retrieve this information in response to an appropriate query.[123] But the form in which these materials are stored, and the techniques for retrieval, are quite different from those of the proposed extension of TAXMAN. The existing retrieval systems generally store the full text of a legal document and retrieve it by key-words, or combinations of key-words, which in its most primitive form amounts simply to requesting, *e.g.*, "all cases in which the word 'jurisdiction' appears." [124] Although this type of retrieval can be very useful in many circumstances, it is confined to looking only at what might be called the "surface statistics" of legal language. By contrast, the extended version of TAXMAN would store a representation of the underlying conceptual structure of a statute or a case and retrieve this structure by a sophisticated pattern-matching operation.[125]

---

[123] The two most prominent systems in this country are LEXIS, developed by Mead Data Central, Inc., and WESTLAW, a service recently offered by the West Publishing Co. Other important legal retrieval systems include LTE, an early system developed by the Air Force; JURIS, developed by the Justice Department; QUIC/LAW, developed at Queen's University in Canada; DATUM, developed at the University of Montreal; UNIDATA in Switzerland; CREDOC in Belgium; and CRIDON in France. *See generally,* J. SPROWL, A MANUAL FOR COMPUTER-ASSISTED LEGAL RESEARCH (Am. Bar Foundation, 1976); AMERICAN BAR ASS'N, AUTOMATED LAW RESEARCH (1973); C. TAPPER, COMPUTERS AND THE LAW (1973).

[124] A typical example of the retrieval process in the LEXIS system is given in Harrington, *What's Happening in Computer-Assisted Legal Research?*, 60 A.B.A.J. 924, 930–31 (1974). Harrington poses the problem of finding a case in the United States Supreme Court which addresses the following question: "If the United States appears in a probate court to assert a claim against a decedent's estate, does it thereby waive its immunity against being sued in a state court so that the fiduciary can assert a counterclaim greater than the amount of the original claim asserted by the United States?" The first request to the LEXIS system is to search for all cases in which the phrase *United States* occurs within ten words of either the word *counterclaim* or the word *crossclaim.* The computer locates 25 cases which satisfy this request, and the researcher specifies further that the case should also contain the phrase *state court.* This time the computer locates 19 cases, and the researcher specifies further that the word *jurisdiction* be found within seven words of the phrase *state court.* The computer locates only one case which satisfies this last request and the researcher displays it on the screen. It is United States v. Shaw, 309 U.S. 495 (1940), which happens to be precisely on point. The researcher can then "Shepardize" *Shaw* automatically by searching for all subsequent cases in which its citation appears. The use of the West Publishing Co. WESTLAW system is similar, except that the data base consists of West headnotes rather than the full text of the opinions. *See* J. SPROWL, *supra* note 123, at 55–78.

[125] As a simple illustration of the difference between these two approaches, suppose we wish to locate a case in which there occurs a single transaction fitting the general pattern of a Type C and a Type D reorganization, and fitting the pattern of a Type B reorganization except for the "solely for voting stock" requirement. In an extended version of TAXMAN, this specification could easily be expressed, and the system would retrieve *United States v. Phellis*, among others.

It would therefore be substantially more powerful and flexible than the systems for legal information retrieval on the market today.[126]

The second approach to computer-aided legal research today is based upon the methods of computer-assisted instruction (CAI). The computer poses questions designed to elicit the essential facts of a case, and then, depending on the information received, either suggests a tentative legal analysis or poses an additional factual question.[127] There are parallels here, of course, to the way an extended version of TAXMAN might develop the description and analysis of a case. But note the rigid structure of these CAI-based systems: the user faces a series of pre-

