

AD665397

AD

**EDGEWOOD ARSENAL
SPECIAL PUBLICATION**

EASP 400-8

**PROCEEDINGS OF THE WISWESSER LINE
NOTATION MEETING OF THE ARMY CHEMICAL
INFORMATION AND DATA SYSTEMS
PROGRAM 6-7 OCTOBER 1966**

edited by

James P. Mitchell (Deceased)

January 1968



**DEPARTMENT OF THE ARMY
EDGEWOOD ARSENAL
Technical Support Directorate
Edgewood Arsenal, Maryland 21010**

Distribution Statement

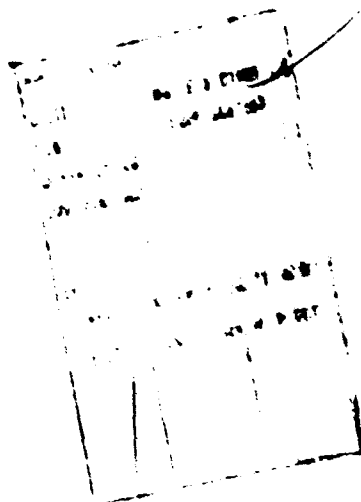
This document has been approved for public release and sale; its distribution is unlimited.

Disclaimer

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Disposition

Destroy this report when no longer needed. Do not return it to the originator.



EDGEWOOD ARSENAL SPECIAL PUBLICATION

EASP 400-8

PROCEEDINGS OF THE WISWESSER LINE NOTATION
MEETING OF THE ARMY CHEMICAL INFORMATION
AND DATA SYSTEMS PROGRAM 6-7 OCTOBER 1966

edited by
James P. Mitchell (Deceased)

Technical Data Coordination Office

January 1968

This document has been approved for public release
and sale; its distribution is unlimited.

Project 2P020401A727

DEPARTMENT OF THE ARMY
EDGEWOOD ARSENAL
Technical Support Directorate
Edgewood Arsenal, Maryland 21010

FOREWORD

The work described in this report was authorized under Project 2P020401A727, Chemical Information and Data System (CIDS). The report is a compilation of papers presented at the Wiswesser Line Notation Meeting of the Army Chemical Information and Data Systems Program at Edgewood Arsenal, Edgewood Arsenal, Maryland, on 6 and 7 October 1966.

Reproduction of this document in whole or in part is prohibited except with permission of the Commanding Officer, Edgewood Arsenal, ATTN: SMUEA-TSTI-T, Edgewood Arsenal, Maryland 21010; however, Defense Documentation Center and the Clearinghouse for Federal Scientific and Technical Information are authorized to reproduce the document.

Acknowledgments

Acknowledgment is made with gratitude to all participants for their contributions to this report and to Mrs. Grace Boddie, Chief of Research Support Management Office, Army Research Office, Durham, North Carolina, for her assistance in making arrangements for the participating attendees.

DIGEST

A significant application of Wiswesser line notation (WLN) is to enable computer discrimination among chemical compounds based on their characteristics as represented by symbolic code designations. These proceedings cover use and techniques of WLN by various industrial, military, and academic organizations. Areas of use discussed include registration of compounds, storage, and retrieval of structures in several collections, and generation of structural fragment codes for rapid structure and substructure searches. Other papers cover quick scan and symbols, a computer-generated open ended fragment code, permutations and classification numbers, the "Dot-Plot" computer program, and a partial algorithm for development of connection tables from WLN. A presentation on the formal analysis of notation systems is also made. Discussions in techniques include file maintenance and updating procedures, and machine management of small chemical data systems.

CONTENTS

	<u>Page</u>
WELCOME, COL Walter J. Davies, Director of Technical Support, Edgewood Arsenal.....	11
INTRODUCTION: CIDS OBJECTIVES, J. P. Mitchell, Technical Support Directorate, Edgewood Arsenal	13
USE OF THE WISWESSER LINE NOTATION AT THE SEARLE LABORATORIES: MOTIVATION AND STATUS, Dr. Howard T. Bonnett, G. D. Searle and Company.....	15
LOW-COST STORAGE AND RETRIEVAL OF ORGANIC STRUCTURES BY PERMUTED LINE NOTATIONS: SMALL COLLECTIONS, J. K. Horner, Stanford Research Institute	25
USE OF THE WISWESSER LINE NOTATION FOR REGISTERING COMPOUNDS, Charles E. Granito, Diamond Alkali Company...	35
FILE MAINTENANCE AND UPDATING PROCEDURE, Dr. Peter F. Sorter, Hoffmann-La Roche, Inc.....	39
QUICK SCAN AND SYMBOLS, Alan Gelberg, Diamond Alkali Company	43
AUTOMATIC GENERATION OF STRUCTURAL FRAGMENT CODES FROM THE WISWESSER LINE NOTATION FOR RAPID STRUCTURE SEARCHES, Carlos M. Bowman, Franc A. Landee, Mary H. Reslock, and Betsy P. Smith, The Dow Chemical Company	49
COMPUTER GENERATED OPEN ENDED FRAGMENT CODE, Ernest Hyde, Canadian Industries Ltd.	57
SUBSTRUCTURE SEARCHING ON NOTATIONS, George F. Fraction, Eli Lilly and Company	69
PERMUTATIONS AND CLASSIFICATION NUMBERS, Alan Gelberg, Diamond Alkali Company.....	81

	<u>Page</u>
SOME TECHNIQUES FOR THE MACHINE MANAGEMENT OF SMALL CHEMICAL DATA SYSTEMS, A. J. Barnard, Jr., W. C. Broad, and C. T. Kleppinger, J. T. Baker Chemical Company, and W. J. Wiswesser, Fort Detrick.....	85
THE "DOT-PLOT" COMPUTER PROGRAM, William J. Wiswesser, Fort Detrick.....	103
UPDATING PROGRAM FOR THE INDUSTRY LIAISON OFFICE, David E. Renard, Industry Liaison Office, Edgewood Arsenal....	117
UTILIZATION OF THE WISWESSER NOTATION IN CIDS, Clarence T. Van Meter, University of Pennsylvania.....	121
CONNECTION TABLES FROM WISWESSER LINE NOTATION: A PARTIAL ALGORITHM, George F. Fraction, Eli Lilly and Company, Justin C. Walker and Stephen J. Tauber, National Bureau of Standards.....	139
THE FORMAL ANALYSIS OF NOTATION SYSTEMS, James Munz, University of Pennsylvania.....	197
DISTRIBUTION LIST.....	203
DD FORM 1473 (DOCUMENT CONTROL DATA - R&D).....	211

ATTENDEES

<u>Name</u>	<u>Organization</u>
Dr. Aaron Addleston	Winthrop Laboratories
Dr. A. J. Barnard	J. T. Baker Chemical Company
Mr. Jerry Beveridge	Fort Detrick
Dr. John Bogusky	US Army Materiel Command, Washington, D. C.
Mr. Rick Bojar	University of Pennsylvania
Dr. Howard T. Bonnett	G. D. Searle & Company
Dr. Carlos Bowman	The Dow Chemical Company
COL Walter J. Davies	Edgewood Arsenal
Mr. Sylvan Eisman	US Army Frankford Arsenal
Mr. George Fraction	Eli Lilly and Company
Mr. Louis E. Garono	Edgewood Arsenal
Mr. Alan Gelberg	Diamond Alkali Company
Mr. Charles Granito	Diamond Alkali Company
Mr. Robert Hartman	Edgewood Arsenal
Mr. Lloyd A. Holly	Edgewood Arsenal
Mr. J. K. Horner	Stanford Research Institute
Mr. Ernest Hyde	Canadian Industries, Ltd.
Dr. David Jacobus	US Army Medical R&D Command
Miss Andrea John	Goodyear Tire & Rubber Company
Mr. Ronald Kent	University of Pennsylvania

<u>Name</u>	<u>Organization</u>
Dr. David Lefkovitz	University of Pennsylvania
Mr. Eugene C. Logue	Edgewood Arsenal
Mr. William Longenecker	Fort Detrick
Mr. Charles E. McKnight	US Army Munitions Command, Dover, New Jersey
Mr. Scot Mannion	US Army Materiel Command, Washington, D. C.
Dr. Boyd Mathers	National Science Foundation
Mr. J. P. Mitchell	Edgewood Arsenal
Mr. James Munz	University of Pennsylvania
COL Ernest A. Nagy	US Army Medical R&D Command
Dr. Warren Powell	Chemical Abstracts Service
Mr. T. W. Quigley	National Science Foundation
Mr. David E. Renard	Edgewood Arsenal
Mrs. Mary Reslock	The Dow Chemical Company
Mr. Edmund H. Schwanke	Edgewood Arsenal
Dr. Elbert G. Smith	Mills College
Dr. Peter F. Sorter	Hoffman-La Roche, Inc.
Miss Lucille H. Thomason	Canadian Industries, Ltd.
Dr. Clarence T. Van Meter	University of Pennsylvania
Mr. Peppino N. Vlannes	Office of Chief of Research and Development, Washington, D. C.

<u>Name</u>	<u>Organization</u>
Mr. Justin Walker	National Bureau of Standards
Mr. Lawrence Ware	US Army Medical R&D Command
Mr. William J. Wiswesser	Fort Detrick

WELCOME

Colonel Walter J. Davies
Director of Technical Support
Edgewood Arsenal

Good Morning. I am pleased to welcome you to Edgewood Arsenal. Your visit here is very meaningful to the Army's information program, and the welcome I wish to express is heartfelt indeed. I hope your stay here will be both pleasant and rewarding for you.

Your presence here affords the Chemical Information and Data System (CIDS) program people to take a good, hard look at the Wiswesser line notation. You who have been using it in industry, at the universities, and at other Government agencies are the experts. It has been established that the notation is well worth a critical examination by the Army. Here is where you can help, and I commend your response as evidenced by this gathering.

The need and the importance of an operating CIDS for the Army is well known to you. I need not dwell on it. I do wish to impress on each of you that your visit here will have contributed tangibly toward that end.

Once again, a very sincere welcome to Edgewood Arsenal. The cooperation of yourselves and your organizations is deeply appreciated. I am certain that the Army will benefit directly from these sessions, and I hope that they will be similarly beneficial to you.

Again, we are glad to have you with us. Mr. Mitchell and his people and Mr. Garono and my office will do anything while you are here to make your stay pleasant and enjoyable and also profitable. Mr. Mitchell would like to kick off now and make a few administrative announcements.

INTRODUCTION

CIDS OBJECTIVES

J. P. Mitchell
Edgewood Arsenal

CIDS, the Army Chemical Information and Data System, is an exploratory development project designed to investigate new techniques and the handling of chemical structures and associated data required for the support of, or produced by, on-going work by the Army. The immediate objective of CIDS is to demonstrate user needs and acceptance by 1 July 1967 with respect to logic, systems and programs, data, and information to establish what specific need exists for a future Army-wide system.

Perhaps the most fundamental phase of CIDS is that which involves the discrimination among chemical compounds on the basis of their structural characteristics, and endeavor in this phase to date has centered almost entirely among structural information gleaned from node connection tables. However, the potential value of the WLN as another, perhaps even better, source for some or all of this structural information has been kept constantly in mind and the time has now come when this potential can be explored in depth.

In essence, then, the main objective of this meeting is to get the ball rolling and we could think of no better way to do this than to solicit the kind willingness of you experts to meet with us and relate some of your experiences with the notation.

In addition to file building, we are also, of course, very much interested in experiences utilizing the notation for search purposes and in observations pertaining to the present state of development of the notation in terms of its structural discrimination capabilities.

At one place in the program, we describe some typical CIDS structure queries and specifically in this vein we hope to have the following questions answered:

1. Use of WLN for registration?
2. Use of WLN as screens?
3. Use of WLN for file classification?

4. Use of WLN as compact storage form?
5. Use of WLN for search of both whole structure and sub-structure?
6. Use of WLN as machine language as contrasted to an alpha numeric language?
7. Use of WLN as an output language?

Before we begin, I would like to introduce my cochairman again, Mr. Gelberg, who was quite instrumental in setting up this meeting.

We will now begin with Dr. Bonnett of G. D. Searle and Co.

Dr. Bonnett:

USE OF THE WISWESSER LINE NOTATION
AT THE SEARLE LABORATORIES.
MOTIVATION AND STATUS

Dr. Howard T. Bonnett
G. D. Searle and Company

I wish to express my appreciation to the Army for arranging this symposium of those using the Wiswesser line notation in their information systems, and for the privilege of attending it.

Before getting into the subject of my discussion, I would like to make three general comments:

1. It has always been an interesting fact to me that, until recently, in the case of the Wiswesser notation, use of the notation was by others than the designer.
2. While Addelston and I were the first to put the Wiswesser notation to actual use, others since have gone farther in the development of programming than have we.
3. To me, it is amazing that industrial organizations are adopting the Wiswesser notation for use in operating systems, in the absence of an "official" published manual reflecting the revisions that have taken place since the original 1954 published manual.

Most of my working career has been spent in the area of patent work in the drug industry. In this type of activity, one is made acutely conscious of previous knowledge and experience. Patents are granted for the new. To determine what is new means one must know what is old, one must find the old. Finding the old presents problems, as any who attempt it learn. Thus, it is quite natural that I should have an acute interest in indexing and retrieval, and especially of chemical structures.

Retrieval can be troublesome. Let me illustrate by a few actual cases that happened in our laboratories. These are simple situations, but since a sizeable proportion of our staff are young college graduates, the turnover is rapid and so these simple situations continually face us.

In figure 1 is a compound that our searcher reported she could not find in the literature.

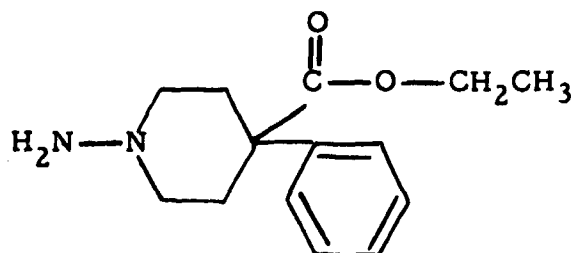


Figure 1. Isonipecotic Acid, 1-Amino-4-phenyl, Ethyl Ester

Sometime later, as is often the case when this sort of failure occurs, our "client" told us the compound was reported in Chemical Abstracts. On checking, it was found readily in the molecular formula index. This compound, obviously, is a derivative of a 4-piperidine carboxylic acid. If, however, you were looking for this entry you would not find the compound because the compound had been indexed under its trivial name as an isonipecotic acid derivative.

The compound in figure 2 illustrates a different problem. In this case, our "client" was looking for process instructions to make the compound shown.

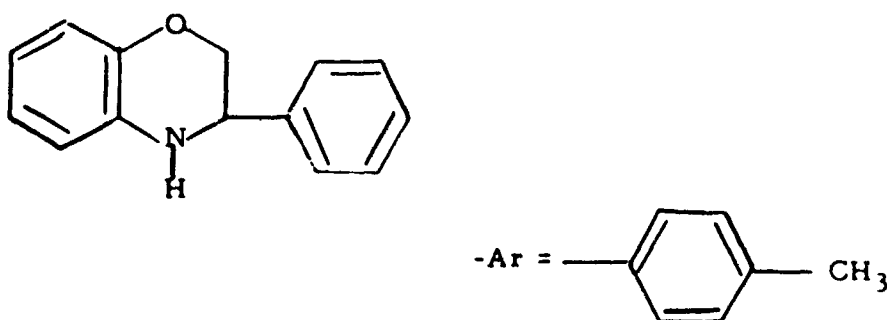
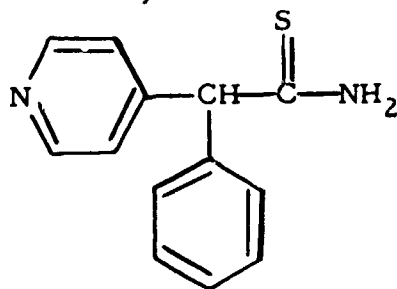


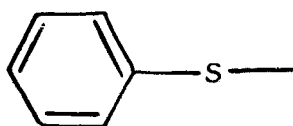
Figure 2. 1,4-Benzoxazine, 3-Phenyl-dihydro

We reported that we could not find this compound in either Beilstein or Chemical Abstracts. Again, our "client" returned later and said that the compound was shown in Elderfield. Sure enough, 1,4-benz-oxazine with an aryl group attached in the 3- position was indexed. The phenyl derivative, which we were seeking, was not specifically shown. When we checked the original reference (which was not in our library) we found it to be the tolyl derivative that was reported. Chemical Abstracts was not at fault here, nor was the searcher wrong in her report. The point here is that a search for a specific compound sometimes turns into a generic search. In this case, presumably, the procedure for making the tolyl derivative would work for the phenyl.

