ISBN 0 86076 005 7

*Printed by the Gomer Press, Llandysul, Dyfed*



# COMPUTER SCIENCE AND THE LAW

*Inaugural Lecture of the Professor of Computer Science delivered at the College on January 25th, 1977*

*by*

PROFESSOR BRYAN NIBLETT
M.Sc., Ph.D., Barrister

UNIVERSITY COLLEGE OF SWANSEA

UNIVERSITY COLLEGE OF SWANSEA

# COMPUTER SCIENCE AND THE LAW

*Inaugural lecture of the Professor of Computer Science delivered at the College on January 25, 1977*

*by*

PROFESSOR BRYAN NIBLETT
M.Sc., Ph.D., Barrister

Swansea
1977

First published March 1978 by the University College of Swansea.
Obtainable from The Publications Office, University College of Swansea, Singleton Park, Swansea SA2 8PP.



ISBN 0 86076 005 7

*Printed by the Gomer Press, Llandysul, Dyfed*



# COMPUTER SCIENCE AND THE LAW

It has often been said that the occasion of a professor's inaugural lecture can be likened to a ceremony of marriage—a form of academic espousal—in which the new professor is symbolically wedded to his university. If this be so then since the first of September 1976 the University of Wales and I have been living together in sin. This illicit relationship has given me a great deal of pleasure and I look forward to a continuing enjoyable and fruitful alliance with the College after this evening when by way of this lecture I make so to speak an honest woman of the University.

The theme of the lecture—computer science and the law—is a large one. The law touches our lives and activities at innumerable points, and the science of computers is coming to have an equally pervasive influence. So it should not be surprising that the conjunction of the two disciplines, the intersection of computer science and the law, is a topic of substantial dimensions.

This evening I propose to consider one limited aspect of the subject: ways in which the modern digital computer can be used to store, to search, to analyse, to classify the corpus of the law. In particular I shall discuss methods of analysing statute law, Acts of Parliament, because of my own special interest in this area.

Let me begin by asking: what are the distinguishing characteristics of a computer that make it suitable for searching and analysing the law? Three particular intrinsic features of a modern computer make it ideal for this purpose. Firstly, computers can store vast quantities of textual material. Many people suppose that a computer's sole purpose is to handle numbers, but what it actually stores is data symbolically represented as binary digits—or bits—and these may as easily be words as numbers. Modern storage devices can provide ready access to as many as $10^{12}$ bits. This corresponds to some $10^{10}$ words which is approximately the size of all the cases published in United Kingdom law reports of the last fifty years or so. We have therefore reached the stage where the text of the law can without difficulty be stored in the memory cells of a computer.

The second feature of a computer which makes it of value for searching the law is that it can manipulate or process these words at speeds as high as a million words a second. By "process" here I mean that the computer can count the words, can compare them, can sort them and merge them, and can place them in any predefined order. Words are the tools of a lawyer's trade. A lawyer holds himself out as an expert in the use of words. Now he has a machine of incomparable power and precision to help him in this task.

Thirdly, a computer is a logic machine. A digital computer is constructed from logical circuits which are designed to perform logical operations. There is an analogy, not an exact one but a close one, between the forms of logic used in computer circuits and the forms of logic used by lawyers. By exploiting this analogy a computer can help to extract the meaning from a legal text.

Of course these characteristics of a computer make it of value for information retrieval generally not merely in legal applications. And so we have seen in the last few years computers being used to store and retrieve information about physics, about medicine and about a wide range of subjects. But there is a difference when information retrieval is applied to legal sources. Legal information systems offer the possibility of retrieving the law *itself* rather than merely information *about* the law. This is true insofar as the law consists, as it largely does, of words in documents. So there is a directness and immediacy in retrieval systems designed for legal sources that is not always present in other applications.

As a result the last fifteen years has seen a growing interest throughout the world in legal information retrieval by computer. In the USA from a host of projects two major commercial systems have emerged for lawyers in private and public practice: the LEXIS system operated by the Mead Data Corporation and the WESTLAW system offered by the great legal publishing house of the West Company. In Europe major systems have also been developed though in contrast to the USA they are operated largely on behalf of government rather than private organisations. In the United Kingdom the



STATUS system (for STATUte Searching) has evolved under the auspices of the UK Atomic Energy Authority and Queen's University of Belfast has sponsored a system called QUOBIRD.

My personal interest in originating the STATUS project was to apply computer analysis to the storage and searching of statutory text. Acts of Parliament are a particularly interesting collection of documents for computer analysis because of the formal and consistent language in which they are drafted and because they directly embody legal norms. More than any other documents they mean what they say and say what they mean. The total number of public general Acts currently in force is about five thousand, their total length about twenty million words, and they date from as long ago as 1267. This body of material is an ideal corpus for a self-contained information system.

The first task in searching Acts of Parliament by computer is to convert them into machine-readable form, that is into a string of binary digits so that the computer can store and process them. During the last ten years I have experimented with all the various methods of data preparation, from punching the text on to Hollerith cards, using paper tape, using key-to-disk systems, to typing on-line direct into memory. I find the simplest and cheapest method is to transcribe the text on to plain sheets of white paper using a special typeface and then scan this with an optical character reader. Nowadays these problems are solved for us by the decision of the Queen's Printer in the shape of Her Majesty's Stationery Office to print the new edition of Statutes in Force using computer typesetting methods so that a machine-readable copy of the text becomes available as a by-product of the printing process. It is a simple matter to take these computer-typesetting tapes, strip them of the special symbols required for printing purposes and convert them to a suitable format for input to an information system. It is expected that the text of all the Statutes in Force will be available in machine-readable form by the early 1980s.