---

But this search request could not even be formulated in the existing legal retrieval systems, nor is it likely that any other request could be formulated which could retrieve with precision the factual pattern of *Phellis*. The difficulty in locating *Phellis* itself is that the statutory reorganizaion provisions were inapplicable to it, and the opinion contains no mention of them. Even confining the search to cases that postdate the reorganization statutes, the existing retrieval systems would still encounter difficulties. The most efficient way to search for a Type C reorganization, for example, is to search for the statutory section number "368(a)(1)(C)," and its predecessor in the 1939 Code, "112(g)(1)(C)." Thus a researcher might request a search for 368(a)(1)(B) AND 368(a)(1)(C) AND 368(a)(1)(D), and then a search for 112(g)(1)(B) AND 112(g)(1)(C) AND 112(g)(1)(D). But the system would then retrieve *any* case in which these three statutory sections were mentioned, whether applicable or not, and *no* case which failed to discuss all three; there would be no way to specify to the system that the three reorganization patterns should fit the same basic transaction, or that the Type B reorganization pattern should only be partially applicable. For example, had it been decided under the 1939 or the 1954 statute, one case that would be retrieved with this request is Helvering v. Elkhorn Coal Co., 95 F.2d 732 (4th Cir. 1938), *see* pp. 848–49 *supra*, where the three reorganization patterns apply to sequential transactions, rather than a single transaction. Similar illustrations could be constructed for most other kinds of retrieval problems in the corporate tax area.

[126] A similar conclusion is advanced by P. SLAYTON, ELECTRONIC LEGAL RETRIEVAL (Report prepared for Canadian Dep't of Communications, 1974). *See also* P. Slayton, Radical Computer Use in Law (Draft Report prepared for the Canadian Dep't of Communications, June 1974).

[127] This approach has not yet been extensively developed as an aid to practicing attorneys, but one experimental project has produced a client interviewing system for use by the Cook County Legal Assistance Foundation in Chicago. *See* Chatterton & McCoy, *Computer-Assisted Legal Services*, LAW & COMPUTER TECH., Nov. 1968, at 2; *Discovery of Jurimetrics Projects*, 15 JURIMETRICS J. 56 (1974). The CAI approach has been used more extensively as an educational tool, with the computer posing questions and hypothetical cases to a law student and evaluating the responses. *See, e.g.*, Maggs & Morgan, *Computer-Based Legal Education at the University of Illinois: A Report of Two Years' Experience*, 27 J. LEGAL ED. 138 (1975). For a thorough survey of computer-assisted instruction generally, see R. LEVIEN, THE EMERGING TECHNOLOGY: INSTRUCTIONAL USES OF THE COMPUTER IN HIGHER EDUCATION (1972); for a sharp critique, see A. OETTINGER & S. MARKS, RUN, COMPUTER, RUN: THE MYTHOLOGY OF EDUCATIONAL INNOVATION (1969).

programmed questions and usually responds in a strict multiple-choice format; the author of the program faces the tedious task of anticipating all possible responses and programming explicitly every logical branch. In the extended version of TAXMAN, by contrast, the most important and difficult work is done at the start, when the description and analysis mechanisms are designed to capture the basic conceptual structure of the problem domain. Once these basic conceptual structures are written, the "analysis" and "planning" modes of the system can then be programmed more easily, in a uniform and systematic way. For the user of the system this permits a much greater flexibility in the interactive process, and for the author of the program it requires much less concern about the specific details of the interaction. In general, then, the TAXMAN approach appears to have much greater long-range potential than the CAI approach as a technique for computer-aided legal research, either for the law student or for the practicing attorney.[128]

But the important question remains: Just how feasible is this extended version of TAXMAN? The answer depends both on the characteristics of the problem domain in which we wish to have the extended system operate and on the expected advances in semantic information processing in the next few years. In order for the current paradigm to be applicable, as we have seen, it is necessary to develop an adequate model of the lower-level factual situations that we expect to occur[129] and an adequate representation of the higher-level legal concepts in the relevant area of the law.[130] The taxation of corporate reorganizations is an excellent problem domain in both of these respects. It appears

