Molecular Substructure Searching: Minicomputer-Based Query Execution

T. R. HAGADONE* and W. J. HOWE*

The Upjohn Company, Kalamazoo, Michigan 49001

Received April 8, 1982

The Upjohn Co. COUSIN system is being developed to allow easy access to in-house, preclinical, compound-related information. The substructure search capability of COUSIN has been implemented by using a dedicated minicomputer to perform substructure query execution. As a result of the efficiencies possible with the minicomputer approach, substructure searches over a sample 63 000-compound database are typically performed in less than 30 s, thus providing a high degree of user interaction.

INTRODUCTION

Scientists and others involved in drug development have a need for convenient access to existing data related to their area of work. At the Upjohn Co., the COUSIN system is being developed to provide such access to in-house, preclinical data on a growing collection of compounds. The COUSIN database contains chemical, biological, sample inventory, patent, and miscellaneous compound-related data. The database may be accessed from a number of graphics and nongraphics terminals located throughout the company. An interactive language with commands for both novice and expert users is available for querying the data base.

A key part of the COUSIN system is its substructure search (SS) capability. The SS component was designed to provide an interactive facility that could be used directly by research scientists and others to meet their substructure searching needs. Searches were required to execute quickly enough to allow the user to perform a number of searches during each session at the terminal. Fast search execution allows exploration of the effects that subtle query modifications have on the search results. The SS component of the COUSIN system was designed and implemented over a 2-year period and has been in general use by Upjohn scientists for over 1 year. A novel approach using a dedicated minicomputer for SS execution has resulted in elapsed search times of less than 30 s for most queries.

A graphical user interface allows chemists with no previous computer experience to enter their own queries after only a few hours of training. An especially attractive feature of the substructure graphics is the ability to specify variable substituents through the use of "Rk" groups. The interface has been described previously[1] and will not be covered here. This paper will focus on the design and performance of the part of the SS system that executes the query once it has been defined by the user. The appearance of the system from the user's point of view will be discussed first, followed by an outline of the overall system design and the flow of control during a search. The screening and atom-matching operations will then be examined, and finally, some performance statistics will be presented on the basis of the results of a set of user-defined SS queries.

USER'S POINT OF VIEW

The user initiates a substructure search by selecting the SS command from the menu of COUSIN commands. The graphical controls are activated, and the user draws the substructure query using the graphics controls described previously.[1] When the user has finished entering the query, he presses the search button, and the query is checked for a number of possible errors. Once all errors have been corrected, the search begins. A message is displayed on the screen, indicating that the initial database scan is being performed. After about 9 s, a message is displayed that indicates that the atom-matching operation is being performed. A counter on the screen is updated every few seconds to display the number of hits that have occurred so far. When the atom-matching operation has been completed, after about 21 s, the user can select the DISPLAY command from the menu to display the compounds that have been found. If the compounds displayed are what the user wanted, then the structures and any additional compound-related information can be saved on disk for later reference, or hard-copy output may be requested. The user also has the option of having compounds displayed at the terminal as search hits occur; however, since the total search usually completes within 30 s, most users prefer to perform the search and see how many compounds have been found before displaying the search results.

Often the results of the search will not be exactly what the user desired, and by looking at a few of the structures, he will be able to get some ideas about what changes should be made to the query. At this point, the user reenters the SS command and modifies the original query to better represent that which is really wanted. In fact, in complex cases, it may take several query modifications and subsequent searches before the desired result is obtained; this is only feasible because of the quick response of the system.

Occasionally a search will take longer than usual because the user has entered a query which the system cannot handle well. In such cases, COUSIN will provide feedback to the user, letting him know how the search is progressing and approximately how long it will take to complete. The user has the option of continuing the search or of stopping it and refining his query before continuing.

OVERALL DESIGN

The traditional approach[2] to SS execution has been to break the searching operation into two steps. First, a screening operation is performed to quickly eliminate those compounds that cannot possibly match the query. Then a detailed atom-matching operation is performed to determine which of those compounds that have passed the screen are actually matched by the query.