The next step is to prepare computer programs for indexing the Acts of Parliament, that is to say providing a pointer to the exact location of each word. My own belief is that in the special case of Acts of Parliament every word in the text, even the most common, should appear in this index. A judge once said in the course of a judgment that "It is a bold man who ignores a word in an Act of Parliament" and it would be an intrepid computer scientist who disregarded this *obiter dictum*. Nonetheless it is customary to allow the user to designate a list of words which may be excluded from the index.

Fig. 1 shows the thirty most common words in the Income and Corporation Taxes Act 1970 as originally enacted. The total number of words in the Act is 268,107 and the number of different words 4,050. The most frequently appearing word constitutes some 8% of the text, so that of the 20 million words or so in Statutes in Force some 1,600,000 of them are the word 'the'—a striking thought! The six most common words, 'the', 'of,' 'to', 'in', 'or', and 'a' represent 22% of the text. The unit which is used for searching purposes is the section of an Act of Parliament which on average is about 200 words. This is a convenient size: large enough for its meaning to be grasped and yet small enough to provide adequate discrimination.

*Figure 1*

INCOME AND CORPORATION TAXES ACT 1970

*List of Most Frequent Words*

| | | | | | |
|---|---|---|---|---|---|
| THE | 21827 | AS | 3703 | UNDER | 2259 |
| OF | 16955 | SHALL | 3614 | ON | 2245 |
| TO | 9003 | BY | 3400 | NOT | 1974 |
| IN | 8092 | THAT | 3361 | IF | 1620 |
| OR | 7973 | THIS | 2999 | AMOUNT | 1538 |
| A | 6073 | IS | 2945 | SUBSECTION | 1451 |
| ANY | 5076 | SECTION | 2729 | AN | 1449 |
| AND | 4840 | INCOME | 2698 | PERSON | 1434 |
| BE | 4576 | TAX | 2680 | ACT | 1425 |
| FOR | 4105 | WHICH | 2657 | COMPANY | 1406 |

Number of different words 4,050

Total number of words 268,107



An extract from an index to selected Acts of Parliament is given in Fig. 2 which lists the occurrences of the two words 'atomic' and 'energy'. It provides the exact address of each word, that is the number of the section, number of the sentence and the position of the word in the sentence. So the word 'atomic' appears first in document one, sentence one, word nine, and then in document two, sentence two, word twenty-seven and so on. The alphabetic character which follows each address is there to specify the nature of the text in which the word appears; whether for example it is a marginal note, or a long title or part of a schedule to an Act. In the particular sub-set of Acts of Parliament from which Fig. 2 is prepared the word 'atomic' appears sixty-nine times and the word 'energy' seventy-one times. It can be seen by inspection that on all but two occasions the word 'atomic' is followed immediately by the word 'energy' to form the phrase 'atomic energy'. Thus by arranging the text in this indexed form the presence or absence of combinations of words can quickly be established without traversing the text sequentially. This form of index, or concordance as it is commonly called, is a powerful tool for rapid searching of the text.

Of course concordances have been known for hundreds of years and were originally made of the text of the Bible. Preparation of the first concordance of the Bible in English was undertaken by John Marbeck who in 1550 produced his

> 'Concordance: that is to saie, a work wherein by the ordere of the letters of the A.B.C. ye maie redely find any word conteigned in the whole Bible so often as it is there expressed or mentioned'.

Marbeck's concordance of the Bible took him six years to compile. Using a large modern digital computer a concordance of the Bible can be produced in six minutes. This vast change of time-scale, from years to minutes, brought about by advances in computer science, enables the concordance to become a standard tool for all sorts of applications.

Another name for a concordance is 'inverted file' since it represents an inversion of the information matrix of the original text whereby rows become columns and vice versa. The notorious inverted file is a source of the fears for our personal privacy which computer science engenders. Consider as illustration a centralised computer system for storing hotel reservations. In the United Kingdom a person staying at a hotel is obliged to record his name, address and the date in a register provided by the hotel for this purpose. If these registers be stored centrally in a computer system then it becomes a simple matter to invert the information file and generate a list of the successive hotels in which a particular person stays. And just as in Fig. 2 the close association of the word 'atomic' with the word 'energy' can readily be established, so from an inverted file of hotel registers the association of a 'Mr. Smith' with a 'Miss Brown' could immediately be identified as they move successively from hotel to hotel. This trivial example illustrates the power of the computer to rearrange a data base of information to yield associations and correlations that could not otherwise be easily obtained. It is with considerations such as these that the recent White Paper on Computers and Privacy was concerned and which has led to the establishment of a Data Protection Committee charged with the task of recommending suitable legislation.

In using the index to search Acts of Parliament an enquirer has to express his legal enquiry in a form suitable for presentation to the computer. He does this using a language specially designed for the purpose, what is known in computer science as an interrogation or command language. A simple form of command language is one which allows the user to look for words in the text—the operands—connected by the Boolean operators .NOT. and .AND. and .OR. to form a compound logical statement. The text of the legal documents is then combed with this logical statement to retrieve those documents which satisfy it. Once identified, the titles of these documents are presented to the user on a television screen or he may display the full text of the retrieval documents in order to formulate an improved question on the basis of this new material.