Figure 3 illustrates a rather subtle problem. We had no difficulty in determining that the root compound might be 4-pyridine-acetamide, but the second portion of the index name was misinterpreted to mean a phenyl with a sulfur directly attached on the alpha position of the acetic acid residue.



4-Pyridineacetamide, α-Phenylthio



α-(Phenylthio)-

Figure 3. Structures Illustrating a Subtle Indexing Problem

While with a previous employer, we attempted to maintain an index of steroid compounds utilizing systematic nomenclature. This proved to be a particularly frustrating type of activity, because the file was maintained by clerical personnel to whom the difference, for example, between pregnene and pregnane was quite obscure. Consequently, this file was never in sequence and, therefore, was untrustworthy. We attempted to get around this problem with a system based on a molecular formula

approach, illustrated in figure 4. In this technique, any hetero atom in a compound was related to, and treated as, an oxygen substituent for classification purposes, so that the compound on the right, even though it contained a bromine and three oxygens, was for purposes of classification put in the category of $C_{21}H_{29}O_4$. This approach simply avoided nomenclature.

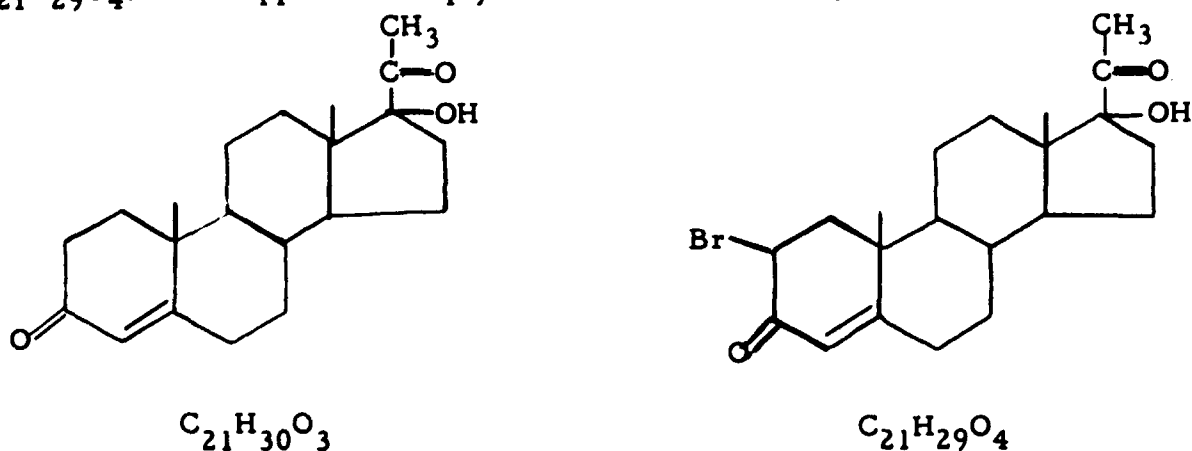


Figure 4. Classification Using a Molecular Formula Approach

The structures in figure 5 illustrate a problem that is inherent in nomenclature as practiced by chemists.

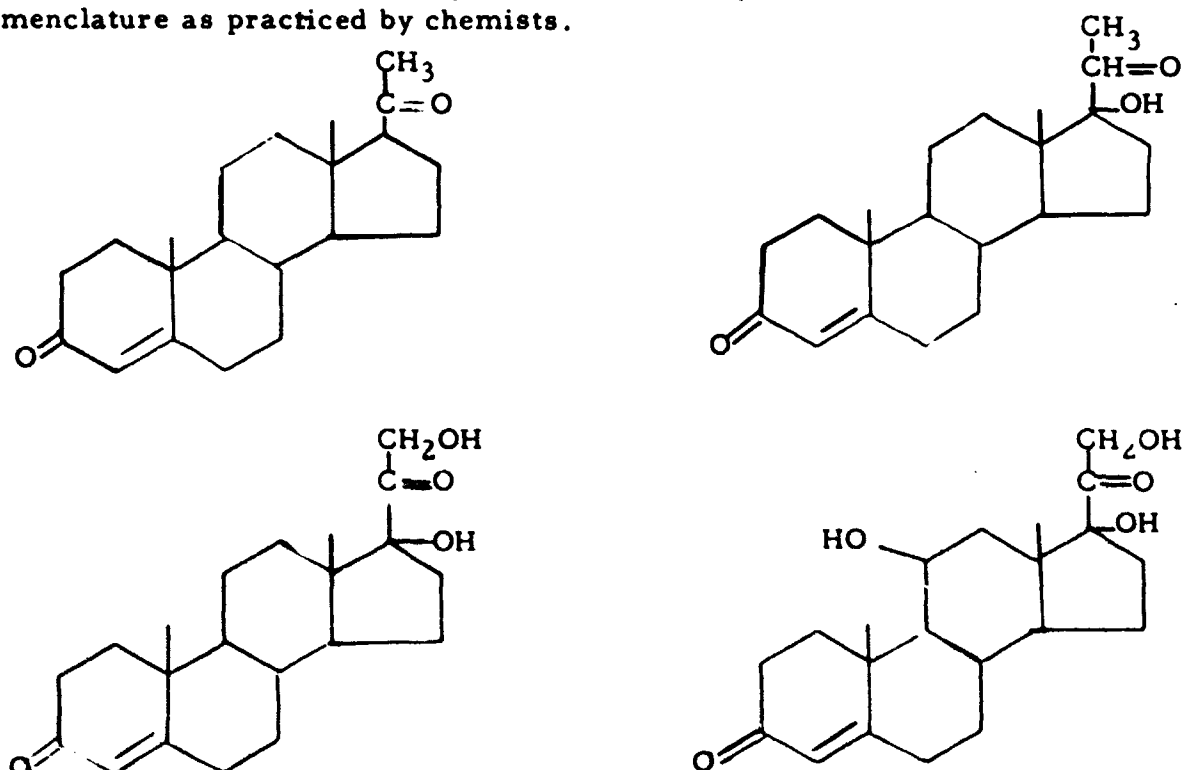


Figure 5. Structures Illustrating Problems Inherent in Nomenclature

The formula in the upper left depicts progesterone. In succeeding structures, hydroxyl groups have been added; at the lower right is the structure of hydrocortisone. Now, at what point does one cease to regard the substance as a progesterone derivative and at what point does it become a corticoid?

As these cases show, a common thread runs through our search problems. This thread has a name, NOMENCLATURE.

The problems inherent in trying to index chemical structures with the techniques available at the time, namely, the structural formula, the systematic nomenclature, and the molecular formula, were so frustrating that I began to wonder if we could find a means of recording structures in a machinable language and throw onto the machine the burden of organizing and maintaining in organized fashion, a file of structures. At about this time, the early 1950's, publicity on the efforts of a volunteer committee which had undertaken to compare four notation systems began to appear. I tried to evaluate the relative advantages and disadvantages of each of the four techniques as reflected by the experience of this committee; it appeared to me that the notation due to Wiswesser merited exploration, and so my interest in it was aroused.

It was not until I joined Searle that I had an opportunity to explore the utility of the Wiswesser notation as an indexing and retrieval tool in an industrial information setting. A friend in another company was also engaged in designing a system; it was interesting to me that both of us chose an identical basic approach. Both of us wanted to record the structure on a machinable card, and both of us wanted a functional group field. My friend chose to put the structure on the machinable card by hand-drawing it. I chose the notation, recorded as a string of characters available in accounting machinery. Obviously, my friend, in order to use his file, had to have a functional group field, because that was his only approach. In my case, I found that the notation with supplementary codes, which could be sorted by machinery and printed to produce "telephone directory" type of indexes, resulted in indexes having considerable search power, which reduced the need for a functional group field. Also, at about this time, the computer began to loom over the horizon. It was recognized that this machine was capable of searching a string of symbols for patterns of symbols (i.e., functional groups), and this, too, would reduce the need for a card-based functional-group search facility. I should add, however, that I did attempt to design a functional group code, but the versions I tested were not completely satisfactory. I do not mean to imply by my remarks that I do not value the functional group facility; I do, and I still intend to have such capability as a part of our system.

As a result of the operating experience, which those of us using the notation have accumulated, it has become apparent that some of the original rules could be modified to produce superior indexing capability. This is illustrated by the three bicyclic heterocyclic structures in figure 6.

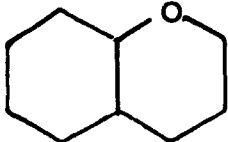
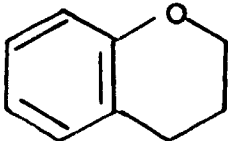
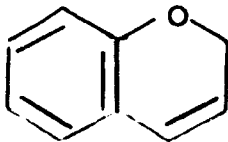
	<u>Old</u>	<u>Revised</u>
	T6T6T BOJ	T66 BOTJ
	T6T6 BOJ	T66 BOT&J
	T66 BO DUJ	T66 BO CHJ

Figure 6. Bicyclic Heterocyclic Structures
Illustrating Effect of Rule Design

Beside each structure appears the notation according to the 1954 edition of the Wiswesser manual, and the notation according to the revised rules. These three closely related structures would appear in three different locations in an alphabetized file of old rule notations, whereas, the revised rule provides identical notations through the first six spaces, and thus, will tend to group these structures more closely together. Yet, I am sure in designing the 1954 rules, it seemed perfectly reasonable to apply the ring saturation symbol (i.e., "T") immediately after the ring size numeral. In practice, however, it has been found that a superior index is achieved by subordinating the degree of ring saturation relative to heterocyclic nature, for example. This, and other rule revision, has come about on the basis of actual operating experience in several different operating systems.

The very simple structures of figure 7 also illustrate the tendency of the Wiswesser line formula notation to group like structures together.

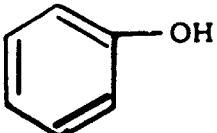
CH_3OH	Methanol	Q1
$\text{CH}_3\text{CH}_2\text{OH}$	Ethanol	Q2
$\text{CH}_3(\text{CH}_2)_2\text{OH}$	n-Propanol	Q3
$\text{CH}_3\text{CHOHCH}_3$	Isopropanol	QY
	Phenol	QR

Figure 7. Simple Structures

The Searle files now include approximately 125,000 entries in the Wiswesser notation. These are divided as approximately 30,000 representing compounds from our Research Laboratories, and 95,000 selected from the literature, primarily steroid in character. We have found the cost of input and generation of the "telephone directory" index to be approximately 25¢ per structure, starting with the structural formula, and winding up with the first printed index. These cost figures were reported in the NAS-NRC study on notation systems and have been confirmed by numerous workers since.

In figure 8 is shown a portion of a page from our "telephone directory" type of index. The family group characteristic of the index is obvious. Each line represents a chemical compound and identifying information about the compound. The information given is:

1. The Wiswesser notation.
2. Identification of the laboratory where the compound was made.
3. Notation prefix.
4. A four-digit classification code, each digit being a separate code, which serves as a useful screen in searches.
5. Country code and type of literature code.
6. Literature reference.
7. Formula sheet number.

T C667 ANV IN MM&TJ EG	2	SAN 3 3750 82	8	3280117 40523
T C667 ANV IN MN&TJ EG M	2	SAN 3 3751 82	8	3280117 40523
T C667 ANV IN MN&TJ EG MVOIR	2	SAN 3 3770 82	8	3280117 40523
T C667 ANV IN MO&TJ EE	2	SAN 3 3750 82	8	3280117 40523
T C667 ANV IN MO&TJ EG	2	SAN 3 3750 82	8	3280117 40523
T C667 ANV IN MO&TJ E02	2	SAN 3 3250 82	8	3280117 40523
T C667 ANV IN MO&TJ EXFFF	2	SAN 3 3770 82	8	3280117 40523
T C667 ANV IN MS&TJ EG	2	SAN 3 3750 82	8	3280117 40523
T C667 ANV IN MS&TJ PG	2	SAN 3 3750 82	8	3280117 40523
T C667 ANV IN MS&TJ P04	2	SAN 3 3550 82	8	3280117 40523
T C667 ANV IN MS&TJ PSY	2	SAN 3 3550 82	8	3280117 40523
T C667 ANV IN MS&TJ FXFFF	2	SAN 3 3770 82	8	3280117 40523
T C657 BN&TJ I &WS01&01	2	KMARX 3 3110 44	8	3280117 40523
T C667 BN&TJ I- AT5NTJ	21	CDS 3 4220 94	8	3280117 40523
T C667 BN&TJ I- AT6N DOTJ	21	CDS 3 4230 94	8	3280117 40523
T C667 BN&TJ I- AT6NTJ	21	CDS 3 4220 94	8	3280117 40523
T C667 BN&TJ I- AT7N DNTJ D	21	CDS 3 4230 94	8	3280117 40523
T C667 BN&TJ IVN2&2	21	CDS 3 3230 94	8	3280117 40523
T C667 BN&TJ BVINI&I	3K	THO 3 3230 22	8	3280117 40523
T C676 AN IM&TJ E	2	GEI 3 3220 82	2	624537 23928
T C676 AN IM&TJ E	2	GEI 3 3220 82	8	3281419 40527
T C676 AN IM&TJ EG	2	GEI 3 3730 82	8	3281424 40530
T C676 AN IM&TJ EG	2	GEI 3 3730 82	8	3281419 40527
T C676 AN IM&TJ E01 F01	2	GEI 3 3730 82	8	3281424 40530
T C676 AN IM&TJ E01 F01	2	GEI 3 3240 82	8	3281419 40527
T C676 AN IM&TJ F	2	GEI 3 3240 82	8	3281424 40530
T C676 AN IM&TJ F	2	GEI 3 3220 82	8	3281419 40527
T C676 AN IM&TJ F	2	GEI 3 3220 82	8	3281424 40530
T C676 AN IM&TJ PG	2	GEI 3 3730 82	8	3281419 40527
T C676 AN IM&TJ PG	2	GEI 3 3730 82	8	3281424 40530
T C676 AN IM&TJ F01	2	GEI 3 3230 82	8	3281419 40527
T C676 AN IM&TJ F01	2	GEI 3 3230 82	8	3281424 40530
T C676 AN IMV&TJ	2	GEI 3 3230 82	8	3281419 40526
T C676 AN IMV&TJ	2	GEI 3 3230 82	8	3281424 40529
T C676 AN IMV&TJ E F	2	GEI 3 3230 82	8	3281419 40526
T C676 AN IMV&TJ E F	2	GEI 3 3230 82	8	3281424 40529
T C676 AN IMV&TJ EG	2	GEI 3 3740 82	8	3281419 40526
T C676 AN IMV&TJ EG	2	GEI 3 3740 82	8	3281424 40529
T C676 AN IMV&TJ EG PG	2	GEI 3 3750 82	8	3281419 40526
T C676 AN IMV&TJ EG PG	2	GEI 3 3750 82	8	3281424 40529
T C676 AN IMV&TJ E01 F01	2	GEI 3 3250 82	8	3281419 40526
T C676 AN IMV&TJ E01 F01	2	GEI 3 3250 82	8	3281424 40529
T C676 AN IMV&TJ F	2	GEI 3 3230 82	8	3281419 40526
T C676 AN IMV&TJ F	2	GEI 3 3230 82	8	3281424 40529
T C676 AN IMV&TJ PCN	2	GEI 3 334& 82	8	3281419 40526
T C676 AN IMV&TJ PCN	2	GEI 3 334& 82	8	3281424 40529
T C676 AN IMV&TJ PG	2	GEI 3 3740 82	8	3281419 40525
T C676 AN IMV&TJ PG	2	GEI 3 3740 82	8	3281424 40529
T C676 AN IMV&TJ F01	2	GEI 3 3240 82	8	3281419 40526
T C676 AN IMV&TJ F01	2	GEI 3 3240 82	8	3281424 40529

Figure 8. Telephone-Directory-Type Index

In undertaking our work, I made an assumption that others, as you will hear, are making good. I assumed that if we could reduce a chemical structure to a machinable language, we should be smart enough to program machines to do many things with this record. This is proving to be the case. For example: one can search the record for symbol patterns; one can calculate the molecular formula; one can do much to check the validity of the notation; one can generate a keyword-in-context functional group index; one can generate fragmentation codes that can be handled as a coordinate-concept type of index; one can generate a type of connection table, either represented in terms of fragments in the notation or as atoms in the compound; and so on.

As you may have guessed from the preceding discussion, I have long been interested in a technique of representing chemical structures that would offer flexibility in manner of use; i.e., both manual and mechanical. It has been my experience that a high ratio of searches can be run quickly and effectively in a suitable manually usable index. At the same time, some searches are not feasible to conduct in manually usable indexes and, consequently, access to suitable equipment to assist the searching is desirable. It was these capabilities that I thought I saw in the Wiswesser notation. I have been pleased by the performance of the notation in our hands as an indexing and retrieval tool.