[128] The position taken here is similar to that of Carbonell and his associates, who have done the pioneering work on the application of artificial intelligence techniques to computer-assisted instruction. *See, e.g.*, Carbonell, *AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction*, 11 IEEE TRANSACTIONS ON MAN-MACHINE SYSTEMS 190 (1970); Brown & Burton, *Multiple Representations of Knowledge for Tutorial Reasoning*, in REPRESENTATION AND UNDERSTANDING 311 (D. Bobrow & A. Collins eds. 1975). Carbonell's views on the relative merits of the traditional CAI systems and the more experimental AI systems are well-balanced: " . . . we are not advocating the complete elimination of [traditional CAI]. It will have its role for some time to come. We see it convenient for cases in which the subject matter is very diversified and the interactions with the students are planned to be brief. In those cases, the development of complex semantic networks is not justified. When discussion in depth is desired, when the student should have some initiative, when detailed anticipation is unwanted, then [CAI] systems are to be preferred. On the other hand, when teaching sequences are extremely simple, perhaps trivial, one should consider doing away with the computer, and using other devices or techniques more related to the task." Carbonell, *supra*, at 201.

[129] *See* pp. 883–84 & notes 106–109 *supra*.

[130] *See* pp. 882–83 & notes 103–105 *supra*.

that a large number of important and interesting problems can be handled here well before we reach the apparent limits of the TAXMAN paradigm.[131] It is a risky business predicting future advances in computer science, but based upon a knowledge of both the law of corporate reorganizations and the current work in semantic information processing, I would be willing to hazard a guess: if a substantial amount of resources was committed to the project, a sophisticated prototype system of demonstrable utility to a corporate tax lawyer could be developed within approximately ten years.

VI. Conclusion

In this Article I have shown how the current paradigm of semantic information processing can be applied to a particular area of the law, and I have indicated how this approach could lead to several practical applications. But the theoretical purpose behind the work on TAXMAN has not yet been addressed. I suggested that the work should provide a tool for the development of our theories about legal reasoning, enabling us to acquire a more precise understanding of the structure and dynamics of legal concepts. The suggestion was that we view the TAXMAN system as a formal model of the concepts of corporate reorganization law and assess its adequacies and inadequacies in this respect. For this purpose, however, it should be clear by now that there are serious deficiencies in even the extended version of TAXMAN. The most important deficiencies appeared in Part I, where it was emphasized that the "continuity of interest," "business purpose," and "step transaction" doctrines were developed in opposition to the mechanical manipulation of abstract statutory rules, and that they seemed to have a different structure from the concepts that have so far been successfully implemented in TAXMAN.[132] A possible explanation of these deficiencies was suggested at the close of Part II, where it was noted that the TAXMAN system was statically equivalent to the higher-order predicate calculus, even though the programming language in which it was written added to this static representation a certain dynamic power and flexibility.[133] In general, then, the detailed analysis of the TAXMAN system has tended to support what

[131] *See* note 111 *supra.*

[132] *See* pp. 846–50 *supra.* A similar observation occurred in the discussion of the feasibile extensions in Part V: although it appeared that a great many of the rules and concepts of corporate reorganization law could be added to an extended version of TAXMAN, this claim could not be made for all such rules and concepts. *See* pp. 882–84 & notes 103–11 *supra.*

[133] *See* pp. 861–62 *supra. See also* note 103 *supra.*

should surely be a lawyer's intuition: that the current TAXMAN paradigm fails to capture many of the significant facts about the structure of legal concepts and the process of legal reasoning.

I do not believe, however, that this is the final word on the subject. A systematic exploration of the limitations of the current TAXMAN paradigm would, I believe, lead us to a modified paradigm which would correct some of these limitations and permit us to say something about the structure and dynamics of even the more vaguely defined concepts of corporate reorganization law.[134] Although we would ultimately come to the conclusion, not unlike the lawyer's intuition, that nothing as complex as legal reasoning could ever be represented in a computer program, I believe it would be possible to sketch out a formal computer model somewhat more realistic than the current version of TAXMAN. And this, after all, is the task of greatest theoretical interest. Even if our formalisms will always be inadequate in one or more respects, the process of constructing and modifying these formalisms, if carefully done, should itself be a source of insight and understanding.

[134] I will pursue this analysis in detail in a future article.