The various logical operations can each be represented by an arrangement of electronic switches and Fig. 3 shows the equivalent circuits. Thus two switches in series, p followed by q, correspond to the logical operation .AND. applied to p and q as operands. Similarly p .OR. q is equivalent to two



*Figure 2*

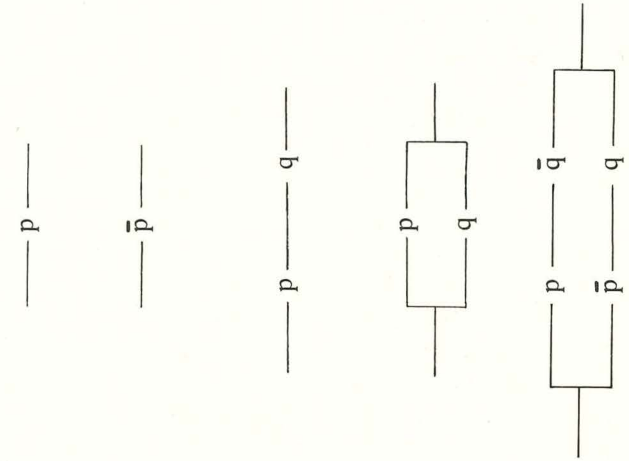
EXTRACT FROM CONCORDANCE

| DOCS | FREQ | WORDS | | |
|---|---|---|---|---|
| 30 | 69 | ATOMIC | | |
| | | 1. 1. 9.K. | 2. 2. 27.L. | 5. 2.119.L. |
| | | 5. 2.149.L. | 5. 2.188.L. | 6. 2. 49.L. |
| | | 8. 2. 64.L. | 9. 2. 76.L. | 10. 2. 22.L. |
| | | 11. 1. 7.M. | 11. 2.125.L. | 11. 3. 57.L. |
| | | 12. 2. 62.L. | 12. 3. 32.L. | 12. 3. 56.L. |
| | | 13. 2. 58.L. | 13. 3. 38.L. | 13. 8. 40.L. |
| | | 13. 9. 43.L. | 14. 2. 14.L. | 17. 3. 26.L. |
| | | 19. 2. 19.L. | 19. 2. 26.L. | 19. 2.133.L. |
| | | 19. 5. 12.L. | 20. 3. 57.L. | 44. 4. 58.L. |
| | | 46. 1. 11.K. | 47. 1. 4.M. | 47. 2. 12.L. |
| | | 47. 4. 54.L. | 48. 3. 31.L. | 48. 3. 80.L. |
| | | 48. 3.175.L. | 48. 3.213.L. | 48. 3.225.L. |
| | | 48. 3.244.L. | 48. 4. 56.L. | 49. 2. 24.L. |
| | | 49. 2. 56.L. | 59. 2. 31.S. | 59. 4. 21.S. |
| | | 59. 4. 44.S. | 59. 5. 19.S. | 59. 6. 39.S. |
| | | 59. 6. 71.S. | 59.12. 35.S. | 59.12.136.S. |
| | | 59.15. 24.S. | 61. 2. 19.L. | 67. 1. 14.K. |
| | | 68. 2. 15.L. | 69. 2. 8.L. | 91. 2. 56.L. |
| | | 113. 1. 6.M. | 113. 2. 7.L. | 113. 2. 23.L. |
| | | 113. 2. 57.L. | 113. 4. 23.L. | 119. 9. 57.S. |
| | | 119.13. 13.S. | 119.14. 11.S. | 121. 2. 34.L. |
| | | 121. 2.282.L. | 125. 2. 78.L. | 125. 2. 98.L. |
| | | 150. 2. 28.L. | 150. 2. 43.L. | 150. 2.429.L. |
| 30 | 71 | ENERGY | | |
| | | 1. 1. 10.K. | 2. 2. 28.L. | 5. 2.120.L. |
| | | 5. 2.150.L. | 5. 2.189.L. | 6. 2. 50.L. |
| | | 8. 2. 65.L. | 9. 2. 77.L. | 10. 2. 23.L. |
| | | 11. 1. 8.M. | 11. 2.126.L. | 11. 3. 58.L. |
| | | 12. 2. 63.L. | 12. 3. 33.L. | 12. 3. 57.L. |
| | | 13. 2. 59.L. | 13. 3. 39.L. | 13. 8. 41.L. |
| | | 13. 9. 44.L. | 14. 2. 15.L. | 17. 3. 27.L. |
| | | 19. 2. 20.L. | 19. 2. 23.L. | 19. 2. 42.L. |
| | | 19. 2.134.L. | 19. 5. 13.L. | 20. 3. 58.L. |
| | | 44. 4. 59.L. | 46. 1. 12.K. | 47. 1. 5.M. |
| | | 47. 2. 13.L. | 47. 4. 55.L. | 48. 3. 32.L. |
| | | 48. 3. 32.L. | 48. 3.176.L. | 48. 3.214.L. |
| | | 48. 3.226.L. | 48. 3.245.L. | 49. 2. 25.L. |
| | | 49. 2. 57.L. | 59. 2. 32.S. | 59. 4. 22.S. |
| | | 59. 4. 45.S. | 59. 5. 20.S. | 59. 6. 40.S. |
| | | 59. 6. 72.S. | 69.12. 36.S. | 59.12.137.S. |
| | | 59.15. 25.S. | 61. 2. 20.L. | 67. 1. 15.K. |
| | | 68. 2. 16.L. | 69. 2. 9.L. | 91. 2. 57.L. |
| | | 113. 1. 7.M. | 113. 2. 8.L. | 113. 2. 24.L. |
| | | 113. 2. 58.L. | 113. 4. 24.L. | 119. 9. 58.S. |
| | | 119.13. 14.S. | 119.14. 12.S. | 121. 2. 35.L. |
| | | 121. 2.283.L. | 121. 2.641.L. | 125. 2. 79.L. |
| | | 125. 2. 99.L. | 150. 2. 29.L. | 150. 2. 44.L. |
| | | 150. 2.430.L. | 150. 2.858.L. | |