LOW-COST STORAGE AND RETRIEVAL OF ORGANIC STRUCTURES BY PERMUTED LINE NOTATIONS: SMALL COLLECTIONS

J. K. Horner
Stanford Research Institute

A sophisticated, low-cost storage and retrieval system is described for a small collection of chemical structures. About 3000 structures from the Stanford Research Institute (SRI) files were coded according to the Wiswesser line notation, a computer program was written and a permuted line-notation index was generated; all for approximately \$700.00.

Earlier reports have shown that permuted chemical line notations in tabulated lists can be used to rapidly locate specific compounds, classes of compounds having similar ring systems, and compounds having the same functional group.¹⁻⁴ Time and cost data on computer-generated indices are available for large collections of chemical structures³ and time data on key-punch-generated indices for small collections are available.² The object of this communication is to report the experiences of the Life Sciences Research Area of SRI in collecting and storing at a low cost a small collection of chemical structures.

BACKGROUND.

In 1963, it became evident to scientists at SRI that a central file of data on organic structures was needed to replace the antiquated molecular-formula files kept in each section. An organic chemist with neither background nor training in data storage and retrieval was assigned the task of assessing the current methods of chemical-data storage and of adapting one to fit the needs of SRI.

Definitive information in this field proved scanty until the publication in 1964 of the "Survey of Chemical Notation Systems,"⁵ a report of the Committee on Modern Methods of Handling Chemical Information. This comprehensive report, plus personal contacts with industrial and governmental personnel, indicated that there existed a lack of sophistication in chemical-data handling in a majority of governmental agencies and industrial concerns. Those groups having well-conceived chemical-data systems usually used punched or notched cards with fragmentation codes, topological codes, or chemical line notations. The codes were usually searched by hand or by means of a computer.

Certain requisite goals were set as a basis for evaluating the existing data programs while searching for the ideal structure retrieval system for SRI. The input procedure was not to require special or expensive equipment normally not found in an automatic data-processing section, and the coding, punching, and preliminary manipulations were to be handled by technicians. With this ideal system, one could retrieve from the files one specific compound, all compounds having specific ring systems in common, all compounds related by specific functional groups, and all compounds having a specified relation between groups. Most important, no machine should separate the literature chemist from the data files. This latter requirement was important because the computer section and the chemistry department are physically distant and also because a machine cannot browse and select interesting structures that do not fit the search requirements. This browsing can be quite useful, as anyone who has ever looked up a compound in a formula or name index knows. A machine cannot browse and neither can a literature chemist if he is separated from his chemical data.

The topological codes and the fragmentation codes were rejected because they required that the files be searched by computer for any semblance of sophistication. The line notations could not be encoded by technicians and some notations used symbols not found on automatic data-processing machines. Most line codes were searched by computer and none could search for spatial relationships. The Wiswesser line notation⁶ did use symbols familiar to chemists and rejected symbols not found on automatic processing machines. The most appealing use of this notation was described in 1964 by Sorter and co-workers¹ at Edgewood Arsenal. This group permuted the notation lists, then printed the permutations and bound them into volumes. The Wiswesser encoding still could not be done by a technician and all spatial relationships could not be found* but this system now utilized the computer only to prepare the dictionarylike lists which then allowed the chemist to make structure searches at his desk and also permitted him to browse. A visit with the Edgewood group indicated that over 60,000 compounds were in their system and searches were being successfully completed with a minimum of effort. This group used a revised version of Wiswesser's line notation[†] and offered to supply their computer program at no cost. SRI decided to use the Wiswesser line notation with permuted notational lists.

* Personal communication with Mr. Ernest Hyde of Canadian Industries, Ltd., has shown that substructures and spatial relationships can be searched by computer-generated fragments of the Wiswesser notation.

** Wiswesser, W. J. A Line-Formula Chemical Notation, as revised by E. G. Smith, will be published by McGraw-Hill in 1967.

STRUCTURE GATHERING AND CODIFICATION.

The project technician gathered from existing files, reports, and notebooks the published, nonsecret and nonproprietary structures that had been synthesized by SRI chemists. The structures and physical data were handprinted onto SRI New Compound Report Sheets (see figure). These pre-printed forms were on Strathmore Simulator paper, which could be reproduced economically on an available Pease Diazo Ammonia Developer. The technician made two copies of each sheet, then prepared two indexes; a Molecular Formula Index and an Accession Number Index. The primary New Compound Report Sheets were filed in a different location in case the copies were ever lost or damaged.

The structural line coding was done by the author who learned the Wiswesser line notation from the revised manual kindly supplied by Dr. E. G. Smith of Mills College. An encoding rate of 125 to 150 structures per hour was possible after 2 to 3 months of part-time study. Of the 5000 structures ultimately encoded, approximately 10 were sufficiently complex to necessitate encoding assistance from Professor Smith. The line notations were entered into the Accession Number Index directly under the structures. The notations were checked once later and poor penmanship and careless errors were corrected. Because only one person at SRI knew the code, it was not always possible to correct the errors resulting from a misunderstanding of the rules. Some 54 encoding mistakes were discovered later, after the line codes were permuted and printed.

AUTOMATIC DATA PROCESSING.

A computer program for permuting the notations was obtained from Edgewood Arsenal. This program was originally written in COBOL language for the UNIVAC II, then modified for the HONEYWELL 400 in EASY II language. The HONEYWELL language and the ALGOL 60 language of the SRI BURROUGHS B-5500 were not compatible and a direct translation, though perhaps possible, would not fully utilize the capabilities of the more sophisticated BURROUGHS machine. As the computer problem was well defined, a new program was written for the BURROUGHS B-5500 in the source language of EXTENDED ALGOL 60.

The combined card punching and verification times were 100 per hour when the SRI punch operators had to turn the pages of the Accession Number Index to read each new line code. When the codes were transferred from this accession file to Punch Card Coding Forms, the operators then punched and verified 198 cards per hour. The punched card inputs were arranged thus:

SR - _____ SRI Labs No. _____	Molecular Formula _____ 19 _____		
Stanford Research Institute NEW COMPOUND REPORT SHEET			
CHEMICAL NAME _____			
STRUCTURE:	Molecular Wt _____ mp bp Established: _____ Observed: _____ Amount submitted: _____		
Remarks (Toxicity, literature ref., stability, special handling, etc.)			
Prepared by	Notebook ref.	SRI Project No.	Project compound No.
Test restrictions			

Figure. Preprinted Form on Strathmore Simulator Paper Used to Report New Compounds

Punch Card 1:

Columns 6 - 10 Identification number and/or letters

Columns 21 - 79 Wiswesser notation

Column 80 Trailer card indication

Punch Card 2 (Optional):

Columns 1 - 37 Continuation of card 1

The program was written as three sections (A, B, and C). The 2808 punched cards used as input created 20,294 entries or 7.3 lines per card. Program A (4.4 min): With this main program, the 2808 cards were read into the computer and notations were scanned and permuted on the desired elements and functional groups. The QUICK SCAN was formed and the lines were edited into a form suitable for sorting. The entries were then written on magnetic tape. Program B (12.2 min): This COBOL program sorted the 20,294 permutations into alphabetical order and wrote them onto a second tape.* Program C (4.2 min): This program did the final editing of the permuted and sorted entries. This final tape was then checked to see if the entry needed one or two lines of output. The page headings and all entries were written onto a printer backup tape. The total computer processing time was 20.8 min. Program B has now been eliminated and Programs A and C combined with the ALGOL sorter, cutting the total processing time from 21 to approximately 15 min.

The tape from Program C was printed onto paper on a UNIVAC 1004 card processor. There were 132 characters per line and the print-out sheets were arranged thus:

Columns 1 - 4 Identification

Columns 5 - 6 Blank

Columns 7 - 18 QUICK SCAN area

Columns 19 - 20 Blank

Columns 21 - 132 Permuted Wiswesser notation

Column 77 Indexed symbol

* The very fast ALGOL sort routine was not available when this program was written. The new routine will perform the sorting operations in approximately one-half of the time.

TIME-COST DATA.

The cost of preparing the permuted index of 2808 cards from the SRI collection was 26¢ per compound. The total cost of \$733.50 (shown below) included the cost of writing the initial program.

Encode 2808 structures (125/hr)	22.4 hr	\$112.00
Writing program		396.00
Punch 2808 cards (150/hr)	19.0 hr	76.00
Verify 2808 cards (200/hr)	14.0 hr	57.00
Permute 2808 lines	4.4 min	13.15
Alphabetize 20,294 lines	12.2 min	36.80
Edit 20,294 lines	4.2 min	12.55
Print 21,539 lines (500 lines/min)	43.0 min	<u>30.00</u>
Total cost		\$733.50

Cost per structure, $\$733.50/2808 = 26\text{¢}$

As this need be done only once, the annual updating should prove rather inexpensive. Our figures show that 400 new compounds could be encoded, the line notations punched into cards and the new permutations merged with the existing system to generate a new index for about \$61.00. These structures, therefore, could be put into the index for 15¢ each.

The experiences of SRI indicate that companies, universities, and others with small collections of chemical structures and data can now file and retrieve information in a sophisticated manner at reasonable cost.

The four collected volumes of Organic Syntheses were encoded next for personal use. The 1850 line notations generated 7885 lines of permuted output or 4.3 lines per card. Time and cost data for preparing an index of these structural notations are as follows:

Encode 1850 structures (125/hr)	14.8 hr	\$ 75.00
Punch 1850 cards (294/hr)	6.3 hr	25.40
Verify 1850 cards (617/hr)	3.0 hr	12.00
Permute 1850 lines	1.7 min	5.20
Alphabetize 7885 lines	4.7 min	14.10
Edit 7885 lines	1.6 min	4.80
Print 8368 lines (500 lines/min)	17.0 min	<u>11.00</u>
Total cost		\$ 147.50

Cost per structure, $\$147.50/1850 = 8\text{¢}$

The card punching and verification times and the costs were less than before because the notations were entered by the encoder onto 30-line coding forms and the Organic Syntheses notations in general were much shorter than the SRI notations. The low cost of 8¢ per compound for generating this permuted index was also possible because the cost of writing the computer program had already been paid. The combined time and cost data for the SRI and Organic Syntheses compounds are shown as follows:

Writing program		\$ 396.00
Encode 4658 structures	37.2 hr	187.00
Punch 4658 cards	25.3 hr	101.40
Verify 4658 cards	17.0 hr	69.00
Permute 4658 lines	6.1 min	18.35
Alphabetize 28,179 lines	16.9 min	50.90
Edit 28,179 lines	5.8 min	17.35
Print 30,907 lines	60.0 min	<u>41.00</u>
Total cost		\$ 881.00

Cost per card, $\$881.00/4658 = 19\text{¢}$

UTILIZATION OF INDEX AND CONCLUSION.

The SRI Permuted Index has been used to search generically for ring systems and functional groups and to retrieve specific compounds. During these searches, 54 encoding errors were detected. Most of these errors were due to carelessness. The remainder of the errors were caused by the author's faulty memory of the rules. This failure to retain certain rules is especially serious when one encodes only once a week or perhaps only once a month. Though 1.9% encoding errors did occur, to our knowledge no structures were "lost," perhaps because some encoding errors were serious in which case the line notation was indexed in a very strange place and thus was quite noticable. Alternatively, the error was trivial, in which case the notation was indexed either directly before or after the correct location, so again, the error was easily detectable. A computer program designed to calculate the molecular formula from the Wiswesser notation has been written.* The calculated formula can then be checked against the correct molecular formula either by hand or by machine. A Checker program such as this should do much to allay the encoding problems of part-time chemical data storage and retrieval workers. A looseleaf notebook containing the structures and notations of the SRI cyclic ring systems was prepared at SRI. This ring system reference book was used to determine quickly whether general cyclic structures were in the index and also it was used as an encoding aid.

Over 5000 compounds were encoded and only 10 structures proved sufficiently troublesome to make outside assistance necessary. Of the first 2808 structures, 54 were incorrectly encoded. These statistics should refute the notion that the Wiswesser line notation is difficult to learn and use. The relative ease with which a novice can learn the Wiswesser system is in part due to Mr. Wiswesser's retention of known chemical symbols and the clever use of mnemonics. It is also due in part to the efforts of Dr. Smith in improving the original notation. Assisting Dr. Smith in evaluating and improving the Wiswesser notation are a group of users known as the Chemical Notation Association. These 14 chemical workers all have encoded at least 5000 structures and are constantly using and improving the notation. News letters are circulated and formal meetings are held to insure that the notation will keep pace with chemical advancements and that problems which arise will be promptly discussed and solved. At least 16 structure files coded according to the Wiswesser line notation are being maintained. These files contain over one-half million compounds.

* Personal communication with Dr. C. M. Bowman of the Dow Chemical Co., Midland, Michigan.

ACKNOWLEDGMENTS.

The author would like to thank Mr. D. A. Kerr for preparing the SRI Molecular Formula and Accession Number Indices and also to thank Mr. William S. Duvall of the SRI Computer Section for writing the fine ALGOL program.

LITERATURE CITED.

1. Sorter, P. F., Granito, C. E., Gilmer, J. C., Gelberg, A., and Metcalf, E. A. J. Chem. Doc. 4, 56 (1964).
2. Granito, C. E., Gelberg, A., Schultz, J. E., Gibson, G. W., and Metcalf, E. A. J. Chem. Doc. 5, 52 (1965).
3. Granito, C. E., Schultz, J. E., Gibson, G. W., Gelberg, A., Williams, R. J., and Metcalf, E. A. J. Chem. Doc. 5, 229 (1965).
4. Gelberg, A. J. Chem. Doc. 6, 60 (1966).
5. Survey of Chemical Notation Systems. National Academy of Sciences National Research Council Publication 1150. Washington, D.C. 1964.
6. Wiswesser, W. J. A Line-Formula Chemical Notation. Thomas Y. Crowell, Co. New York, N.Y. 1954.

USE OF THE WISWESSER LINE NOTATION FOR REGISTERING COMPOUNDS

Charles E. Granito
Diamond Alkali Company

The registration of chemical compounds may be considered to involve two steps: (1) determining whether a given compound is already in the system, that is, has an address; and (2) assigning an address to compounds that are not in the system.

The address may be alphabetical, numerical, or a combination of the two. Chemical Abstract's registry system involves strictly numerical addresses; e.g., 114,000. Most organizations, however, have an alphabetic prefix for their numbers; e.g., DAC-2787.

In order to determine whether a given compound is already in the system, one must compare its structure, or a representation thereof, to all of the other structures present in the system. The most popular ways of doing this are via: (1) the structure itself, (2) nomenclature, (3) molecular formulas, (4) chemical notations, and (5) connection tables.

The comparison of structures per se is useful only for small files. It is very time consuming, especially for closely related compounds.

Nomenclature, although widely used, leaves much to be desired when it comes to comparing compounds. The numerous rules, their complexity, and inconsistencies have been discussed many times. Suffice it to say that Chemical Abstracts, long a nomenclature protagonist, has decided against using nomenclature as the basis for registration.

The use of molecular formulas is limited in that it is only a screen that leads you back to a manual operation. This is because many compounds may have the same molecular formula (see for example any formula index from Chemical Abstracts). Despite this limitation, it is still the most widely used approach because most organizations add few compounds per week. Consequently, the total cost of registration is hidden; however, since the average chemist earns 5¢ to 10¢ per minute, the cost per item could be significant. When one considers adding hundreds or thousands of compounds per week, the impracticality of this approach becomes obvious. It should also be remembered that in using the molecular formula approach you gain only registration. (As will be shown later, the line notation provides additional benefits.)

Although a line notation file could be used to advantage for manually registering compounds, I will restrict my discussion today to machine methods. For large files that have rapid growth rates (such as CIDS anticipates) one must consider automatic registration.

The two main automatic registration procedures being used are: (1) chemical notations (for example, the Wiswesser line notation for the Industry Liaison Office file) and (2) connection tables (for example, the new Chemical Abstracts registry system).

I will not present any details concerning the use of connection tables today but will, instead, speak about the Wiswesser line notation. You may wish to keep in mind, however, the fact that Chemical Abstracts revealed at a recent national ACS meeting that their overall costs to register a new compound is about 88¢ per compound.

All of my experience concerning the registering of compounds via the Wiswesser line notation was gained here at Edgewood while I was a member of the staff of the Industry Liaison Office. We became interested in this approach when we passed the 50,000 structure mark and were receiving "new" structures at the rate of about 2000 per month. We decided to extend the use of the line notations that we were preparing as the basis for the chemical information retrieval program.