A number of methodologies have been proposed for generating and searching screens.[3-7] The current state of the art of screening allows screen sets to be generated that will provide 95–99.99% screenout for most substructure searches. Occasionally, there will be "poorly defined" queries that give low screenout no matter how carefully the screen set is defined. Screens may be organized into sequential lists of bit strings, inverted lists of bit strings or registry numbers, or a combination of the two. In addition, a screen may be physically close to its associated connection table or may be in a separate file. Sequential lists have the advantages of simplicity of data structure and associated algorithms, whereas inverted lists have

0095-2338/82/1622-0182$01.25/0

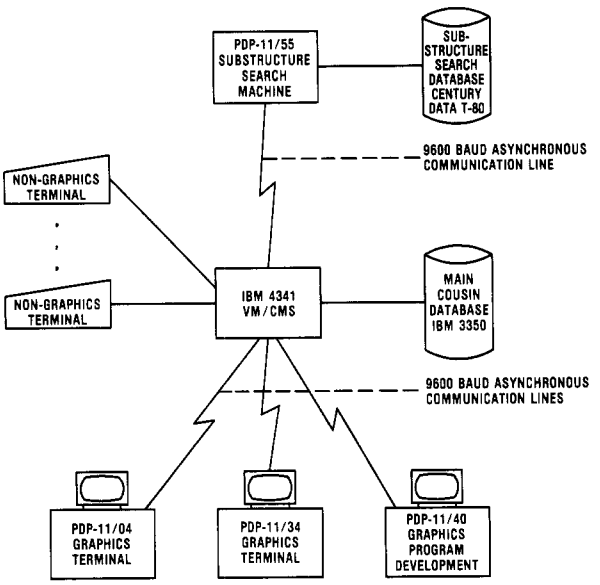

Figure 1. COUSIN hardware configuration. Each graphics terminal consists of a PDP-11 processor, small disk drive, graphics display, graphics tablet, and keyboard.

the advantage that only a fraction of the screening data must be processed for a given query. The inverted list approach, however, usually requires many movements of the disk heads which can have an unfavorable effect on the elapsed time of a search.

The practical methods of atom matching are set reduction[8-10] and breadth- or depth-first exhaustive matching.[11] Exhaustive-matching performance may be improved by matching the statistically least common atoms and fragments first. Current atom-matching algorithms provide atom matches which average 2–10 ms per match for present-day medium-range mainframe computers. With atom matching there will occasionally be cases where the matching takes considerably longer due to the need to test a large number of alternative matchings (e.g., queries consisting of all carbon atoms).

Current SS systems performing both a screening and atom-matching step do not provide a high degree of interaction. The best systems require several minutes of elapsed time to search a database of 63 000 compounds, and many systems require an overnight wait for the search results. The total amount of central processor unit (CPU) time used by these systems during a search is not a limiting factor as much as is the amount of time the search program spends waiting for other users to run and for data to be transferred into main memory from secondary storage. For example, one good current SS system was reported to have an average elapsed search time of 17 min for a number of searches over a 120 000-compound database while the average CPU time was only 1 min 21 s.[12]

By eliminating CPU contention with other users and decreasing the time spent waiting for data to be transferred, it is possible to substantially improve SS response time. In the COUSIN system we have removed these delays by moving query execution from the mainframe computer to an attached minicomputer dedicated to SS. Figure 1 shows the COUSIN hardware configuration. SS queries are sent from the mainframe to the minicomputer system, called the substructure search machine (SSM), which contains a specially formatted database of screens and connection tables. Software running on the SSM executes the query and returns the results to the mainframe system as hits occur. Since the SSM is dedicated to SS, it is possible to program it in such a way that a much higher rate of data transfer can be achieved than is possible on a multiuser mainframe system. This will be described in detail in a later section. The SSM approach also provides a uniform response to SS queries of equal complexity, since search time is almost completely independent of the current load on the mainframe system.