*Figure 3*

SOME LOGICAL FUNCTIONS AND THEIR CIRCUIT EQUIVALENTS

| | | |
|---|---|---|
| p | logical affirmation | p |
| not p | logical negation or complement | .NOT.p |
| p and q | logical product or conjunction | p.AND.q |
| p or q or both | logical sum or inclusive disjunction | p.OR.q |
| p or q but not both | logical sum or exclusive disjunction | p.XOR.q |



switches in parallel. By means of these circuit diagrams the syntax of a sentence may be represented in a flowchart showing the arrays of switches connected in a pattern which corresponds to the logic of the statement. Flowcharts of this type are widely used in computer science as an aid in the design of computer programs and as a means of communicating the structure of a program in a form independent of a particular computer language. But they are also of use in demonstrating unambiguously the logic inherent in a legal statement.

A noteworthy example is provided by s.7 of the Official Secrets Act 1920 whose text is as follows:—

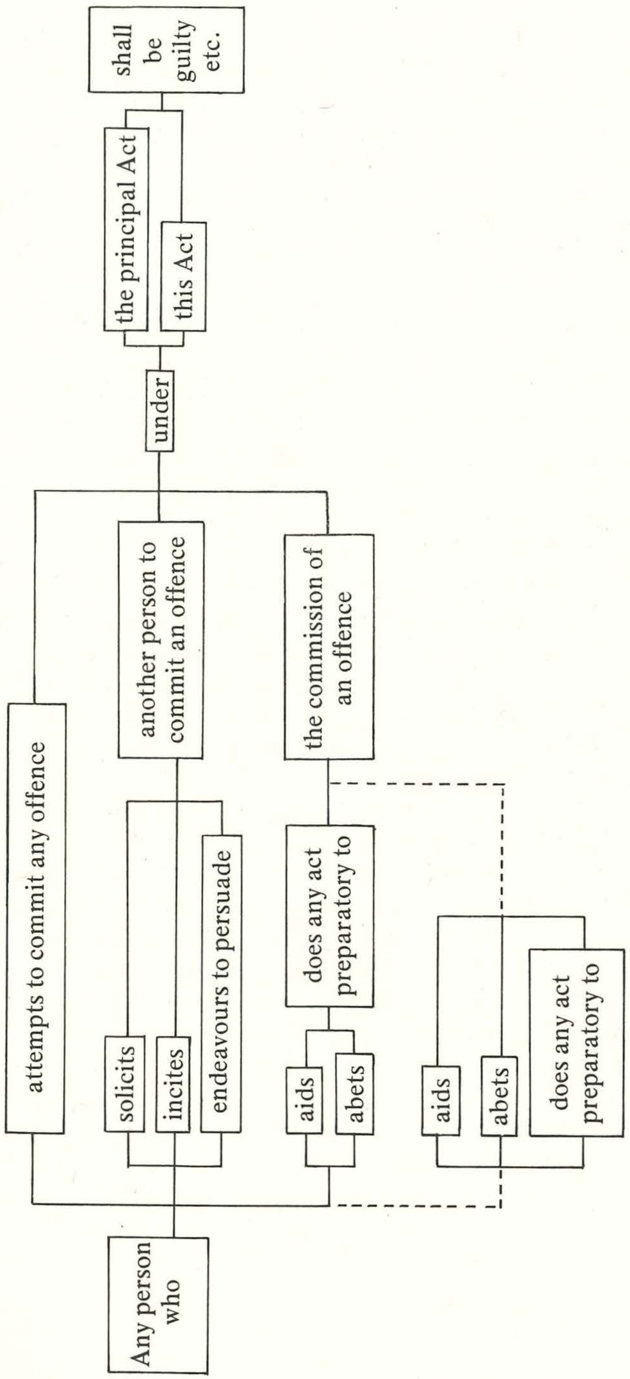
> 'Any person who attempts to commit any offence under the principal Act, or solicits or incites or endeavours to persuade another person to commit an offence, or aids or abets and does any act preparatory to the commission of an offence under the principal Act or this Act, shall be guilty . . .'

The corresponding flowchart is shown by the full lines in Fig. 4. To commit an offence within the section an accused person has to conduct himself so as to close the circuit between the left and right hand sides of the flowchart. There are twelve ways of doing this, that is twelve distinct routes across the chart.