The Wiswesser chemical line notation is:

1. An unique and unambiguous method of depicting chemical structures.
2. A representation that can be processed by unmodified automatic data processing equipment.
3. Concise (the average length for a file of over 90,000 structures has been determined to be 16.5 symbols, including the space as a symbol).
4. Economical (we estimated input costs to be 10¢ per compound).

These attributes combine to make the notation an ideal means of registering compounds.

The computer times for one of our first duplication check runs are shown as follows:

	<u>Minutes</u>
1772 "New" items, card-to-tape	5
Sort "new" items	9
Duplicate check (against 84,716 items) and updating of master	18
Sort duplicates	10
Print list of duplicates	<u>8</u>
Total	50

The next logical step was to let the computer assign unique numbers. This has now been accomplished.

A registration run for some 7000 notations, against a master file of > 36,000 unique notations, required 90 min of computer time. This includes: card-to-tape input, sorting, checking, registration, updating, and print-outs. The computer cost (at \$75.00/hr) comes to about 1.6¢ per compound.

The cost from structure through registration is, therefore, 20¢ per compound.

It is believed that the above cost could be further lowered through the use of an automatic checker program such as developed by Dow.

The Wiswesser chemical line notation appears to be the best method of registering compounds that is available today.

FILE MAINTENANCE AND UPDATING PROCEDURE

Dr. Peter F. Sorter
Hoffmann-La Roche, Inc.

Until 1962, Hoffmann-La Roche, Inc., used a fragment code, on edge notch cards, to perform structure searches on our own compounds. This file assumed such proportions that it became too unwieldy so we investigated alternative procedures. We chose the Wiswesser notation because we found, among other reasons already outlined by Horner, that by judicious use of permuted notations, we could fulfill our substructure search requirements with minimum use of a computer. It has been our philosophy, for numerous reasons, to minimize computer time, and for this reason, you may find our system archaic; however, we have found it adequate for searching our file, which is now on the order of 70,000 notations.

In order to facilitate control, all notations for compounds synthesized by Hoffmann-La Roche either in the United States, Switzerland, or England are written in the United States.

The procedure for entering a new compound into the file is as follows:

1. The data sheet is checked to see that the name, the molecular formula, and the structural formula are correct and compatible.
2. The molecular formula is checked against the molecular-formula file to see if it is a new (our file) compound. In the future, uniqueness may be checked by notation. The molecular-formula file will be filed by rubric in notation sequence. We are investigating the use of the Dura Mach 10 for this input. By typing the data sheet on Dura, we can reproduce the data sheet, generate the molecular-formula card, and input notation all in one step, thus eliminating a key punch step.
3. Notation is written on a coding sheet, key punched and verified, and a list is printed out in numerical order. This is done on a weekly basis.
4. Notations are proofread (not by the person who wrote the original notation). At this point, we find an error rate of less than 3%.
5. Corrections are key punched and verified. Quarterly, a list of corrections is printed out.

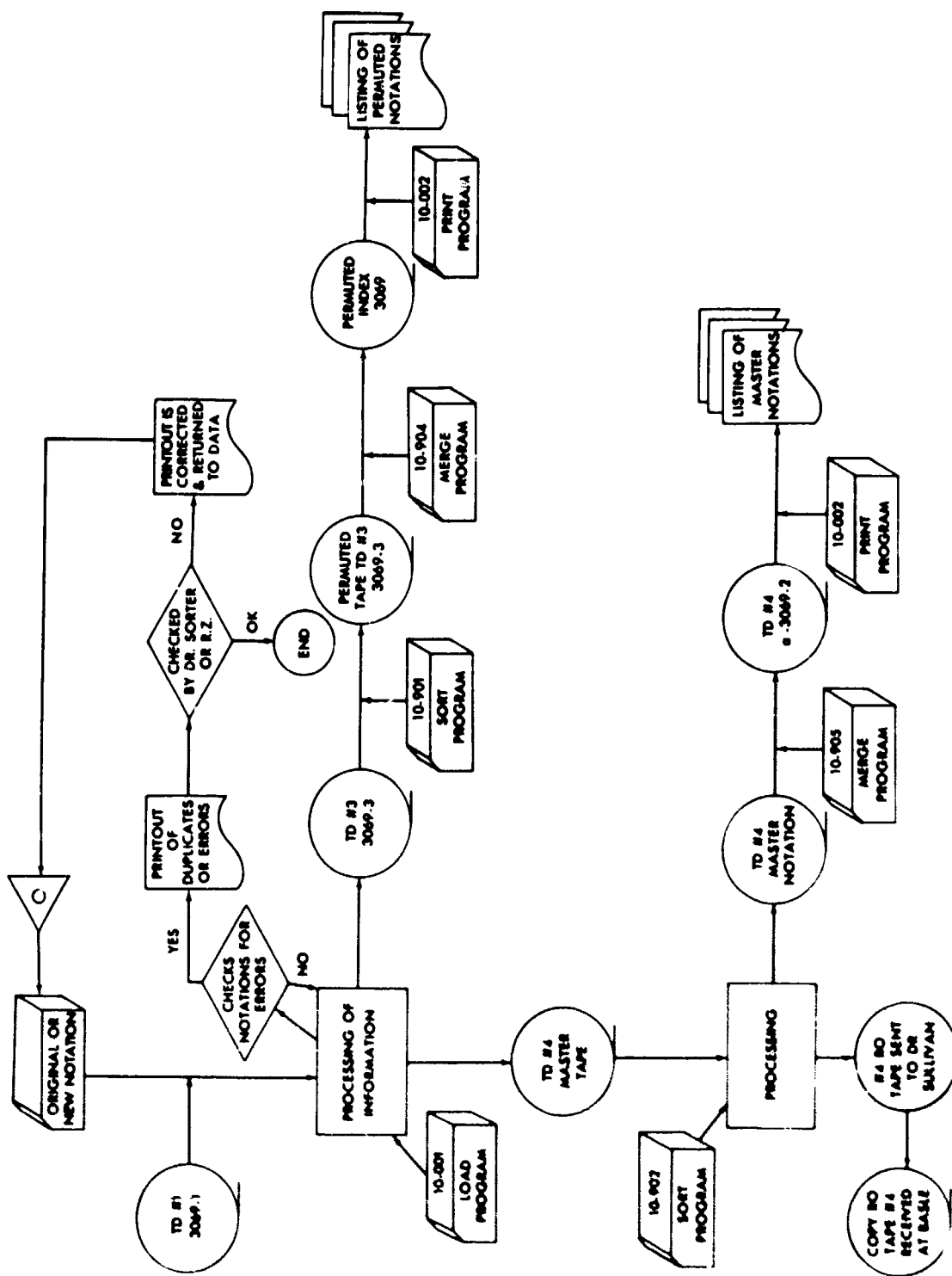
6. Updating on the 1401 computer is done as outlined on the flow chart.

- a. Cards are input.
- b. Error checks:
 - (1) Ro number already on file.
 - (2) Unacceptable chemist number.
 - (3) Cards out of sequence or card missing.
 - (4) Notation already on file.

In a study we did some time ago, we found that this checking procedure uncovered 1% of duplicate compounds in our files. This program also pulls together for us all stereisomeric compounds.

- c. Errors are printed out, checked and recycled.
- d. Permuted notations are prepared. Permuted symbols are extracted (R is included here though not permuted, to assist in searching), alphabetized, and listed to the right of the notation. The permuted notations are sorted alpha-numerically, merged with the old file, and a new permuted index is printed.
- e. Original notations are also sorted alpha-numerically, then merged with old file and printed.
- f. Numerical sort done on Ro number, merged with old file, and list printed in Ro number order.

7. Error corrections: card with Ro number and DELETE is fed in. This purges the notation with that Ro number on all tapes. The maximum number of deletions acceptable in one pass is 100. Then a card(s) is entered with the correct notation and fed in. This then goes through the entire cycle as if it were a new compound.



FLOW CHART

GENERAL INFORMATION:

42,000 notations when permuted = 4000 pages at 54 notations/page

1 notation = 5 entries

4600 notations	Load and permute	45 min
	Sort	2 hr 15 min
	List master notation	15 min
	List permuted notations	<u>1 hr 15 min</u>
	Time	4 hr 30 min

If done at outside service bureau:

Cost, machine time, \$433.00 or about 10¢ per compound.

We are studying methods for reducing this cost by going to variable instead of fixed record length. Furthermore, upon receipt of our new computer, a GE625, we hope to have instant access, on a time sharing program, to the updated file, as it is produced. This would allow us to do searching in the computer and have the pertinent notations instantaneously displayed on a TV screen, thus doing away with printed lists. In addition, we could then communicate with the computer, in a conversational mode, and ask a series of questions based on the answers obtained.

QUICK SCAN AND SYMBOLS

Alan Gelberg
Diamond Alkali Company

The Wiswesser line notation uses alpha-numeric symbols in addition to three characters (- / and &) commonly found on a typewriter keyboard. As seen in the appendix, the symbols A through Z all have specific meanings (and in some cases, more than one meaning) but in the context of the notation, the symbol meanings become readily apparent.

It is known that certain elements such as C, H, O, and N occur abundantly in organic compounds. W. J. Wiswesser assigned different symbols to account for the different manners in which these elements can be bonded to other atoms.

Such an example is carbon. Six carbon atoms (plus six hydrogen atoms) bonded in a resonating cyclic structure comprise benzene. Using approximate figures from the Chemical Biological Coordination Center (CBCC) file, more than 30% of the known organic compounds have a benzene ring as part of the molecule. This special case was given a symbol of its own, R.

Other ring systems, excluding benzene, make up another 30% or so of the known organic compounds. Those ring systems that are comprised of carbon atoms are offset with the introductory symbol L. If at least one atom in the topology of the ring system is noncarbon, then this heterocyclic is offset with the symbol T. This simple arrangement distinguishes major classes of compounds in a quick glance and permits rapid arrangement of structural types.

Alkyl chains or carbon atoms connected linearly are represented by a numeral that indicates the number of carbon atoms. When a branch

atom occurs such as in isobutane, $\text{H}_3\text{C}-\text{CH} \begin{array}{c} \text{CH}_3 \\ | \\ \text{CH}_3 \end{array}$, the graph can be 1Y, or,

linearly, 1Y1&1, or 1Y with contraction. Similarly, X represents a tetrasub-

stituted carbon atom as in neopentane, $\text{H}_3\text{C}-\text{C} \begin{array}{c} \text{CH}_3 \\ | \\ \text{CH}_3 \end{array} \text{CH}_3$, 1X1, 1X1&1&1, or 1X.

The next most frequently occurring atom is oxygen. This atom can occur as:

- a hydroxyl group (—OH) represented by Q,
- an ether (—O—) represented by O,
- a carbonyl (>C=O) represented by V, or
- a dioxo ($\text{O}=\text{C}=\text{O}$) represented by W.

The third most frequently occurring atom is nitrogen. This atom can occur as a:

- primary amine (—NH_2) represented by Z,
- secondary amine (—NH—) represented by M,
- tertiary amine (—NH—) represented by N, or a
- Katernary amine (—N—) represented by K.

Therefore, three of the four most frequently occurring atoms found organically in nature (excluding hydrogen) can be represented by at least four different symbols, plus combinations, to give a chemist a representation of the atomic bonding. Other symbols become fairly apparent, as seen from the list, with many symbols being the recognized chemical symbols from the Periodic Table, B, P, S, F, I, etc.

Molecular formulas are used in indexes to rapidly locate specific compounds, since a specific compound can have only one molecular formula; however, from the molecular formula, we do not know how the atoms are connected to each other, and more than one compound can be represented by the same molecular formula. With the Wiswesser line notation, each atom is defined so that molecular isomers are severely reduced, and specificity in a search is easily achieved.

From chemical nomenclature, the generation of the molecular formula has only partially been achieved as demonstrated by Gene Garfield's doctoral thesis at the University of Pennsylvania. Unfortunately, this study has not been completed. From the notation symbols, by assigning the appropriate multipliers to the atoms represented by symbols, molecular formulas are easily generated and serve as part of the checking procedure presented by Carl Bowman.

In 1962, when Charlie Granito and I were investigating the use of permuted notations, it became readily apparent that scanning lengthy listings of computer printouts caused eye fatigue. Therefore, the program was written to permit extraction of the symbols that we arbitrarily decided were significant. These were printed to the left of the page. We initially had the symbols presented in the order that they appeared in the notation. We felt that multisymbol combinations such as WN or NW, OVM, or MVO, etc., should appear together to be easily distinguished. Pete Sorter thought that it would be easier to "eyeball" the QUICK-SCAN area if the symbols were alphabetically ordered. After a year's use, indications were that alphabetizing the QUICK-SCAN area would be easier to work with, and I changed my program.

These symbols serve as a scanning device for viewing a listing of many entries in a specific section. In a matter of seconds, a page of 60 notation entries can be rapidly scanned.

It was interesting to learn at the Airlie House meeting in 1962, from Dyson and Lynch, that the Dyson cipher symbols were similarly being computer scanned as a screen prior to searching the matrix they had developed. It was also revealed by Landee and Bowman that the "Wis-symbols," as Franc called them, were being studied both as a screen and as a means of generating the molecular formula.

Therefore, it seems obvious that the symbols in the notation can serve as the basis for: (1) generating listings that can be visually scanned; (2) a computer scanning method; (3) a computer method for rapidly calculating molecular formulas and molecular weights, and, as will be presented later; (4) a computer method for generating tables of symbols in an inverted order to be used as fragment tables; and (5) a computer method for generating a connectivity table, from which a packed storage can be obtained.

APPENDIX

REFERENCE LIST OF THE LINE-FORMULA NOTATION SYMBOLS

All the international atomic symbols are used except K, U, V, W, Y, Cl, and Br. Two-letter atomic symbols in organic notations are enclosed between hyphens. Single letters preceded by a blank space indicate ring positions. Single letters not preceded by a blank space have the following meanings:

- A generic alkyl
- B boron atom
- C unbranched carbon atom multiply bonded to an atom other than carbon
- D proposed symbol for a chelate bond and initial symbol of a chelate notation
- E bromine atom
- F fluorine atom
- G chlorine atom
- H when preceded by a locant within ring signs, shows the position of a carbon atom bonded to four other atoms; elsewhere H means hydrogen atom.
- I iodine atom
- J generic halogen; also sign for the end of a ring description
- K nitrogen atom bonded to more than three other atoms
- L first symbol of a carbocyclic ring notation
- M imino or imido —NH— group

N	nitrogen atom, hydrogen free, attached to no more than three other atoms
O	oxygen atom, hydrogen free
P	phosphorus atom
Q	hydroxyl group, —OH
R	benzene ring
S	sulfur atom
T	first symbol of a heterocyclic ring notation; or within ring signs indicates a ring containing two or more carbon atoms each bonded to four other atoms
U	double bond
V	carbonyl connective, >C=O (carbon attached to three other atoms)
W	nonlinear (branching) dioxo group (as in —NO ₂ or —SO ₂ —)
X	carbon atom attached to four atoms other than hydrogen
Y	carbon atom attached to three atoms other than hydrogen or double bonded oxygen
Z	amino or amido —NH ₂ group
&	punctuation mark showing the end of a side chain; or preceded by a space, sign of ionic salt, addition compound or suffixed information; or within ring signs indicates a ring NOT containing two or more carbon atoms that are bonded to four other atoms; or following a hyphen, shows certain spiro ring connections
-	separator or connective or other special uses
/	stops action of a multiplier or encloses multiplied symbol groups

Numerals preceded by a space are multipliers of preceding notation symbols; or within ring signs L...J or T...J show the number of multicyclic points in the ring structure.

Numerals not preceded by a space show ring sizes if within the ring signs; elsewhere numerals show the length of internally saturated, unbranched alkyl chains and segments.

Letters following a space and hyphen are proposed as symbols with special meanings to denote stereoisomerism.

AUTOMATIC GENERATION OF STRUCTURAL FRAGMENT
CODES FROM THE WISWESSER LINE NOTATION
FOR RAPID STRUCTURE SEARCHES

Carlos M. Bowman
Franc A. Landee
Mary H. Reslock
Betsy P. Smith
The Dow Chemical Company

INTRODUCTION.

The problem of establishing chemically oriented files consists of arranging many different types of information, including structural information, in such a manner that one general type of search procedure can search all manners of questions with reference to that diverse information.

A solution is to numerically code all types of information into the same inverted file. This coding can be done by computer programs, which allows some very complex coding procedures to be used.

When this coding is done to Wiswesser representation of structure, it is called "fragmenting." This short note will describe something about the process, something about the code definitions, and something about how inverted files formed from such fragments (codes) can be searched.

FRAGMENT CODES.