The design of a SS database structure can greatly affect SS performance and is not only determined by the need for fast searching but is also influenced by the ways in which the database is intended to be used. The following questions must be answered: Is multiple, simultaneous user access to be allowed? Must quick, full-structure searching be supported as well as substructure searching? Should searches return registry numbers in increasing order?

Different situations will result in different answers to these questions. On the basis of our particular environment, we made the simplifying decision that only one search at a time would be allowed. Since, on the average, only a few substructure searches are performed each day at Upjohn, there is a small probability that two users will execute searches at the same time. If a conflict does occur, the user who is blocked out will only need to wait about 30 s for the machine to become available. Since full structure searching is performed on the mainframe system, no facilities to support it are necessary on the SSM. Registry numbers are returned from the search in approximate increasing order, but an automatic sort following the search assures that the returned numbers are completely ordered.

The database on the SSM consists of three files. The screen file contains the structure screens, the connection table file contains the connection tables in a special format, and the index file is a hash table that allows the quick location of the screen and connection table for a compound, given the registry number. The index file is not used for searching but allows the database to be updated efficiently. New compounds are added to the ends of the screen and connection table files, and updates to existing compounds are performed in place.

The SSM approach has allowed us to obtain the type of response we desired for SS; however, there are some disadvantages in using a separate minicomputer for SS that should be mentioned. Additional equipment must be purchased, installed, and maintained. Most of the time this equipment is sitting idle, waiting for a user to make a search request, although it is possible to use the minicomputer for background jobs such as printing and plotting during periods when no searching is occurring. The extra expense and complexity added by the SSM approach must therefore be balanced against the need for fast substructure searches.

FLOW OF CONTROL DURING A SEARCH

Although the SS system consists of a network of three computers, the user is not aware of this and instead sees a single homogeneous system. The user defines a query using one of the PDP-11-based graphics terminals (see Figure 1). When the query is complete, it is checked for possible errors by the PDP-11 and is then sent to the mainframe system, where further error checking takes place. Once the query has passed all error checks it is processed on the mainframe system to produce a structure screen and a specially formatted connection table which are then sent to the SSM. Next, the SSM performs a full-structure screen on the 63 000 compounds which takes an average of 9 s. If the percent of compounds that are screened out is below a certain threshold value, the user is notified that the search may be slow and is given the option of refining the query to make it more specific. If the screenout is above the threshold value, there is no need to interact with the user.

Following the screening operation, atom matching is performed on the remaining compounds, and registry numbers

are sent back to the mainframe system as hits occur. An area on the display screen is updated by the mainframe system every few seconds to indicate the current number of search hits.

If atom matching is still in progress after 1 min, the SSM suspends its operation and sends the current search statistics to the mainframe system, which then sends a message to the user indicating how much of the database has been searched, how many hits have been found, and approximately how long it will take to finish the search. The user then has the option of cancelling the search and refining the query or of continuing the search. In most cases a search will complete within 30 s, and no interaction with the user will be necessary; however, there will always be occasional difficult cases where the search will require a longer time. In such cases, we have found that it is best to provide information on how the search is progressing and let the user decide whether to continue or not.

Screening Operation. The screen set used by COUSIN is based on the BASIC[7] screens and contains several screen classes including atom count, augmented atoms, bond composition, degree of connectivity, element composition, linear sequence, ring count, and single-atom screens. We regenerated the set of augmented atom screens on the basis of statistics derived from our database and added a few additional screens such as ring screens that we felt would be useful. Our screen dictionary consists of 1590 individual screens which are mapped onto a 332-bit field for database storage. The mapping function was algorithmically determined, on the basis of the frequency of incidence of each screen. The probability of each bit being set in the final screen field is somewhat less than 50%. When the SS query is received from the user, the connection table is processed to generate all possible fragments for the screen classes in the dictionary. For each fragment generated, a search is made to determine if the fragment is in the screen dictionary, and if so the appropriate bit is set in the screen for the query. Rk groups[1] within the query are included in screen generation if they are required to occur at least once within the compound. Since the screen is generated automatically from the query connection table, the user is not required to have any knowledge of the structure of the screen dictionary. The screens for compounds to be inserted into the database are generated in a similar manner.