The construction of this section was a matter which came before the Court of Criminal Appeal (as it then was) in 1959 in the case of *R. v. Oakes.** The appellant had been charged (inter alia) with doing an act preparatory to the commission of an offence against the Official Secrets Act 1911—the principal Act—and it was argued on his behalf that since there was no evidence that he had aided or abetted the commission of an offence the doing of a preparatory act by itself was not an offence against the section. In other words the 'and' in the words 'or aids or abets and does any act preparatory to' is conjunctive. The argument was rejected by the court on the grounds that this literal construction produces an unintelligible result. In reaching this decision the court observed that aiding or abetting the commission of an offence is itself an offence by reason of the Accessories and

* R. v. Oakes, (1959) 2 Q.B. 350.

Figure 4




S.7 Official Secrets Act 1920

*R. v. Oakes* 1959 2Q.B.350

Abettors Act 1861 and if the literal interpretation of s.7 were followed then the offence of aiding or abetting would in that context be limited to aiding or abetting accompanied by an act preparatory to the commission of an offence, and this cannot be so. Thus the decision of the court was that in s.7 the 'and' has to be read as 'or' so that the true construction is shown by the dotted lines in Fig. 4. To a computer scientist it seems remarkable that in a penal statute carrying heavy consequences for a convicted person a conjunction must be read as a disjunction but that is the law as it stands. My purpose in presenting this example is to show how one of the tools of the computer scientist, the flowchart, is a clear and unambiguous way of displaying the logic of a legal statement.

Turning now to the types of logical operator used in legal interrogation languages these are shown in Fig. 5 for the LEXIS and WESTLAW systems which are marketed in the USA and the STATUS system as originally developed in the United Kingdom. From the figure it is seen that the languages are based primarily on the Boolean operations of conjunction, disjunction and negation applied to the words of the text. In addition two of the systems include a collocation operator, that is they allow a search for words in specific relation to each other, for example the word DUTY followed within five places by the word CARE. All three systems offer the possibility of searching for phrases, that is words in sequence such as RES IPSA LOQUITOR and will allow a search for words that have been truncated, thus the search for ESTOP* or ESTOP! would find words such as ESTOPPEL, ESTOPPAGE, or ESTOPPED having the same five characters as root.

These three interrogation languages use different symbols to denote the same operators. A search for the word CAR in conjunction with the word AUTOMOBILE but in disjunction with the word BICYCLE would in the LEXIS system be written as

CAR AND AUTOMOBILE OR BICYCLE ______ (1)

whereas in the STATUS system it would be

CAR .AND. AUTOMOBILE .OR. BICYCLE ______ (2)

*Figure 5*

COMPARATIVE SEARCH REQUESTS IN THREE
LEGAL INTERROGATION LANGUAGES

| Primitive Operation | LEXIS | WESTLAW | STATUS I |
|---|---|---|---|
| Conjunction | CAR AND BICYCLE | CAR & BICYCLE | CAR .AND. BICYCLE |
| Disjunction | CAR OR AUTOMOBILE | CAR AUTOMOBILE | CAR .OR. AUTOMOBILE |
| Negation | BICYCLE AND NOT CAR | BICYCLE % CAR | BICYCLE .AND. .NOT. CAR |
| Collocation | DUTY W/5 CARE | — | DUTY / 5, 5/ CARE |
| Phrase | RES IPSA | "RES IPSA" | RES IPSA |
| Truncation | ESTOP! | ESTOP* | ESTOP* |

It might be thought that given the same set of documents (what logicians call the same universe of discourse) the documents retrieved by these two statements would be the same. But surprisingly this would not be so because the precedence value given to the logical operators differs in the two systems. The LEXIS question is identical to

CAR AND (AUTOMOBILE OR BICYCLE)