The structural fragment codes consist of five digit numbers, the first of which is decimal 0, 1, 2, --- 7, 8, 9 and the next four of which are octal in nature 0, 1, 2, --- 6, 7. (No eights or nines.) Altogether from 00000, the lowest, to 97777 the highest, there are 40,960 numbers almost all of which have assigned meanings.

These assignments have been done in a regular orderly pattern, which makes them easy to remember.

MOLECULAR FORMULA CODES.

All codes starting with zero refer to elemental and atomic consideration of the molecule. As such, they are found by a computer

program analysis of the molecular formula, which may or may not have been derived from the Wiswesser notation by some other program.

The codes that start with 04iii refer to the N, O, P, S content of the molecule and these will be discussed in more detail.

If the code has a zero in the fourth position (from the left), i.e., of this form:

04i0j

then the molecule contains only one type of the four atom types listed above. The "i" indicates the type, the "j" indicates how many. For example:

04103	three nitrogens
04407	seven (or more) sulfurs
04301	one phosphorous

If the code does not contain a zero in the fourth position and if the third position is not a six or a seven, hence of this type:

04ijk (j≠0 i≠6 or 7)

then pairs of the types of atoms listed exist in the molecules. The "j" indicates which type pair, i.e.,

0	N+O	3	O+P
1	N+P	4	O+S
2	N+S	5	P+S

The "j" indicates how many of the first type, the "k" indicates how many of the second type, thus:

04123	two nitrogens, three oxygens
04466	six oxygens, six sulfurs

If three or four types appear in the molecule then two codes are generated and of the type:

046ij i is number of nitrogens; j is number of oxygens

047kl k is number of phosphorus; l is number of sulfurs

Thus, all combinations (up to 7 of each) can be uniquely defined by these 512 codes so described.

"DO NOT CARE" OPTION.

These same codes are used to describe the searching questions with some refinements which allow broader search questions to be phrased more simply than by using the continued "AND" or "OR" construction.

The codes as described up to now are very specific in nature. For example, a request for:

04101

would give all compounds that contain only one nitrogen and no oxygen, phosphorus, or sulfur. To get all compounds containing only nitrogen one would have to ask for:

04101 or 04102 or 04103 or --- -4107

This has been simplified by asking only

0410*

since the asterisk indicates that one does not care what is in that last position.

The numbers eight or nine can also be used in these searching words with the meanings up to and including three and four to seven, respectively. Thus:

04*08 will give all compounds that have only one of the elements N, O, P, or S present but one, two, or three of that element.

04609 the molecules selected will all contain
oxygen, phosphorus, and sulfur and at least
four oxygen atoms.

The power and the range of selectivity that have been built into
the coding and searching combination can now be seen more readily.

The codes that start with 05iii are very similar (exactly
parallel) to the 04iii codes except that they refer to the fluorine, chlorine,
bromine, iodine combinations, and possibilities. Thus, 1,000 or one-
fourth of the elemental codes have been described.

WISWESSER FRAGMENTS.

The codes that start with a 1 and hence have the form:

liiii (i's are octal)

are derived from the suffix's of the Wiswesser notation and, hence,
refer to matters such as mixtures, tautomers, charges, isotopes, poly-
mers, stereoisomers, and other considerations. Time will not allow a
discussion of these codes.

In all cases a great deal of redundancy and overlapping has
been built into the system. This will become more evident as we proceed.

The codes that start with a 2 are derived from the Wiswesser
notation itself, and refer to side chains, by themselves, on rings, between
rings, etc. As such, they are of the form:

2ikjj

The "jj" pair has 77 octal or 64 decimal possibilities and is
used to code groups and fragments for which the single letter Wiswesser
symbols are used. Actually, one place is taken from the k value so that
177 octal or 128 decimal possibilities are available here. For example:

-072 is for a hydroxyl or Q in Wiswesser

-160 is for a benzene ring or R in Wiswesser

The 128 numbers are assigned in patterns so that the 8, 9, or *
questions can be effective. Thus, any code that is in the lower quadrant is

that of a terminal group or terminal combination, since common combinations are also assigned codes. Thus, codes 000, 001, 002, 003, 010, 011, 012, and 013 refer to terminal methyl, ethyl, n-propyl, n-butyl---n-octyl groups, while codes 004, 005, 006, 007, 014, 015, 016, and 017, which are in the upper quadrant, refer to connective methylene, ethylene, n-propylene, ---n-octylene groups. Continuing, code -150 is for $=CH_2$, while code -154 is for $=CH-$, code -151 is for $\equiv CH$, while code -155 is for $\equiv C-$, etc.

Asking for -048 will give all compounds containing single halogens per group, asking for -058 will get all compounds containing doubled halogens on the chain, while asking for -098 will give all side chain halogen, except iodonium and similar connective halogens.

The "i" values of the 2 codes have the following meanings:

20k11	- no rings in compound.
21k11	} - the side chain in question starts from a benzene ring--but the "11" group is not attached directly to that ring.
22k11	
23k11	} - starts from an L---J ring not directly attached.
24k11	
25k11	} - start from a T---J ring not directly attached.
26k11	
27k11	} - same but directly attached to ring carbon.
	- same but directly attached to noncarbon ring element.

There remains only the "k" values to be explained. Since one bit was taken from k for the use of "11," "k" can have only the values 0, 2, 4, and 6. These have the meanings:

k = 0	- the side chain is simple--does not connect ring to ring.
k = 2	- the side chain connects similar rings; i.e., R to R, L---J to L---J, etc.

- k = 4 - the side chain connects between rings of one degree difference in complexity; i.e., R to L---J, L---J to T---J.
- k = 6 - the side chain connects between rings of two degrees difference in complexity; i.e., R to T---J.

This concludes the description of the 4,096 2 codes. As may be seen, a great deal of structure is built into these basic fragment codes so that searching questions can be very narrow and specific, or very broad and generic, in both a group and in a structure sense.

The 3, 4, 5, 6, 7, 8, and 9 codes deal progressively and in similar manners with the problems of groups within rings, rings themselves, either single rings or fused rings, up through the problems of bridges and overall ring of rings considerations.

SEARCHING.

As stated earlier, a computer program has been written which generates the fragment codes described. The program is not complete in that all of the fragments are not generated as yet; however, sufficient codes are selected so that meaningful search questions can be asked. The program examines each notation and its molecular formula, which have been stored on magnetic tape, and creates first the molecular formula fragments and then the structure fragments, which are then also stored on magnetic tape.

Each fragment is coupled to the compound number in one computer word so that a simple numeric sort will bring together all entries for a given fragment. From this sorted tape, an updating program creates an inverted index tape, which can then be searched using the standard Dow searching program.¹ This is a very efficient program which permits a search to be carried out in a matter of minutes at a very low cost.

Any combination of fragments with the connectives AND, OR, NOT may be used. The "do not care" option was used to allow the questions to specify easily a large number of terms connected by the OR operator. At run time, the "do not care" portions of the fragment are expanded into the appropriate list.

The present output of the search program is simple compound numbers. Work is underway to add the notation, formula, and other pertinent data about each compound to the output. Earlier work^{2, 3} has shown

the feasibility of creating structural formula diagrams from the notation. It is anticipated, especially with the advent of a much improved graphic technology or computers, to add this very desirable feature to the search. Thus, the inquirer will be able to obtain not only the compound number that he might use to reference other data, but he will be able to see the picture of the compound.

The structure handling system, which has been referred to in this paper, is only a part of a comprehensive computer-based information-handling system designed at the Dow Computation Laboratory. Other features of this system have been described⁴ and consist of current awareness techniques;⁵ retrospective searching; printed indexes; etc.

SUMMARY.

A computer program has been described, which can create fragments from the Wiswesser notations, suffixes, and molecular formulas to give a comprehensive numerical fragment code. Some detail of the structure of these codes has been given.

The fragments have been organized into an inverted file index, thus allowing the application of concept coordination techniques at search time. This results in fast and economic handling of large volumes of information.

It has also been pointed out that this system is a part of an integrated information system, which handles many other types of information.

LITERATURE CITED.

1. Farris, R. N. Computers Cut the Cost of Literature Searches. Chem. Eng. Progr. 62, No. 5, 89-91 (1963).
2. Landee, F. A. Computer Programs for Handling Chemical Structures Expressed in the Wiswesser Notation. Presented Before the Division of Chemical Literature, 147th National Meeting of the American Chemical Society. Philadelphia, Pennsylvania. 8 April 1964.
3. Landee, F. A. Computer Methods of Handling Files of Chemically Oriented Information. Presented in Moscow, U.S.S.R. October 1965.

4. Bowman, C. M. A Corporate Attack on Personal Files. Chem. Eng. Progr. 62, No. 5, 85-88 (1962).

5. Bond, L., Bowman, C. M., and Hartman, D. User Reaction to Three New Services Offered by the Chemical Abstracts Service. Presented Before the Division of Chemical Literature, 152nd National Meeting of the American Chemical Society. New York, New York. 4 September 1966.

COMPUTER GENERATED OPEN ENDED FRAGMENT CODE *

Ernest Hyde
Canadian Industries Ltd.

In general, fragment codes break a molecule into recognizable part structures, and the fragments chosen depend very much on the nature of the file, and the end to which it is employed. The following method takes a slightly different approach.

The object has been to allow the computer to generate fragments having been programmed along certain guide lines. Therefore, at the commencement of the operation, the fragments that will be generated are largely unknown. As novel compounds are added to the file, the computer will generate new fragments every time it meets a new condition, and hence, the fragment code is open ended.

The program operates from either the notation using portions of the matrix programs or directly from the compacted matrix. Each fragment generated is composed of a string of Wiswesser symbols in canonical order and varies from 2 to 10 symbols in length, the majority being four symbols long. Every fragment is assigned a number, and a compound is registered by entering its serial number under each fragment contained in the molecule in an inverted magnetic tape file.

The fragments thus obtained are listed using a KWIC program. This brings together all fragments containing the same Wiswesser symbols. An enquiry made to the file is examined against pertinent sections of the KWIC to establish under which numbers the search should be performed.

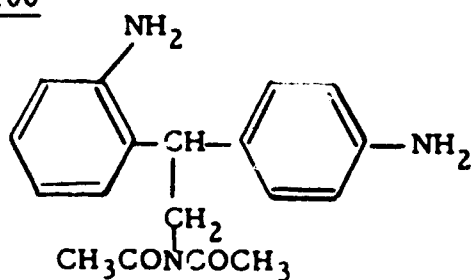
One advantage of the KWIC list of fragments is that it will quickly lead the searcher to subfragments contained in larger fragments. Hence, you do not face a situation where you cannot decide whether to say diguanide is a guanide plus, or a guanide is a diguanide minus. This method will always choose diguanide, and by consulting the list of fragments you will extract all fragments listed that contain at least all of the desired symbols in the correct Wiswesser order. An advantage in preserving the larger fragment is that both the larger and the smaller fragment can be correctly identified. If a fragmentation code chooses the smaller fragments and expects the questioner to reconstruct the larger, there is always the danger, present in all fragmentation systems, of false coordination.

* Additional information on this subject has been published in the Journal of Chemical Documentation, November 1967.