In order to maximize the speed of the screening operation, it is necessary to keep the CPU constantly supplied with data and so minimize the data-transfer wait time. To accomplish this, we designed a special storage format for the screening data which is shown in Figure 2. The screening file contains a contiguous set of 512-byte disk blocks which consist of alternating ten-block data areas, called big blocks, and two-block delay areas. The purpose of the delay areas is to allow the software and hardware enough time to start a new read operation before the next big block comes below the disk head. In this way, all of the data on a single track of the disk is read during one revolution. In addition, the disk block numbering system is skewed in such a way that when all of the data on a track has been read and a head switch occurs, the first block to be read on the next track will just be coming up to the new head when switching has completed. When all of the tracks in a cylinder have been read, the disk heads are moved to the adjacent cylinder, where reading continues. While the heads are being moved to the next cylinder, the disk makes one revolution during which no data is transferred.

Two buffers which each hold ten disk blocks are allocated in main memory so that the disk may be transferring into one of the buffers while the CPU is screening the data that has been previously read into the other one. The CPU and disk were selected so that the CPU is able to screen the data slightly faster than the disk can read it, for the majority of searches. In this way the disk is prevented from getting ahead of the CPU which would lead to wasted revolutions of the disk.

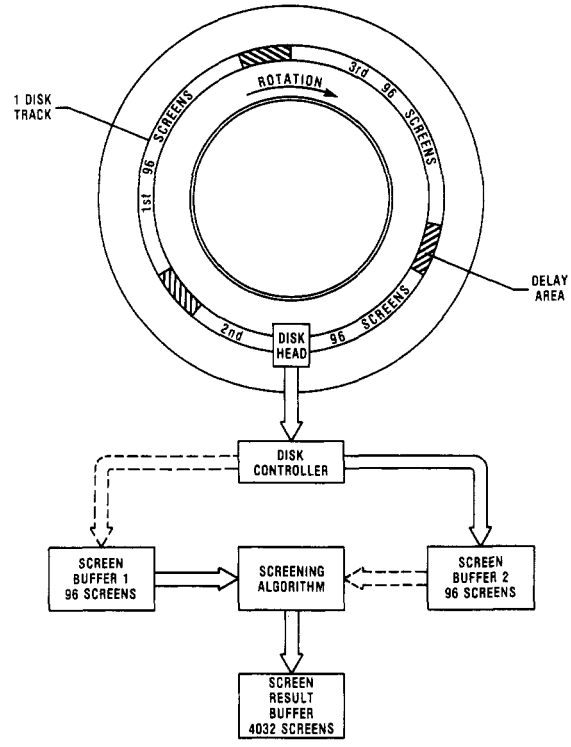

Figure 2. High-speed data flow during the screening process. The screening algorithm is processing data in buffer 1 while data are being transferred into buffer 2 from the disk. The delay areas on each track allow input operations to be initiated between the data areas which contain the screens, thus eliminating idle disk revolutions.

Each big block contains the screens for 96 compounds. The data is organized into inverted bit strings of length 96, where each bit string represents a single screen attribute. The screening algorithm then simply intersects the lists for those screen attributes that have been set in the screen generated for the query. The resulting lists indicate which compounds have passed the screen and are written to disk for later use by the atom-matching algorithm.

When the screening operation is complete, the SSM sends the screening statistics to the mainframe system. The mainframe system then decides, in conjunction with the user if necessary, whether or not to continue with the atom-matching part of the search.