whereas the STATUS question is equivalent to

(CAR .AND. AUTOMOBILE) .OR. BICYCLE

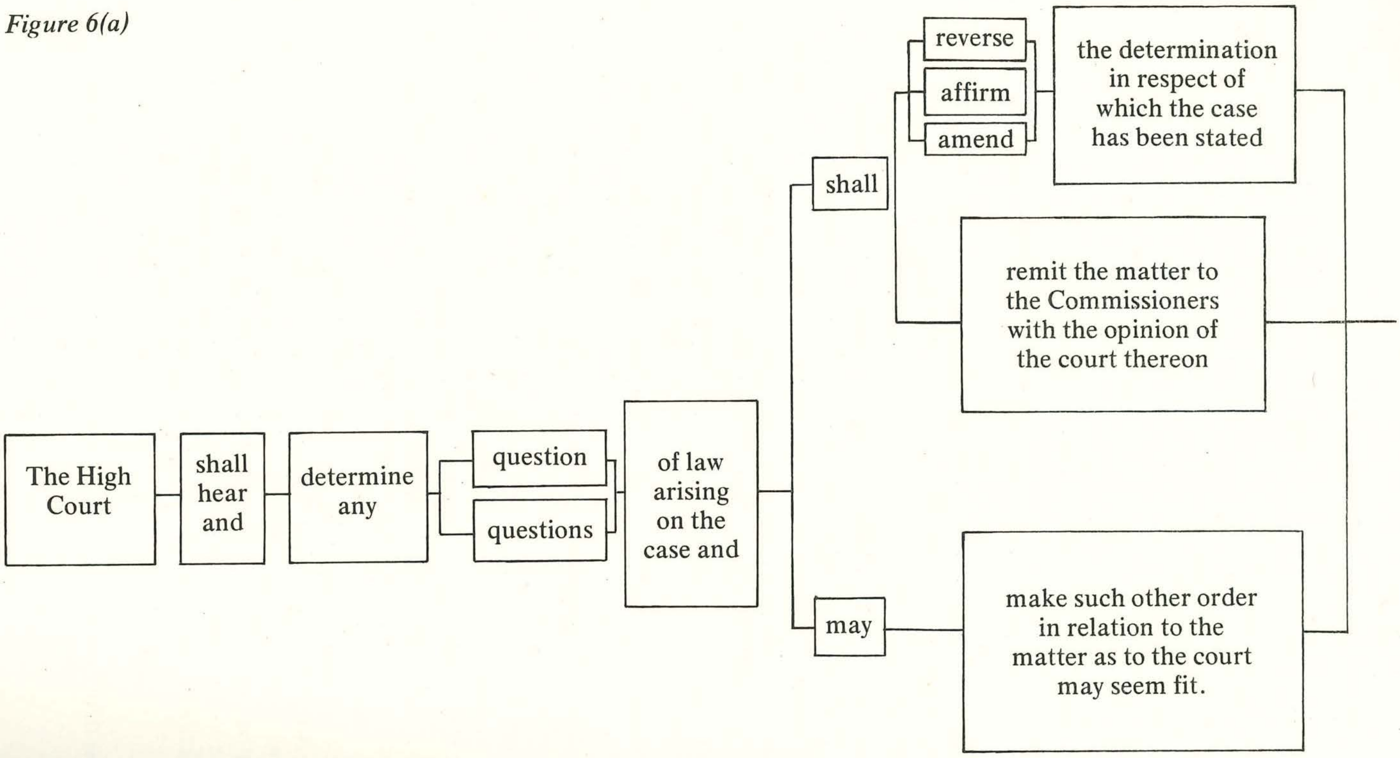
and these are different questions. This is because LEXIS gives higher precedence to OR than to AND whereas STATUS deals with the operators from left to right. Thus there is an ambiguity in expressions (1) and (2) which is resolved by the two systems in different ways. Computer scientists are familiar with these parsing problems which have to be solved in the construction of compilers for programming languages such as FORTRAN or ALGOL or APL. In these languages the order of precedence of the operators is unambiguously defined by the compiler which translates programs written in these languages into the basic machine language which drives the computer.

In natural languages such as English the precedence of the small but vital words 'and' and 'or' is not so defined and the resultant ambiguities have on occasion to be resolved by the courts. Such an occasion arose in an income tax case, *Slaney (Inspector of Taxes) v. Kean**, where the construction of what is now s. 56(6) of the Taxes Management Act 1970 fell to be decided. The subsection reads as follows:—

> 'The High Court shall hear and determine any question or questions of law arising on the case, and shall reverse, affirm, or amend the determination in respect of which the case has been stated, or shall remit the matter to the commissioners with the opinion of the court thereon, or may make such other order in relation to the matter as to the court may seem fit'.

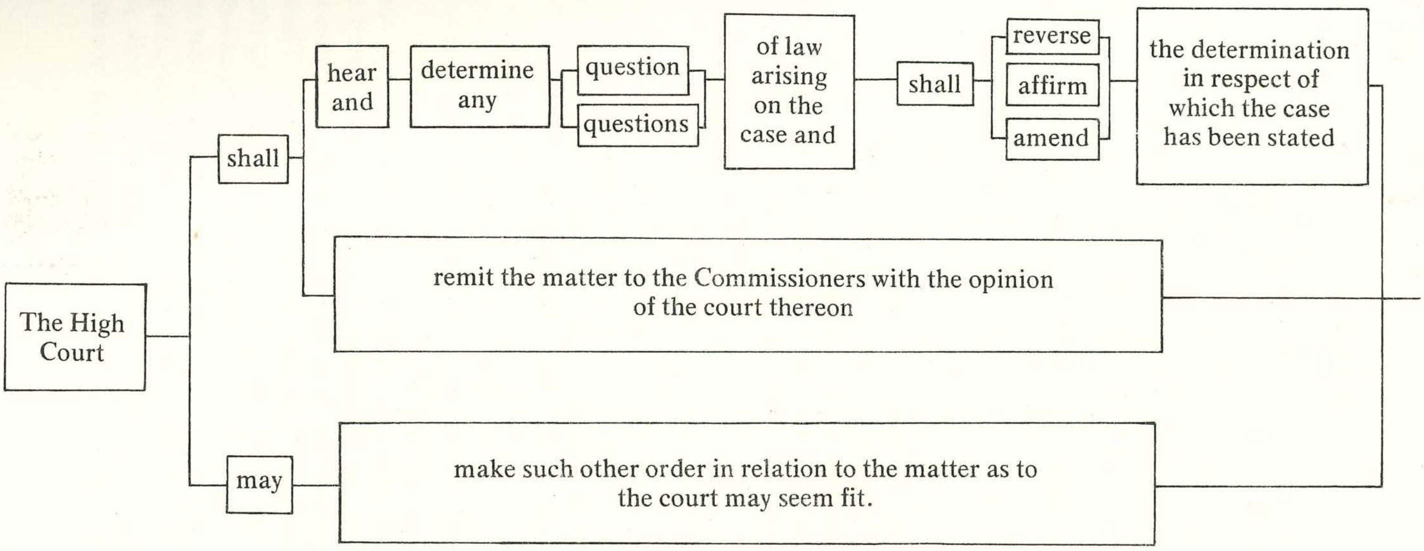
* Slaney (Inspector of Taxes) v. Kean, (1970) 1 Ch. 243.