Compounds

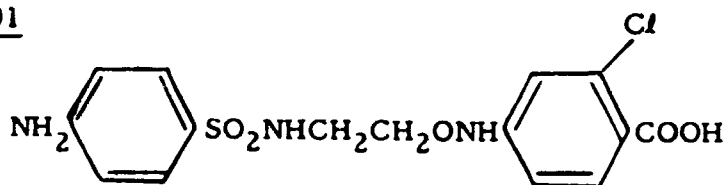
Fragments in
Wiswesser Symbols

000100



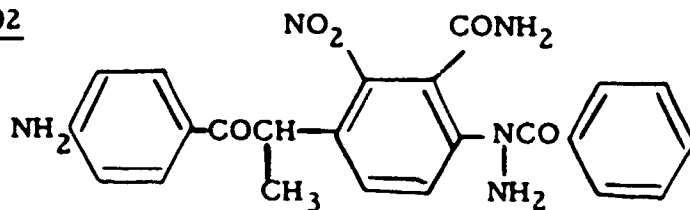
RZ
RAR
ANVAVA

000101



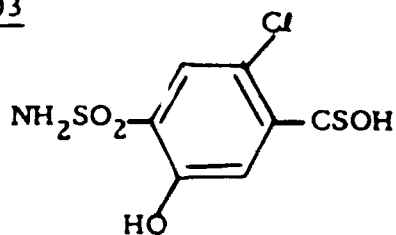
RZ
RG
RVQ AOMR
RSWMA

000102



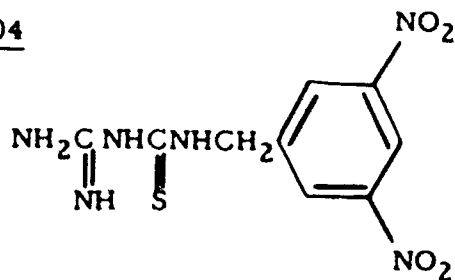
RZ RVA
RNW AR
RVM RNZVR

000103



RG
RQ
RSWZ
RYQS

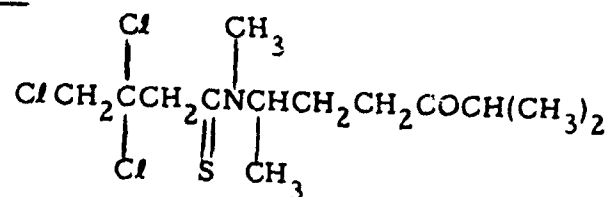
000104



RNW
ZYMMYMA

Fragments in
Wiswesser Symbols

000105



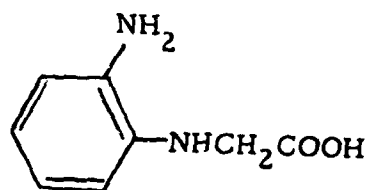
AG

GAG

AYSNAA

AVA

000106

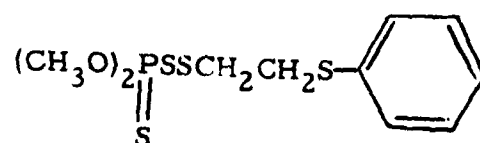


RZ

RMA

AVQ

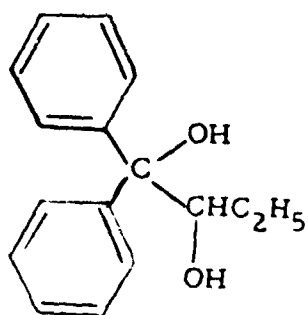
000107



AOPSOASSA

ASR

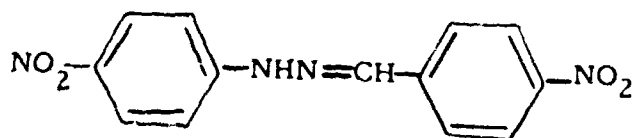
000108



RAR

AQ

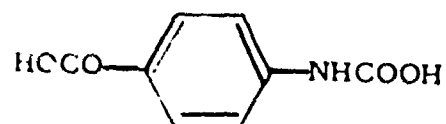
000109



RNW

RMNA

000110



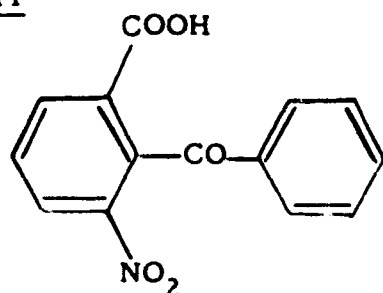
RA

RVQ

RMVQ

Fragments in
Wiswesser Symbols

000111

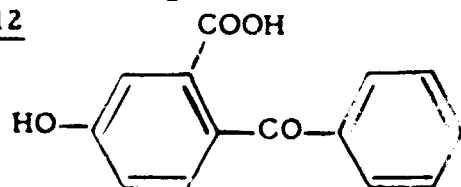


RVQ

RVR

RNW

000112

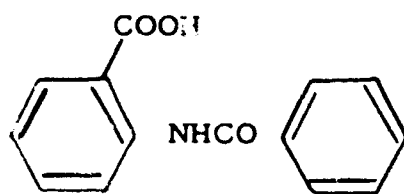


RVQ

RVR

RQ

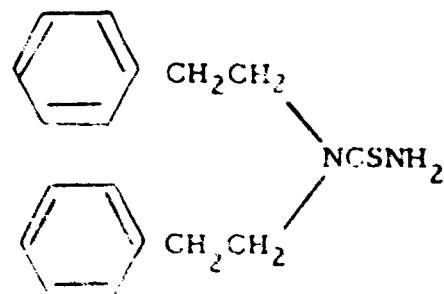
000113



RVQ

RMVR

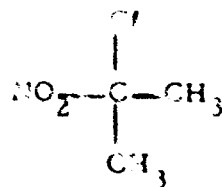
000114



ZYSNAA

RA

000115

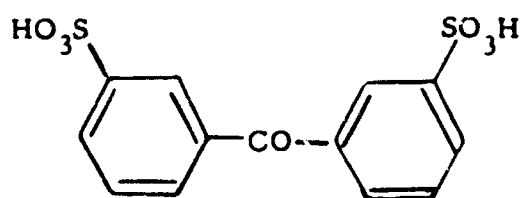


ANW

AG

Fragments in
Wassesser Symbols

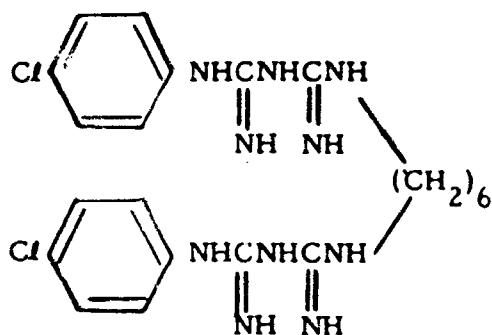
000116



RSWQ

RVR

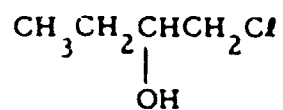
000117



RG

RMYYMYMA

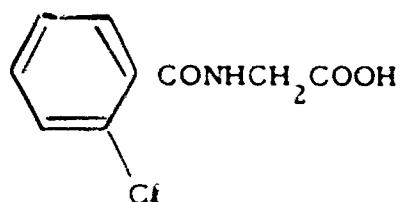
000118



AG

AO

000119



RG

RVMA

AVQ

OPEN ENDED FRAGMENT CODE DERIVED FROM
WISWESSER CONNECTIVITY MATRIX

<u>Fragment</u>	<u>Fragment Number</u>
*A C N	061
*R C N	066
*R E	003
*A E	033
*E A E	039
*E R E	339
*E A E	039
*E R E	339
*R F	004
*A F	034
*F R F	010
*F A F	040
*F A F	040
*F R F	010
*R G	001
*G A G	037
*A G	030
*G R G	007
*A V G	062
*R V G	067
*G A G	037
*G R G	007
*R S H	065
*A S H	060
*R I	002
*A I	031
*I A I	038
*I R I	008
*I A I	038
*I R I	008
*R Y Z M	028
*A M Y Z M	59
*A M Y M M Y Z M	032
*A Y Z M	057
*R Y S M A	457
*R S W M A	355
*A M V M M V M A	615
*A M V M V M A	271
*A M V M A	159
*A M A	101
*R M A	235
*A Y S M A	255
*A M Y M M Y M A	476

<u>Fragment</u>	<u>Fragment Number</u>
*R M M R	789
*A M Y M M R	693
*A M V M M V M A	615
*A M Y M M Y M A	476
*A M Y M M Y Z M	032
*A M N A R	494
*R M N R R	024
*A M O A	838
*A M Q	272
*R M Q	579
*R M M R	789
*R V M R	135
*R O M R	538
*R S W M R	518
*A M Y M M R	693
*A M R	235
*A S W M R	756
*R M R	802
*A O M R	278
*A M V A	815
*A M V M M V M A	615
*A M V M V M A	271
*A M V M A	159
*A M V M M V M A	615
*A M V M V M A	271
*A M V N A R	493
*A M V Q	837
*R M V Q	791
*A M V R	570
*R M V R	135
*R M V Z	456
*A M V Z	059
*A M Y M M Y M A	476
*A M Y M M R	693
*A M Y M M Y M A	476
*A M Y M M Y Z M	032
*A M Y S N R R	312
*R M Y S S Y S N A A	680
*A M Y M M Y Z M	032
*A M Y Z M	259
*R S W M Z	123
*R V M Z	234
*R M Z	468
*A C N	061
*R C N	066
*R N A	246
*R Y S N A A	891
*R M Y S S Y S N A A	680
*A N A A	499

<u>Fragment</u>	<u>Fragment Number</u>
*R O N A R	812
*A M N A R	494
*A M V N A R	493
*R N N R	578
*R N Q A	801
*A N Q A	334
*R N N R	578
*R S W N R R	911
*R N R R	136
*R M N R R	024
*A M Y S N R R	312
*R N W	913
*A N W	810
*R V N Z A	567
*A O A	045
*R O A	046
*R V O A	026
*A M O A	838
*A V O A	055
*R O M R	588
*A O M R	278
*R O N A R	812
*R O R	016
*R O Z	023
*A O Z	053
*R Q	006
*A Q	036
*R V Q	025
*Q R Q	012
*A M Q	272
*A M V Q	837
*R M V Q	791
*R M Q	579
*R S W Q	069
*A V Q	054
*A S W Q	064
*Q A Q	042
*R N Q A	801
*A N Q A	834
*Q A Q	042
*Q R Q	012
*R Y Z S	017
*A Y Z S	047
*R Y S A	027
*R S S A	020
*A Y S A	056
*A S A	048
*R S A	049
*A S S A	050

<u>Fragment</u>	<u>Fragment Number</u>
*A Y S A	058
*R S H	065
*A S H	060
*R Y S M A	457
*A Y S M A	255
*R Y S N A A	891
*R M Y S S Y S N A A	680
*A M Y S N R R	312
*R Y S R	029
*R S R	018
*R S S R	019
*R S S A	020
*A S S A	050
*R S S R	019
*R M Y S S Y S N A A	680
*R S W A	021
*A S W A	051
*R S W M A	355
*R S W M R	518
*A S W M R	756
*R S W M Z	123
*R S W N R R	911
*R S W Q	069
*A S W Q	064
*A S W Z	063
*R S W Z	068
*R M Y S S Y S N A A	680
*R S Z	022
*A S Z	052
*A M V A	815
*A V A	043
*A V G	062
*R V G	067
*A M V M M V M A	615
*A M V M V M A	271
*A M V M A	159
*A M V M M V M A	615
*R V M R	135
*A M V M V M A	271
*R V M Z	234
*A M V N A R	493
*R V N Z A	567
*R V O A	026
*A V O A	055
*R V Q	025
*A M V Q	827
*R M V Q	791
*A V Q	054
*A V R	013

<u>Fragment</u>	<u>Fragment Number</u>
*R V R	014
*A M V R	570
*R M V R	135
*R M V Z	456
*A V Z	044
*R V Z	015
*A M V Z	059
*R N W	913
*A N W	810
*R S W A	021
*A S W A	051
*R S W M A	355
*R S W M R	518
*A S W M R	756
*R S W M Z	123
*R S W N R K	911
*R S W Q	069
*A S W Q	064
*A S W Z	063
*R S W Z	068
*A M Y M M Y M A	476
*A M Y M M R	693
*A M Y M M Y M A	476
*A M Y M M Y Z M	032
*R Y S A	027
*A Y S A	056
*A Y S A	058
*R Y S M A	457
*A Y S M A	255
*R Y S N A A	891
*R M Y S S Y S N A A	680
*A M Y S N R R	312
*R Y S R	029
*R M Y S S Y S N A A	680
*R Y Z M	028
*A M Y M M Y Z M	032
*A M Y Z M	259
*A Y Z M	057
*A Y Z S	047
*R Y Z S	017
*R Z	005
*R M V Z	456
*R S W M Z	123
*R V M Z	234
*A Z	035
*A V Z	044
*R S Z	022

<u>Fragment</u>	<u>Fragment Number</u>
*R O Z	023
*Z R Z	011
*R V Z	015
*A M V Z	059
*R M Z	468
*R S W Z	068
*A S W Z	063
*Z A Z	041
*A S Z	052
*A O Z	053
*R V N Z A	567
*Z A Z	041
*R Y Z M	028
*A M Y Z M	259
*A M Y M M Y Z M	032
*A Y Z M	057
*Z R Z	011
*A Y Z S	047
*R Y Z S	017

SUBSTRUCTURE SEARCHING ON NOTATIONS

George F. Fraction
Eli Lilly and Company

The ability to provide the research scientist with chemical information from internal files is dependent on: (1) the organization of the information, (2) the nature of the query, and (3) the strategy of the search system. This discussion is mainly concerned with providing chemical structure information.

Because no comprehensive search procedure, which will anticipate all types of queries, has yet been formulated, nor will it be in the near future, we must use search procedures that can provide answers for specific types of questions. In a man-machine system, search procedures by the machine can be made more comprehensive through the ability and understanding of the man function. The machine has a finite capacity (although the capacity may be large) and it can operate on only a finite number of question types. The function of the man, then, in this man-machine system is to reformulate the question types to new questions, or sets of questions, which can be understood and operated on by the machine. The magnitude of the man functions may be reduced by a factor of to what degree machines, or, perhaps, more accurately stated, machine programs, are made intelligent enough to assume more of these man functions. Figure 1 is a diagram of how information flows throughour man-machine system.

Since our search procedure is a two-component system of man and machine, each component must be assigned to those tasks it can do most effectively. One task the man can, at present, do better than the machine is to select the search mode which can most easily satisfy a given query. To this end, we at Lilly are considering the following search modes:

1. Manual searches based on machine-prepared media. These are directories prepared by the KSOC¹ (Key Symbol Out of Context) and the permuted chemical line notation² computer programs.

2. Machine searches of manually prepared questions on machine-stored media, QUEEK and SUBMAP computer programs.

SUBMAP is an atom-by-atom, bond-by-bond substructure mapping program where both the reference structure and question structure are in the form of connection tables.

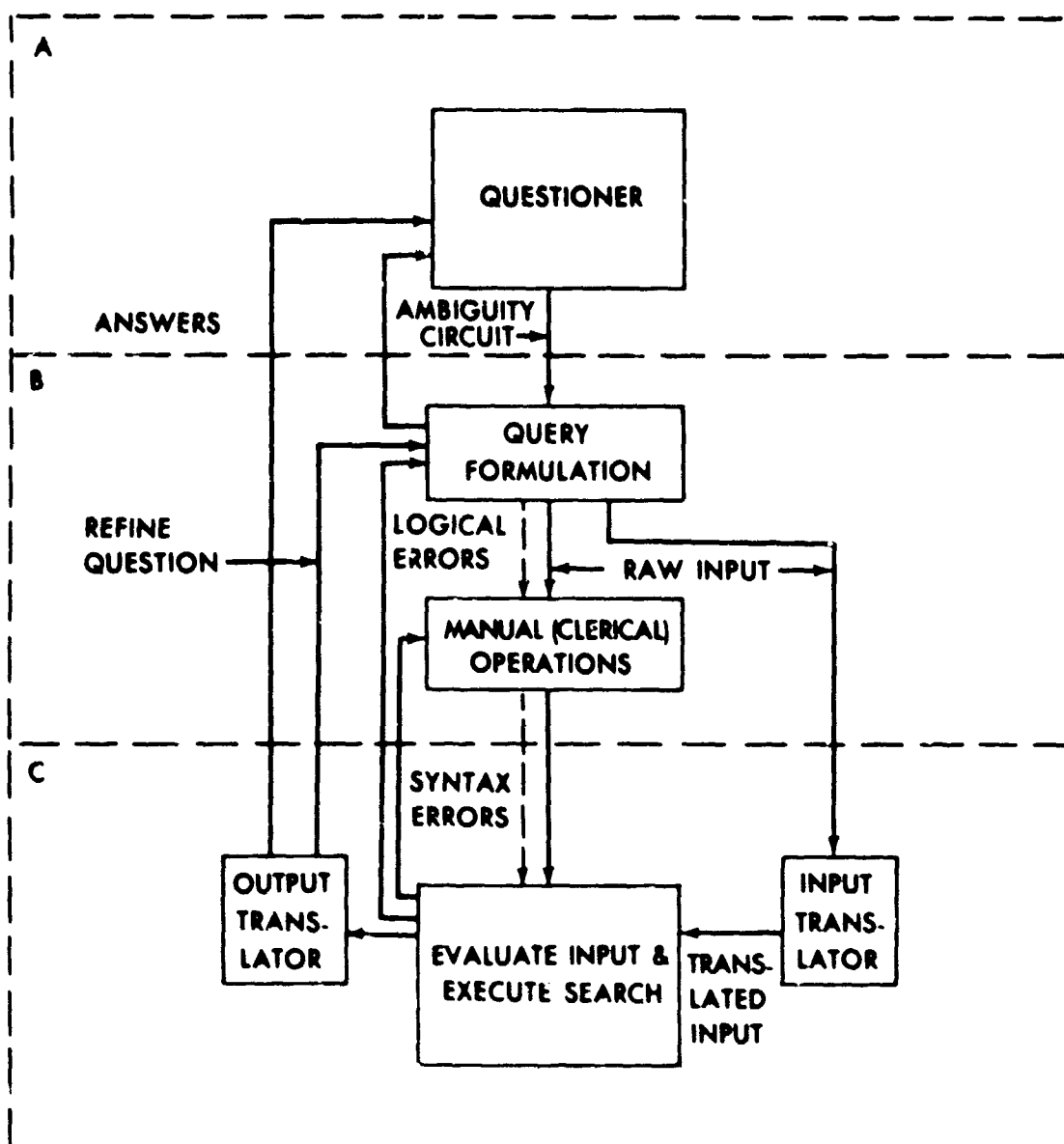


Figure 1. Information Flow Through a Man-Machine System

Since there have been various techniques developed for searching connection tables, let us turn our attention to the substructure search system based on notation symbols. This system is QUEEK, or Quick-Seek Search.

QUEEK is a computer program that will execute searches for questions within a given system of symbols; for instance, a system of symbols for describing uniquely and unambiguously some information, say chemical structure, wherein the questions can be completely specified by sets of symbol strings.

Smith³ has described a search system based on Wiswesser symbols to be used with punched cards. To overcome the problem of unwanted cards or "false drops," context information was provided by punching a specific position in the question card. This system, as well as the KSOC system, was inflexible in the respect that the permissible context definitions were provided by the system rather than the question.

The QUEEK program overcomes this difficulty. The context information is provided at search time by symbol strings containing fixed, variable, and logic symbols.

The QUEEK program uses the following logic:

1. AND
2. OR
3. NOT
4. PRECEDED BY
5. FOLLOWED BY

PRECEDED BY and FOLLOWED BY logic is currently restricted to single symbols or lists of single symbols within symbol strings (SYMSTR).

QUEEK recognizes the following logic and variable symbols within SYMSTR for searching Wiswesser line notations:⁴

- = any symbol
- # not
- (,) ORLIST Delimiters

- % any alphabetic character
- \$ any numeric value (0 to 99)
- @ any halogen

Input to the search program is a question list. The syntax for such a question list is shown in the appendix in Backus Normal Form notation.⁵

As shown in figure 2, QUEEK is written in four main modules: BUILD, SEARCH, SUMATE, and POST.

BUILD receives as input a deck of punched cards containing question and/or user file updating information. BUILD then constructs three files, QFILE, SFILE, and the UPDATE file. QFILE is the list of symbol strings for which the search is made. SFILE contains the identification of the question symbol strings to be given the AND operation. UPDATE contains the user file updating information. New users can be logged into the system and present a query on the same computer run. Since the system is designed to accommodate any number of users in any computer run, the questions of each individual user must be uniquely identified. This is provided by the USERID which is a concatenation of the UID, generated by the POST module, and the QID, provided by the user. When a new user makes a query, his question block is preceded by a header card identifying him by name and department number. He is assigned a temporary identification number, which is used for this run only. The POST module will assign the permanent identification that should be used on all subsequent runs.

SEARCH performs the search of the CFILE (cipher file) for the notation symbol strings (figure 3).

The SUMATE module performs the AND function for the questions specified on the ANDSPECS cards. This module allows a maximum of 36 QCARDS to be ANDed for each major question and allows many major questions submitted by one or more inquirers to be run simultaneously. The SFILE, which was constructed from the input deck by the BUILD module, contains all the AND information and a description of the form in which the requester wants his results.