Atom-Matching Operation. The atom-matching operation uses data passed from the screening operation to determine which compounds to process further. For maximization of the data-transfer rate, the connection table file is organized in a manner similar to the screen file; however, during atom matching, it is only necessary to read those big blocks in the file that contain connection tables for compounds that have passed the screening operation. A double-buffering scheme is used, where the read operation for the next big block that will be needed is started before the big block currently in main memory is processed. Although this technique helps keep the disk and CPU operating concurrently, it is possible that one will get ahead of the other, depending on the number of compounds to be processed and their complexity. In general, the atom-matching operation tends to put greater demands on the CPU than it does on the disk.

Each big block contains 26 registry numbers and connection tables. The connection tables contain one entry for each atom which describes the atom type, atom attributes, and a special atom screen. The atom attributes indicate whether the atom is a halogen or a metal, its charge, and how many hydrogens are attached. A single bit is assigned to each attribute to allow fast checking for the proper attributes during the search. The

atom screen is a 16-bit screen that is calculated and stored for each atom in the molecule and indicates the attributes of the bonds and atoms, up to four bonds away, in all directions from the atom for which the screen is calculated. The atom screens are used during atom matching similarly to the way the structure screen is used during screening. The difference is that the atom screens determine if a particular structure atom is a possible candidate for a substructure atom and thus help to reduce the number of false paths that must be followed. We have found that the atom screens do improve performance for many searches; however, the increase in search speed must be weighed against the cost of the extra disk storage required for the atom screens.

There is one entry in the connection table for each bond which describes the bond attributes and gives the atom numbers of the two atoms that the bond connects. The bond attributes indicate the bond type (single, double, triple, or aromatic) and whether the bond is cyclic or acyclic.

The connection table for the SS query is divided into a number of sections each of which describes a basic fragment or a Rk group fragment. A fragment is a connected structural entity. A query may contain one or more basic fragments, each of which must be present in a candidate structure for a successful match to occur. The atoms and bonds in each section are ordered according to the sequence in which they will be matched against candidate structures. Information is appended to the connection table which describes the attachment points on the basic fragments and valid group counts for each Rk definition.

The atom-matching algorithm matches each basic fragment of the substructure in turn and then matches any Rk definitions. Matching for each basic fragment begins at the least common atom (determined from overall database statistics) and continues, using a breadth-first search method, until a match occurs or all possible alternatives are exhausted.

Each Rk definition is matched by using a three-step process. In the first step a check is made to see that the candidate atom assigned to each position of attachment on a basic fragment matches the first atom in at least one of the Rk groups. A check is also made that at least one of the valid counts for each group falls within the range of possible counts for that group, based on the first-atom match at each attachment position. If the first step is passed, then an Rk group is matched to each attachment position, while checking to see that the maximum count is not exceeded for any Rk group. Once a group has been matched to each attachment position, the final step is to check that the number of times each group has been matched corresponds to one of the valid counts for that group. Each Rk definition is completely matched before an attempt is made to match the next. If a match fails at some point, each remaining alternative is tried in turn. As hits occur, the corresponding registry numbers are sent back to the mainframe system where they may be viewed immediately, together with associated data, or stored for later reference.

SEARCH STATISTICS

To obtain an objective view of how long SS queries were taking to execute, we gathered statistics on the searches that were performed during a 2-month period. The SS queries were formulated and entered into COUSIN by a test group of approximately ten scientists from different areas within Upjohn. The queries that the scientists defined were designed to help them in their work. No conscious attempt was made to influence them to select queries that we knew COUSIN would handle especially well. Statistics on a total of 103 SS queries were collected.

Table I. Search Statistics for 103 Searches Performed over the Sample Database of 63 000+ Compounds[a]

	mean	std dev	min	median	max
screenout percentage	98.31	4.66	69.65	99.70	100.00
total screening time, s	9.0	2.5	6.2	10.7	13.9
total atom-matching time, s	21.2	40.8	0.5	8.0	209.0
mean atom-matching time per compd, ms	12	10	1	10	79
total elapsed search time, s	30.1	40.2	6.8	15.9	215.3

[a] The high maximum elapsed search time was caused a "poorly" defined query that resulted in a 72% screenout.