*Figure 6(a)*




*Slaney (Inspector of Taxes) v. Kean, 1969*

*Figure 6(b)*




*Slaney (Inspector of Taxes) v. Kean, 1969*

In this case the taxpayer was employed in the film industry and was assessed to income tax under schedule E on the emoluments from his employment. He claimed to be entitled to deduct a sum of £200 expended in the purchase and maintenance of dress clothes for his wife. The Inspector of Taxes did not accept his claim and so he appealed to the General Commissioners who held that in the special circumstances of the film industry the expenses were wholly, exclusively and necessarily incurred in the performance of his duties. From this decision the Crown appealed and when it came before the High Court on case stated a document was placed before the judge, Mr. Justice Megarry, recording that the parties had agreed that the appeal be allowed and that the decision of the General Commissioners be reversed. The court was invited to make an order in these terms. The question before the court was whether s.56(6) of the Taxes Management Act 1970 gave the court jurisdiction to make such an order without first hearing any question or questions of law arising on the case. That is to say, is the true interpretation of the subsection as displayed in Fig. 6(a) or as in 6(b)? Does the obligation to hear and determine any question or questions of law govern all three limbs as in Fig. 6(a) or is the third limb of the sentence free of these opening words as in Fig. 6(b)? The answer turns on the relative precedence given to the words 'and' and 'or' in the sentence. In the result the court preferred the interpretation of Fig. 6(a) so that no order could be made until the appeal was heard. This example again shows how instructive it is to display the logic of the section using the type of flowchart commonly used in computer programming.

In discussing this case I have strayed from the task of explaining how these logical operators can be used to express legal enquiries in a form which allows the computer to retrieve those passages relevant to an enquiry. These logical expressions can be as complex and elaborate as desired; the process of searching the text is nonetheless rapid because it is the concordance that is used rather than the text itself. This makes the search a parallel rather than a sequential process.

A cardinal matter is the effectiveness of this method of searching. How easy is it to retrieve all those documents that



contain the legal concepts one is looking for without being overwhelmed with a large surplus of irrelevant material? My experience shows that Boolean searching on statutory data bases is remarkably effective if an interactive system is used. Feedback is the key to success. A first question is used to retrieve a few relevant documents, these are used to improve the question and by this iterative procedure a question is finally formulated which summons up all the answers that are required. The effectiveness of an information retrieval system is usually evaluated in terms of recall and precision. By recall is meant the proportion of all relevant documents in the data base which are retrieved, and precision is defined as the proportion of retrieved documents which are relevant. Using a data base consisting of the full text of statutes and treaties a trained enquirer can achieve a recall approaching 100% with a precision of about 50%. A lawyer demands high recall but can tolerate low precision—he simply transfers to the waste-paper basket those retrieved documents which are of no use to him. It is a feature of full text systems that high recall is accompanied by low precision. One is reminded of the small boy who asked his mother to explain why the television set went on when the switch was turned. When his mother suggested that he ask his father, the boy replied that he didn't want to learn *that* much about it. Similarly with full-text information systems, they tend to supply more information than we wish to have, but included in this is all we need to know.

An interrogation language founded on Boolean operators is inherently a crude one because any document can only take up one of two values—true or false—in relation to a question. A Boolean seach is a black or white process. What is really required is some method of ranking the documents in shades of grey so that they may be placed in an order of relevance. This brings us face to face with a problem of great current interest in computer science: computers can readily be used to find documents or symbols which match exactly, but how can computers be used to identify near matches, close associations which do not possess exact correspondence?

One method of ranking documents in relation to a question is provided by what is known as the vector method. In this technique each document is stored as an array of numbers—that is a vector—representing the words in the document and their frequency of occurrence. A question is then expressed as free text rather than as a logical expression and this is similarly converted into a vector. By comparing the question vector in turn with each document vector a similarity coefficient can be calculated. Each document becomes a vector in n-dimensional space, where n is the number of different words in the document set and its closeness to the question is measured by the angle between it and the question vector.

Documents can be compared not only with a question but with each other to determine their similarity on the basis of the words they contain. The similarity coefficients for each pair of documents can then be used to classify the collection using some standard form of clustering algorithm. Such methods of classifying legal documents have been successfully applied to the Conventions and Agreements of the Council of Europe using the full text of these treaties in both English and French versions.

Recently my colleagues and I at the University of Kent used this type of cluster analysis to generate a new classification of Acts of Parliament. The way in which statutes are arranged is of concern to the practitioner who constantly consults them. The first three official editions of Statutes Revised published at various dates between 1885 and 1950 arranged the statutes in chronological order but the new edition of Statutes in Force which began publication in 1972 groups them by subject matter. Statutes which are considered to belong to a common category are assembled together in the same loose-leaf volume. There are many different ways of selecting these groups but the aim must be to arrange them in a series of volumes so that a practitioner consulting the statutes on a legal problem does not have to consult more than one or two volumes.

The computer can be of great assistance in this task of classification and the method that my colleagues and I have used is to construct a vector for each Act of Parliament not from the words they contain but from their marginal citations. This reduces the size of the vectors by a hundredfold



and renders the process an easily manageable one. The use of citations or bibliographic references in classifying documents, particularly scientific papers, is a common one nowadays. It seems particularly apt, as well as inexpensive, for generating a new classification of Acts of Parliament. Statutes commonly cite other statutes, statutory instruments, and in recent years the regulations of the European Communities, and one can be sure, as one cannot always with scientific papers, that material cited is pertinent. It is highly likely that two Acts which have similar citation patterns will be closely similar in content.