SUMATE builds two tables, USERNDX and USANSBLK, which together meet the requirements of an inverted file of serial numbers of notations for which the AND function, as specified by an ANDSPECS card, is

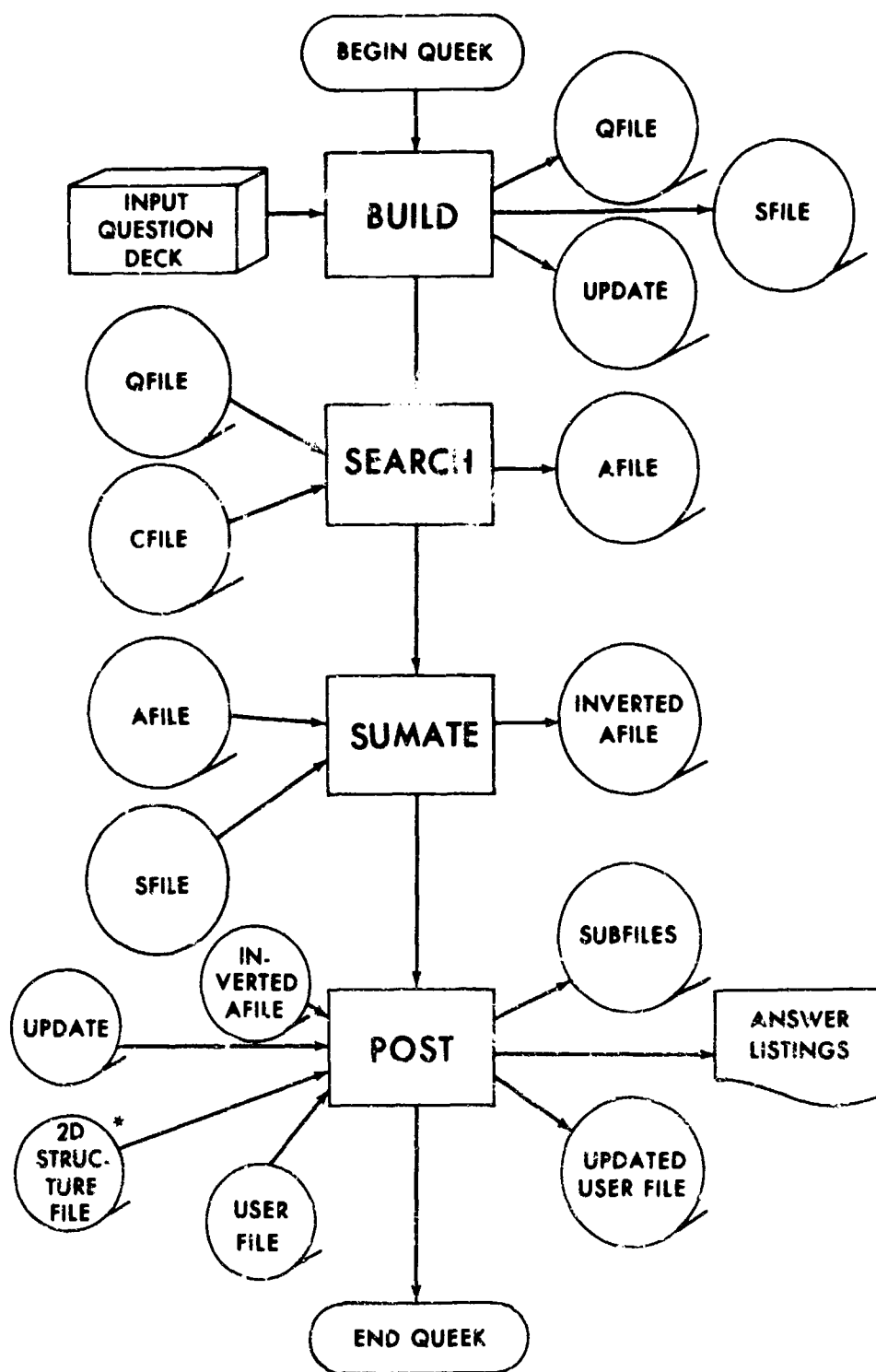


Figure 2. I/O Chart and Module Hierarchy

* Magnetic tape image of the Dura Mach 10 chemical typewriter output.

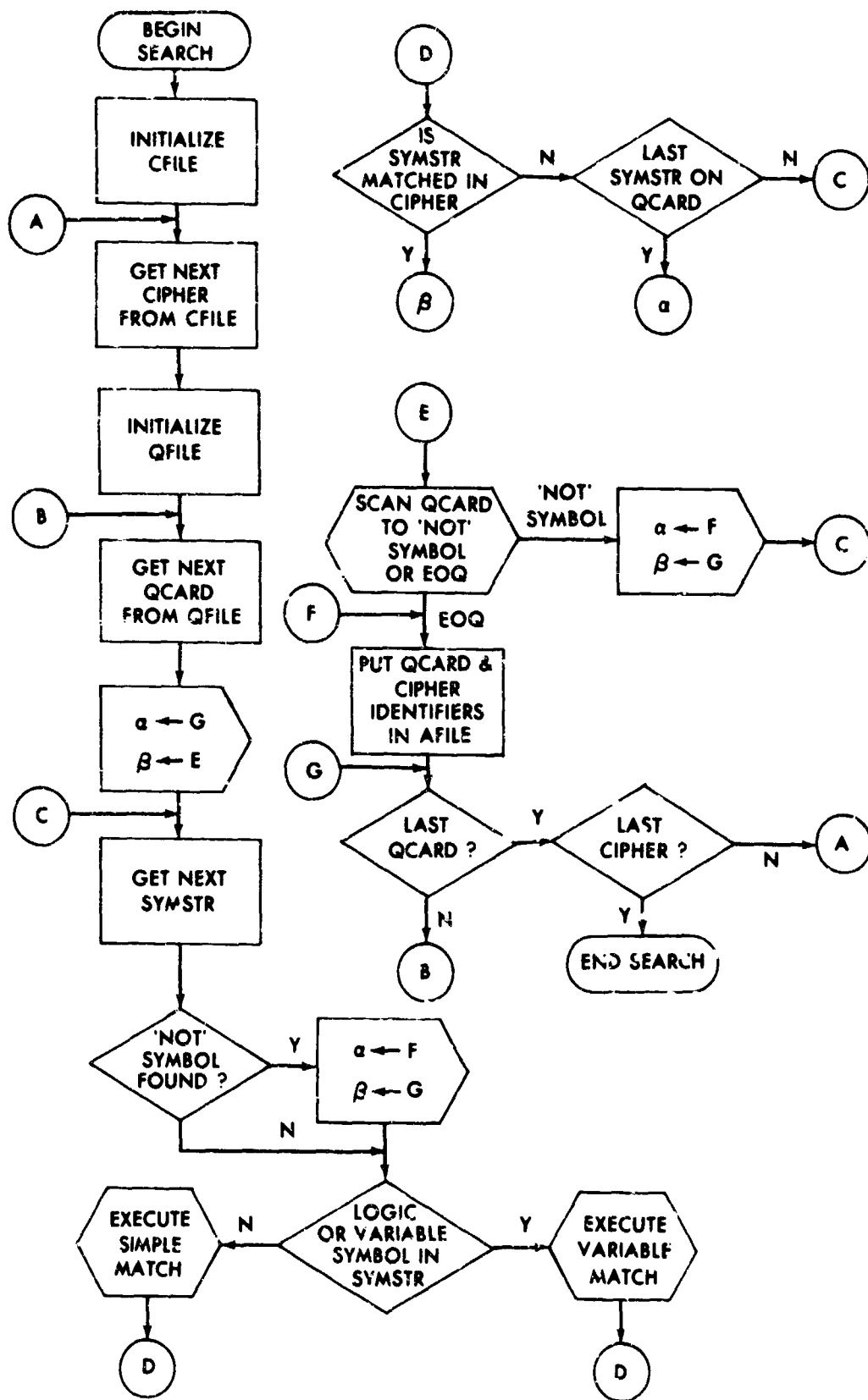


Figure 3. Logic Diagram for the Search Module of QUEEK

true. Pointers in the USERNDX table point to contiguous areas in USANSBLK where the answers to any specific major question are located.

The POST module has a number of clerical jobs.

1. Update the USER file.
 - a. Add new users and generate unique I.D.
 - b. Delete old users.
2. Maintain 'hit' statistics for evaluation of the effectiveness of the system.
3. Provide output as specified by ANDSPECS cards.
 - a. Print the number of answers found.
 - b. List the serial numbers.
 - c. Print the Wiswesser line notations.
 - d. Print the two-dimensional structure diagrams.
 - e. Create subfile tapes.
4. Provide routing information to the computer operators (i.e., who gets what output).

Before committing time and effort to QUEEK, a study was made of the basic system and a program of the simplified forms of SUMATE and SEARCH was written and run. The results of the study were encouraging. Total computing time was dependent only on the speed of the I/O gear. For this pilot operation, both the CFILE and QFILE were on punched cards.

We call your attention to the fact that the program we have just outlined is a nonintellectual search procedure. It performs only matching and adjusting functions on symbol strings provided by the man component of the system. The program's ability to find answers is dependent on the man component's knowledge of the notation system. This facility for formulating questions such that they are uniquely specified resides somewhere in the man-machine overlap area shown in figure 1.

After our description of QUEEK, we went on to discuss the application of this program to searches on one such system of symbols, the Wiswesser line notation. Clearly, there are other chemical structure notation systems based on symbols; likewise, data systems are based on symbols. This search procedure can be easily adapted to any such system.

We have, at Eli Lilly and Company, several files based on systems of coding in symbols. For example:

1. Fragmentation code file.
2. Physical data file.
3. Varian coding for NMR data.

Each entry in any particular file is identified by a Lilly compound number. Therefore, a search system which provides this unique number can be used to integrate these files in a logical, if not physical, manner.

Planned extensions to this program are in two, seemingly opposite, directions.

1. Increased specificity: teaching the machine more of the logic of the WLN such that it can assume more responsibility in the man-machine overlap area.
2. Increased generality: involving the writing of a preprocessor that will take, as input, definitions of the users' files to be searched, question formats, and search logic symbols.

Since this program has modular structure, both of these objectives can be achieved by inserting or deleting the desired module.

We have shown how the QUEEK program can be used to:

1. Execute real time searches.
2. Prepare listings or directories.
3. Create subfiles.
4. Integrate existing files.

It has been pointed out by Marron⁶ et al that a notation (symbol sequence) for a given substructure depends on what that substructure is attached to. But there are still only a specific number of ways any substructure can be cited. More cleverness in question formulation may be required to effect some searches than others. The thoroughness of the search is directly proportional to the thoroughness of the question body preparation. In some cases, the number of symbol variations and combinations to gather all the answers and only the unique answers - i. e., no "false drops" - may be so large that it would be undesirable to use this method. For these cases, QUEEK can be invoked as a screening device to provide candidates for a more exhaustive, and perhaps time-consuming, procedure.

QUEEK does not do everything one would desire of a total search system. It does, however, partially bridge the gap between manual searches on computer-prepared directories and atom-by-atom, bond-by-bond substructure searching.

LITERATURE CITED.

1. Ofer, K. D., Rice, C. N., Bourne, R. B., and Logan, S. W. Paper presented at the 151st National Meeting of the American Chemical Society. Pittsburgh, Pennsylvania. March 1966.
2. Sorter, P. F., Granito, C. E., Gilmer, J. C., Gelberg, A., and Metcalf, E. A. J. Chem. Doc. 4, 56 (1964).
3. Smith, E. G. Science 131, 142 (1960).
4. Smith, E. G. Revised Rules of W. J. Wiswesser's Line-Formula Chemical Notation. (Copyright 1965.) To be Published.
5. Backus, J. W. The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference Information Processing. Proceedings of ICIP. Paris, 125-132 (1959). UNESCO, Paris.
6. Marron, B., Bolotsky, G., and Tauber, S. J. J. Chem. Doc. 6, 72 (1966).

APPENDIX

QUESTION LIST SYNTAX IN BACKUS NORMAL FORM (BNF)

$\langle \text{QLIST} \rangle := \langle \text{Q} \rangle \mid \langle \text{QLIST} \rangle \langle \text{Q} \rangle$

$\langle \text{Q} \rangle := \langle \text{HCARD} \rangle \langle \text{QBODY} \rangle \langle \text{ANDSPECS} \rangle \mid \langle \text{MCARD} \rangle$

$\langle \text{HCARD} \rangle * := \langle \text{MCARD} \rangle \mid \langle \text{NULL} \rangle$

$\langle \text{MCARD} \rangle * := \langle \text{ACARD} \rangle \mid \langle \text{DCARD} \rangle$

$\langle \text{ACARD} \rangle * := \text{'ADDU'} \langle \text{BLIST} \rangle \langle \text{SYSID} \rangle$

$\langle \text{DCARD} \rangle * := \text{'DELU'} \langle \text{BLIST} \rangle \langle \text{UID} \rangle$

$\langle \text{SYSID} \rangle := \text{name and department number of user}$

$\langle \text{UID} \rangle := \text{user identification, provided by the system when a new user is added}$

$\langle \text{BLIST} \rangle := \text{' ' } \mid \langle \text{BLIST} \rangle \text{' '}$

$\langle \text{QBODY} \rangle := \langle \text{QCARD} \rangle \mid \langle \text{QBODY} \rangle \langle \text{QCARD} \rangle$

$\langle \text{QCARD} \rangle * := \langle \text{USERID} \rangle \langle \text{QPHRASE} \rangle$

$\langle \text{USERID} \rangle := \langle \text{UID} \rangle \langle \text{QID} \rangle$

$\langle \text{QID} \rangle := \langle \text{LETTER} \rangle \mid \langle \text{DIGIT} \rangle$

$\langle \text{QPHRASE} \rangle := \langle \text{ORPHRASE} \rangle \langle \text{NOTPHRASE} \rangle \mid \langle \text{NOTPHRASE} \rangle$

$\langle \text{ORPHRASE} \rangle := \langle \text{SYMSTR} \rangle \mid \langle \text{ORPHRASE} \rangle \text{' , ' } \langle \text{SYMSTR} \rangle$

$\langle \text{NOTPHRASE} \rangle := \text{' . ' } \langle \text{ORPHRASE} \rangle \mid \langle \text{NULL} \rangle$

$\langle \text{SYMSTR} \rangle := \langle \text{SYMBOL} \rangle \mid \langle \text{SYMSTR} \rangle \langle \text{SYMBOL} \rangle$

$\langle \text{SYMBOL} \rangle := \langle \text{FXDSYM} \rangle \mid \langle \text{LOGIC EXP} \rangle$

$\langle \text{FXDSYM} \rangle := \langle \text{SPEC CHAR} \rangle \mid \langle \text{QID} \rangle$

$\langle \text{LOGIC EXP} \rangle^* := \langle \text{VARSYM} \rangle \mid \langle \text{TYPE} \rangle \langle \text{OBJECT} \rangle$
 $\langle \text{OBJECT} \rangle := \langle \text{VARSYM} \rangle \mid \langle \text{FXDSYM} \rangle \mid \langle \text{ORLIST} \rangle$
 $\langle \text{TYPE} \rangle := \text{'\#'} \mid \text{'>'} \mid \text{'<'}$
 $\langle \text{ORLIST} \rangle := \text{'('} \langle \text{FXDSYMLIST} \rangle \text{')'}$
 $\langle \text{FXDSYMLIST} \rangle := \langle \text{FXDSYM} \rangle \mid \langle \text{FXDSYMLIST} \rangle \text{' , ' } \langle \text{FXDSYM} \rangle$
 $\langle \text{VARSYM} \rangle := \text{'='} \mid \text{'@'} \mid \text{'\%'} \mid \text{'\$'}$
 $\langle \text{LETTER} \rangle := \text{'A'} \mid \text{'B'} \mid \text{'C'} \mid \dots \mid \text{'Z'}$
 $\langle \text{DIGIT} \rangle := \text{'0'} \mid \text{'1'} \mid \dots \mid \text{'9'}$
 $\langle \text{SPEC CHAR} \rangle := \text{'\&'} \mid \text{'-'} \mid \text{' ' } \mid \text{'/'}$
 $\langle \text{ANDSPECS} \rangle^* := \langle \text{COMMAND} \rangle \langle \text{ANDLIST} \rangle$
 $\langle \text{COMMAND} \rangle := \text{'SHOW'} \mid \text{'GFIL'} \mid \text{'NMBR'} \mid \text{'LIST'}$
 $\langle \text{ANDLIST} \rangle := \langle \text{UID} \rangle \langle \text{QIDLIST} \rangle$
 $\langle \text{QIDLIST} \rangle := \langle \text{QID} \rangle \mid \langle \text{QIDLIST} \rangle \langle \text{QID} \rangle$

* Each of the asterisked definitions is further specified as being a physical input record; i.e., punched card in present implementation.

PERMUTATIONS AND CLASSIFICATION NUMBERS

Alan Gelberg
Diamond Alkali Company

As presented earlier by Howard Bonnett, and to be presented tomorrow by A. J. Barnard, the Wiswesser line notation lends itself well to assisting in the classification of organic structures. Structures can be ordered: (1) in listings; (2) on computer tape as aliphatics, benzenoids, and multicyclics; (3) according to elemental compositions; and (4) as double bond counts.

Structural organization can be accomplished for catalogs or any other compendia of chemical structures. A chemist can rapidly affix the appropriate classification assignment by reviewing the drawn structure, preparing the notation, and then ordering his file of notations as he wishes.