The SSM contains a real-time clock as part of its hardware. Instrumentation was included in the SSM software to use the clock to collect statistics on the screening operation, the atom-matching operation, and the time spent transmitting results to the mainframe computer. The following items were recorded for each search: (1) the total number of compounds screened, (2) the amount of time spent performing the screening operation (both CPU and I/O wait time), (3) the number of compounds passed to the atom-matching step, (4) the amount of time spent performing the atom-matching operation (both CPU and I/O wait time), (5) the number of compounds that passed the atom match, (6) the amount of time spent sending registry numbers of matching compounds back to the mainframe system.

The results of the 103 searches are summarized in Table I. The screenout percentages were generally good except for three searches that had poor screenout (72%, 70%, and 79%). These queries had at most two noncarbon atoms and no interesting ring structures, resulting in only a few screen bits being set. The total screening time was usually 6.3 or 11.7 s, depending on the number of screen bits set. If the number of bits was above a certain threshold value, disk-head positioning was adversely affected. The atom matching time was usually about 10 ms per attempted match, but queries with all carbon atoms or many Rk definitions took considerably longer.

The most significant statistic, total search time, had a median value of 15.9 s, thus providing the high degree of interaction initially desired. The average total search time of 30.1 s is higher than the median as a result of the three searches with poor screenout. In such cases, the SSM does its best to inform the user of how the search is going and when it is expected to complete. The user can choose to cancel the search and refine the query or to continue with the search.

The combination of rapid searching and graphical query entry has resulted in the system being quite heavily used. Since COUSIN was released to the Upjohn research population, 130 scientists and support personnel have been trained in its use and have been performing between 130 and 200 substructure searches per month. This represents a large increase in usage over our previous batch-oriented intermediary-controlled system and indicates the benefits of an interactive graphics-based approach.

ACKNOWLEDGMENT

We thank Chemical Abstracts Service and the Basel Information Center for Chemistry for their help in obtaining a copy of the BASIC screen dictionary. We also thank Chemical Abstracts Service for helpful discussions involving their design for a substructure search machine using a network of minicomputers.[13]

REFERENCES AND NOTES

(1) Howe, W. J.; Hagadone, T. R. "Molecular Substructure Searching: Computer Graphics and Query Entry Methodology". *J. Chem. Inf. Comput. Sci.*, **1982**, *22*, 8-15.
(2) Ray, L. C.; Kirsch, R. A. "Finding Chemical Records by Digital Computers". *Science*, **1957**, *126*, 814-819.
(3) Lefkovitz, D. "Substructure Search in the MCC System". *J. Chem. Doc.* **1968**, *8*, 166-173.
(4) Rössler, S.; Kolb, A. "The GREMAS System, an Integral Part of the IDC System for Chemical Documentation". *J. Chem. Doc.* **1970**, *10*, 128-134.
(5) Adamson, G. W.; Cowell, J.; Lynch, M. F.; McLure, A. H. W.; Town, W. G.; Yapp, A. M. "Strategic Considerations in the Design of a Screening System for Substructure Searches of Chemical Structure Files". *J. Chem. Doc.* **1973**, *13*, 153-157.
(6) Feldman, A.; Hodes, L. "An Efficient Design for Chemical Structure Searching. I. The Screens". *J. Chem. Inf. Comput. Sci.* **1975**, *15*, 147-152.
(7) Graf, W.; Kaindl, H. K.; Kniess, H.; Schmidt, B.; Warszawski, R. "Substructure Retrieval by Means of the BASIC Fragment Search Dictionary Based on the Chemical Abstracts Service Chemical Registry III System". *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 51-55.
(8) Sussenguth, E. H. "A Graph-Matching Algorithm for Matching Chemical Structures". *J. Chem. Doc.* **1965**, *5*, 36-43.
(9) Ming, T. K.; Tauber, S. J. "*Chemical* Structure and Substructure Search by Set Reduction". *J. Chem. Doc.* **1971**, *11*, 47-51.
(10) Figueras, J. "Substructure Search by Set Reduction". *J. Chem. Doc.* **1972**, *12*, 237-244.
(11) Lynch, M. F.; Harrison, J. M.; Town, W. G.; Ash, J. E., Eds. "Computer Handling of Chemical Structure Information"; Macdonald: London, 1971; pp 73-74.
(12) Brown, H. D.; et al. "The Computer-Based Chemical Structure Information System of Merck Sharp and Dohme Research Laboratories". *J. Chem. Inf. Comput. Sci.* **1976**, *16*, 5-10.
(13) Farmer, N. A. "The Proposed Chemical Abstracts Service's Substructure Search System". Proceedings of the Technical Information Retrieval Committee of the Manufacturing Chemists Association, Arlington, VA, Aug 1977; McNulty, P. J., Smith, R. B., Eds.; Manufacturing Chemists Association: Washington, DC, 1977.