Accordingly we collected the marginal citations for the public general Acts for the three years 1973-75 and used the computer to construct a vector for each Act. In these three years there were 210 Acts containing 4390 marginal citations, 1110 of which were different, an average of 21 citations per Act. The computer was then used to construct similarity coefficients for each of the 21,445 different pairs of Acts.

Some of the results are shown in Fig. 7 which lists the more highly similar Acts for the years 1973-75 with the calculated similarity coefficients. On this basis the most nearly similar Acts are the Pensioners' Payments and National Insurance Act 1973 and the Pensioners' Payments Act 1974 having a similarity coefficient of 0.891. As would be expected the four Finance Acts of 1973-75 group closely together as does the Social Security legislation. From these similarity coefficients using standard clustering methods we have generated a dendrogram, or tree diagram, of the 210 Acts showing the groups into which they form. With little further complication this method can be used to produce a new classification of all the Acts of Parliament currently in force.

These are some of the ways in which the youthful subject of computer science may be of use to the mature discipline of the law. The topics I have used to illustrate the theme have been chosen with partiality to reflect my own interests but there are many others that could have been selected. Lawyers are slowly beginning to be aware of the benefits computers have to offer and a momentum is gathering about the discovery of new applications. In the United Kingdom the Society for

Figure 7

PUBLIC GENERAL ACTS 1973—1975

AN EXTRACT FROM THE CLASSIFICATION
BASED ON MARGINAL CITATIONS

The figures show the similarity coefficient for each cluster.

| | | | |
|---|---|---|---|
| Pensioners' Payments & National Insurance Act 1973 | 0.891 | | |
| Pensioners' Payments Act 1974 | | | |
| Export Guarantees Act 1975 | 0.455 | | |
| Export Guarantees Amendment Act 1975 | | | |
| Representation of the People Act 1974 | 0.444 | | |
| Representation of the People (No. 2) Act 1974 | | | |
| Finance Act 1974 | 0.708 | 0.622 | 0.433 |
| Finance (No. 2) Act 1975 | | | |
| Finance Act 1973 | | | |
| Finance Act 1975 | | | |
| Breeding of Dogs Act 1973 | 0.438 | | |
| Guard Dogs Act 1975 | | | |
| Social Security Amendment Act 1974 | 0.615 | 0.612 | 0.385 |
| Social Security Benefits Act 1975 | | | |
| National Insurance Act 1974 | | | |
| Social Security Pensions Act 1975 | | | |
| Insurance Companies Amendment Act 1973 | 0.856 | 0.405 | 0.373 |
| Insurance Companies Act 1974 | | | |
| Iron and Steel Act 1975 | | | |
| Trade Union and Labour Relations Act 1974 | | | |



Computers and Law, a charitable corporation, has been founded to encourage the use of computers in legal practice; and this year (1977) the Commission of the EEC has embarked on a major design study of a legal retrieval system to serve the member countries. In the next few years we may expect the application of computers to the service of the lawyer to become commonplace.

Nevertheless the most significant outcome of the interaction of computer science with the law will not in my opinion be the direct assistance lawyers will receive in their daily work, substantial and beneficial though this may be. The more far-reaching consequence will be a new insight into the law and the legal process. For computer science is a discipline based ultimately on that most fundamental of distinctions, the choice between zero and unity. The application of computer science to legal problems means that the law will have to be re-examined and re-stated in these fundamental terms and from this salutary process the law is likely to emerge both better understood and more effectively administered.

# *Bibliography*

A recent account of the development of computer-based methods of legal information retrieval with an appraisal of current North American and European systems is provided in *Legal Decisions and Information Systems* by Jon Bing and Trygve Harvold published by Universitetsforlaget, Oslo (1977). The LEXIS and WESTLAW systems are described and compared by James A. Sprowl in 'Computer-Assisted Legal Research: Westlaw and Lexis', *American Bar Association Journal,* 320-323 (March 1976). An early paper on the STATUS system is 'Mechanised Searching of Acts of Parliament' published in *Information Storage and Retrieval,* 289-297 (1970).

The use of symbolic logic to analyse the syntax of legal statements has been discussed in a long series of papers by Layman E. Allen of which one of the earliest is 'Symbolic Logic: a razor-edged tool for drafting and interpreting legal documents', *Yale Law Journal,* 833-879 (1957). A formal introduction to these techniques is also given in *Outlines of Modern Legal Logic* by Ilmar Tammelo published by Franz Steiner Verlag Gmbh Wiesbachen (1969).

Professor G. Salton in his book *The Smart Retrieval System—Experiments in Automatic Document Processing,* Prentice-Hall, New Jersey (1971) has discussed the representation of documents in vector form and their searching and classification by this means. The technique has been applied to legal documents on a substantial scale by Jillian Boreham in 'Cluster Analysis of Legal Documents', Ph.D. Thesis, University of Kent at Canterbury (1976). An interesting application of vector classification methods to test the authenticity of evidence given in criminal proceedings is discussed in 'Cluster Analysis in Court', *Criminal Law Review,* 175-180 (March 1976).