Evaluation of the Quality of Symposia Papers. Status Report on the Symposium on Photochemistry in Japan

AKIHIDE KITAMURA* and KUNIO OOHASHI

Department of Chemistry, College of Arts and Sciences, Chiba University, Yayoi-cho, Chiba 260, Japan

TATSUO ARAI and KATSUMI TOKUMARU

Department of Chemistry, the University of Tsukuba, Sakura-mura, Ibaraki 305, Japan

MASAYUKI YOSHIDA

University of Library and Information Science, Yatabe, Ibaraki 305, Japan

Received December 29, 1981

Papers presented at the annually held Symposium on Photochemistry in Japan were reviewed to ascertain which of these had subsequently been published. Certain research trends became evident in reading through the papers; in particular, it appeared that studies on photoreductions had been prevalent in the Symposium in recent years. From the investigation regarding the quantity and yield of the published papers it appears that a high proportion of the papers presented at the Symposium have been subsequently published, and it is concluded that the general level of quality of the papers is accordingly also high.

INTRODUCTION

In order to keep abreast of current developments, chemists actively engaged in research may scan regularly the contents of seven or eight journals devoted to their research speciality.[1] However, most chemists feel that "current" information from primary printed sources is not necessarily current, since it can take from 3 months to over 1 year before an article submitted to a journal is finally published.[2] Therefore, active research chemists tend to obtain pertinent "current" information at meetings and symposia; but while such information may be of value for current awareness, the quality of the information so obtained is another question. With this in mind, the present investigation took the annually held Symposium of Photochemistry in Japan and, using statistical analysis, scrutinized its status as revealed in the published literature.

RESEARCH TRENDS OBSERVED IN THE SYMPOSIUM ON PHOTOCHEMISTRY AND THE PUBLISHED LITERATURE

During the last 25 years photochemistry has grown rapidly into a major interdisciplinary field of research. Because of that trend, the Symposium on Photochemistry in Japan was first organized in 1960 and has been held every year between September and December since that date. As shown in Figure 1, the number of papers presented at symposia increased slowly from 1960 to 1970 and then stagnated in 1971-1975. The number began to grow again in 1976 and has been increasing ever since. As the annual Symposium has become one of the representative symposia in Japan, during the same period the number of papers on photochemistry cited in *Chemical Abstracts* has continued to increase steadily, suggesting that the field of photochemistry is still in a state of development.

The difference in pattern between the curves for (a) the papers presented at the Symposium and (b) published papers may be ascribed to the difference in the fields covered. The general literature on photochemistry includes a wide variety of photochemical studies, while the Symposium papers are limited to those related to pure chemistry. Any comparison of general research trends with those covered in the Symposium should be based on an identical field. Thus a start was made in this investigation by examining the growth of papers on photochemical reactions.

In Figure 2 is shown the difference in the growth curves between the published papers on the one hand and Symposium papers on the other. The latter show a maximum in 1972 and

0095-2338/82/1622-0186$01.25/0