At the Industry Liaison Office, we had initially used a classification assignment to order our punch cards. At that time, we were working with alpha-numerically arranged listings of notations. The punch cards used to generate the lists were also used to locate fragments and group classify ring systems. This procedure frequently caused frustration and time loss because of the size of the file and the rapid growth rate.

Since the notation represents all of the atoms in a chemical compound, we pondered as to how to use these symbols without additional human effort. The problem of searching notation symbols for specific functional groups was recognized to be analogous to the selection of keywords in an ordinary list of the titles of scientific papers. Pete Sorter suggested that notations be treated in a similar manner as that proposed by H. P. Luhn for the creation of the entries in the journal called Chemical Titles; i.e., permuted indexes.

It became apparent that a list of permutations of chemical line notations alphabetized on individual symbols could be used to readily locate all compounds containing any specified functional group as well as specific compounds and specific classes of carbocyclic or heterocyclic structures.

Using part of a typical page of the chemical compounds in the Pesticide Index, 3rd Edition, the utility of a permuted listing is apparent. The nine compounds are symmetrical triazines, T6N CN ENJ. Reading across the notation and down the page, we see an organized table of

substitutions at the b, d, and f locants. This table, as seen on the next page, can be rapidly scanned for finding alkoxy triazines, chloro-substituted materials, etc., or can be arranged into a Markush tabulation.

The pyridines cited in the Pesticide Index are similarly organized whether they initiate the notation or are connected to another ring system. The pyridines are T6NJ entries.

Since my company is interested not only in the reaction products, but also in the reactants that were used to prepare the products, I include the reactants in my listings. Any other peripheral information such as catalysts, solvents, etc., can also be included if desired.

An average of five to six entries for each notation is generated. This is expanded to almost 10 entries when the reactants are included.

At Diamond, using the IBM-1401, we found that 5,991 compounds in notation generated 33,080 entries. Input time was 1 hr, 15 min. Alpha-numeric sorting time was 30 min. Printing time was 55 min. In under 3 hr machine time, the cost per compound (entered 5.5 times in a listing) was \$0.0275.

At Edgewood, using the Honeywell 200/400, 6.3 entries were created per compound for a file of 55,000 items. Greater machine capability and more sophisticated programming reduced the cost per compound to \$0.015.

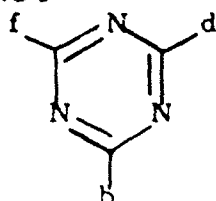
As you will recall, Charlie Granito stated earlier that you could register compounds for approximately \$0.20 per item. By raising this to \$0.215 per compound, you could get registration and an index of permuted notations.

Since the file at Diamond is small, about 10,000 compounds, we can afford the luxury of updating our file frequently. For larger files, it may be impractical to update more than once a year. An index of 50,000 compounds could be frozen and a second index could be started at compound No. 50,001. Both indexes could be used as a chemist would use the cumulative indexes of Chemical Abstracts and then the annual indexes in a search. Since all of the indexes would be on tape, a cumulative index could be prepared if desired (perhaps at the 500,000 mark or 1,000,000 mark). The frequency of doing this would depend upon economics.

Table. Substitutions at the b, d, and f Locants

<u>Pesticide Index 3rd edition</u>	<u>Page and position</u>	<u>Quick-Scan</u>	<u>Reported activity</u>	<u>Wiswesser Notation</u>
P13	189B	GMNNNNT	HERB	T6N CN ENJ BN2&2 DMY FG
P13	281A	GMNNNNT	HERB	T6N CN ENJ BN2&2 DM2 FG
P13	069A	GNNNNNT	HERB	T6N CN ENJ BN2&2 DN2&2 FG
P13	243A	MMNNNOT	HERB	T6N CN ENJ B01 DMY FMY
P13	206A	MMNNNOT	HERB	T6N CN ENJ B01 DMY FM1
P13	040A	MMNNNOT	HERB	T6N CN ENJ B01 DMY FM2
P13	256A	MMNNNOT	HERB	T6N CN ENJ B01 DM2 FM2
P13	205A	MMNNNOOOT	HERB	T6N CN ENJ B01 DM301 FM301
P13	205C	MNNNNNOT	HERB	T6N CN ENJ B01 DN2&2 FMY

Markush Tabulation



		<u>b</u>	<u>d</u>	<u>f</u>
P13	189B	—N(C ₂ H ₅) ₂	—NHCH(CH ₃) ₂	—Cl
P13	281A	—N(C ₂ H ₅) ₂	—NHC ₂ H ₅	—Cl
P13	069A	—N(C ₂ H ₅) ₂	—N(C ₂ H ₅) ₂	—Cl
P13	243A	—OCH ₃	—NHCH(CH ₃) ₂	—NHCH(CH ₃) ₂
P13	206A	—OCH ₃	—NHCH(CH ₃) ₂	—NHCH ₃
etc.				

The file in the Industry Liaison Office has more than 100,000 compounds permuted into an index containing over 600,000 entries. This system is reported to be working very well.

The permuted indexes at Hoffman-LaRoche contain between 50,000 to 100,000 compounds and is also an operating system.

Other organizations are expressing an interest in indexes of permuted notations. Those of us using this approach have found that it meets our needs, is inexpensive to operate and maintain, and is practically fool-proof. Other advantages are that the machine processing time can be used in periods of low activity as fill-in work; searches are performed at a desk with an almost immediate response time to the inquirer; and, as Charlie Granito hopes to establish by early next year, the notation can be used for tape storage and retrieval studies. This program should provide recommendations for further synthesis by our research chemists and will make the file a research tool and not just a repository.

SOME TECHNIQUES FOR THE MACHINE MANAGEMENT OF SMALL CHEMICAL DATA SYSTEMS

A. J. Barnard, Jr., W. C. Broad, and C. T. Kleppinger
J. T. Baker Chemical Co.

and

W. J. Wiswesser
Fort Detrick

The economic machine management of chemical information and data in small or special systems presents as real a challenge as that for large or general systems. It is the purpose of this paper to delineate the continuing studies at J. T. Baker Chemical Company in the machine management of a user-oriented chemical-data system. It is believed that many of the techniques introduced in these studies with a system of modest size are applicable to larger ones. Some aspects of the J. T. Baker studies represent the implementation of approaches suggested by W. J. Wiswesser as early as 1953¹ and have been the subject of previous publications.²⁻⁵ These studies have been paralleled by additional collaborative research at Fort Detrick, Maryland.

BACKGROUND OF THE J. T. BAKER STUDIES.

The direction of the J. T. Baker Chemical studies is influenced by the special character of the company's offerings. The J. T. Baker Chemical Company for over 60 yr has produced and merchandized chemicals of defined purity with informative labeling. This company has a large stake in the laboratory chemical market throughout the world. The ultimate consumer in this market are scientists, technologists, and students. They purchase chemicals, offered as chemicals, for chemical and diverse uses.

The J. T. Baker offerings to the laboratory market as of September 1966 amounted to over 5,700 chemicals, about 5,200 of which are compounds of carbon. In contrast, in 1963, the offerings numbered under 1,000. In the intervening years, changes in the laboratory market, especially in the research and development sector, had made it clear that the offering of research grade liquified and compressed gases and many organic research chemicals was timely. Catalogs of the company issued after this line expansion brought to the user not only systematic chemical names, specifications, and commercial information, but as valuable aids extensive cross-referencing and an empirical formula index for organic compounds.

Early in this line growth, it was clear that new techniques and approaches were needed to cope with the technical and commercial information and data being developed. It was recognized that efficient machine management might make possible the placement of additional data in the hands of users at low cost. One of the capabilities of small-to-medium sized data systems, not always realizable with larger systems, for economic or operational reasons, is the printout of portions of the stored information in various classified arrangements. Subsequently, many retrieval requests can be answered directly by inspection of such listings. By offset printing, such classified listings can be made available to many users. A large portion of the J. T. Baker studies is directed to the economic use of this listing capability.

ELEMENTS OF A CHEMICAL DATA SYSTEM.

The "hard-core" elements of a machine-managed chemical-data system can be described as:

1. Structure diagram or its delineation (e.g., by a unique notation).
2. Empirical formula (in sortable form).
3. Chemical or common name (in computer-oriented typography).
4. Synonyms and trade names.
5. Indexing and classification numbers, codes, and terms.
6. Registry numbers.
7. Biophysicochemical constants and data.

If the system is to have a bibliographic aspect, "literature citations" would be added to this list. The treatment of topics in this paper parallels the order of seven "hard-core" elements in this list.

WISWESSER LINE NOTATION.

The Wiswesser line notation was adopted in the J. T. Baker studies because it offered a unique, unambiguous, concise description of a chemical structure with characters that are available in standard electronic-data processing equipment.^{6,7} The J. T. Baker offerings of organic

compounds to a large extent are those most readily accessible and best characterized and, consequently, are relatively simple. Some statistics as to the size of Wiswesser line notations assigned to almost 5,000 such nonpolymeric organic compounds are presented in table I.

Table I. Distribution of Notation Size for 4,982 Organic Compounds
Assigned Wiswesser Line Notations
(J. T. Baker Commercial Deck, October 1, 1966)

Size designation (S)	Number	Cumulative	Size designation (S)	Number	Cumulative
		%			%
1	1	0.0	J	20	95.1
2	2	2.4	K	21	95.6
3	3	10.5	L	22	96.2
4	4	18.7	M	23	96.7
5	5	28.8	N	24	97.1
6	6	38.3	O	25	97.4
7	7	48.6	P	26	97.6
8	8	57.8	Q	27	97.8
9	9	65.2	R	28	98.1
Ø	10	71.7	S	29	98.4
A	11	76.9	T	30	98.5
B	12	81.6	U	31	98.8
C	13	85.0	V	32	99.1
D	14	87.1	W	33	99.3
E	15	89.1	X	34	99.4
F	16	91.0	Y	35	99.4
G	17	92.5	Z	36	99.6
H	18	93.4		37-44	100.0
I	19	94.3			

Attention is directed to the finding that the notation for 85% of these compounds required 13 or fewer columns of an IBM card. The arithmetic mean was 9.0 columns per notation and the calculated median of the distribution was 7.2 columns per notation. It is expected that this notation-size distribution will not shift markedly as the collection grows since both simple and complex structures are probable additions and certain simple systematic contractions for fused ring structures⁷ are still to be exploited fully.

Recently, the permuted line notation program of Edgewood Arsenal, developed by Sorter and coworkers,⁸ has been applied to the collection. The printout, in addition to the previously employed elements of a registry number, QUICK SCAN Index, and a line notation, includes the BATCH number and the computer-oriented chemical name (see below). The latter feature, in the short time this permuted index has been available, has proven of salient value since technologists with only a mere inkling of understanding of the line notation can find the appropriate page or pages of the printout and then use the name as a familiar tool to focus the search to the desired degree. It may be of interest to note that with a computer input of 4,982 cards having permutable Wiswesser line notations, a total output of 20,171 lines resulted, corresponding to 4.1 permutations per card.

Admittedly, the use of a conventional IBM card sorter to retrieve compounds of desired structural features from column-by-column sorting of a Wiswesser line notation field can be tedious, even with a small collection of some 5,000 cards. The use of the BATCH number as a "screen," usually reduces the cards remaining to be sorted to reasonable numbers. In addition, by keeping these cards segregated appropriately according to values of the B, A, and T digits of the BATCH number, the strategy employed in the examination of the notation field can be varied to reduce the sorting time. For example, based on BATCH number and line notation sorts, all hydroxy compounds and all cyclic nitrogen compounds in the J. T. Baker collection have been retrieved. Each of these retrieval tasks, including the punching of a satellite deck for the retrieved cards, required less than 90 min on the part of a technologist. On such sorts has been based the development of special BATCH directories for hydroxy compounds and cyclic nitrogen compounds.⁹ It may be added that the empirical formula also on occasion can be used as an effective screen.

STRUCTURE CARD LAYOUT.

Table II presents the IBM-card layout for "structure" cards in the J. T. Baker studies.

The organization and use of this card with organic compounds has been described fully.² Note that the Wiswesser line notation can be allowed to overlap the name field. Where the line notation exceeds 13 columns, it can be interrupted in that column by an asterisk and the full notation be carried on a second card. The two cards are distinguished by control punches in column 80. The cards with interrupted notations are placed in the deck when clean, ordered printing of the chemical names is sought.

Table II. Structure IBM-Card Layout for Organic and Inorganic Compounds

Organic Compounds			Inorganic Compounds	
Columns			Columns	
1-9	Wiswesser line notation Prefix, chemical name Organic chemical name BATCH number Size, notation (S) Space C punched, carbon compounds Number of carbons H punched, hydrogen compounds Number of hydrogens Remainder, empirical formula Registry and control field	1 to 9	1	Space
10		Ø	2-22	Computer - oriented line formula
11-13		A to C	23-45	Inorganic chemical name
14-22		D to L	46-50	* ABET number
51			51	Size, line formula (S)
52			52	Space
53			53	Space
54-56			54-72	Empirical formula
74-80			74-80	Registry and control field

The layout shown in table II can be described as the "universal" one, since for other uses the empirical formula field can be suppressed in the punching of a satellite deck to yield a field of 22 columns. In addition, if the line notation is suppressed, a second field of 13 columns is obtainable.

EMPIRICAL FORMULA FIELD.

A field of 21 columns has been found sufficiently large for the J. T. Baker studies, using the practices that have been developed.² The treatment of carbon and hydrogen is noteworthy (cf. table II). Space is provided for C₉₉₉H₉₉₉; however, seldom will a nonpolymeric compound exceed C₉₉H₉₉. Consequently, one space can be used as a separator. The value of the empirical formula field can be enhanced by first sorting the collection according to each value of the A digit (atomic classes) of the BATCH number. By a sort in column 61 for B, C (that is, of BR and CL), F, and I, all halogens are isolated from compounds that also may contain nitrogen, oxygen, phosphorus, and sulfur. Further sorts are only required for an A digit value of 9, which corresponds to the presence of the other (rarer) elements such as arsenic, boron, and metallic elements. Such a sorting approach has been utilized for the development of a special BATCH directory for halogen compounds.⁹

Although not considered for the J. T. Baker studies, some merit exists for adopting the single-letter G and E symbols of the Wiswesser line notation for chlorine and bromine in the machine management of empirical formulas. In this way, all the halogens have the same sorting pattern and better alignment of formulas is secured in listings.

COMPUTER-ORIENTED CHEMICAL NAMES.

As mentioned above, an important capability of the small chemical data system is the possibility of listing stored information in various classified arrangements. A vital element of any such listing, if it is to find use by many chemists, is a recognizable chemical name. Systematic chemical names present various typographical complexities; consequently, special practices must be adopted for the transcription of such names into the limited typography of punched cards and standard computers. Barnard, Kleppinger, and Wiswesser³ have studied this transcription problem and have recommended restriction of the characters employed to capital letters, Arabic numbers, and just four additional signs, - & / *, all of which are widely available on tabulating and electronic processing equipment. Incidentally, these also are the typographical restrictions placed on the Wiswesser line notation.

In addition to the mere transcription of a chemical name, it must be kept within reasonable length so that invested capacity in the punch-card or its computer equivalent is reasonable, and economy of space is secured in extensive listings. The twofold goal is, therefore, (1) to transcribe the chemical name into computer-oriented typography, and (2) to contract it, where necessary, without loss of structural information and still achieve a result capable of being read by the interested chemist after only brief study of the practices adopted. Full details on how so called "computer-oriented" chemical names are devised have been published³ and a checklist of frequently required contractions has been made available.¹⁰ Here an attempt will be made only to highlight the approach by the consideration of selected examples.

If Greek letters are spelled out, costly space would be required. A better solution is shown in table III; namely, the use of the corresponding Latin or phonetic or pictorial Roman-letter equivalent followed by an ampersand. For example, alpha is denoted by A&. It will be noted that M&, O&, and P& can be reserved to denote the meta, ortho, and para locants for isomeric phenylene entities.

Table III. Computer-Oriented Transcriptions of Greek Letters*

A&- alpha- or 1st-	O&- <u>ortho</u> -
B&- beta- or 2nd-	OM- <u>omicron</u> - or 15th-
C&- chi- or 22nd-	P&- <u>para</u> -
D&- delta- or 4th-	PI- pi- or 16th-
E&- epsilon- or 5th-	Q&- theta- or 8th-
F&- phi- or 21st-	R&- rho- or 17th-
G&- gamma- or 3rd-	S&- sigma- or 18th-
H&- eta- or 7th-	T&- tau- or 19th-
I&- iota- or 9th-	U&- upsilon- or 20th-
J&- (generic)	V&- (variable or uncertain)
K&- kappa- or 10th-	W&- omega- or 24th-
L&- lambda- or 11th-	or terminal
M&- <u>meta</u> -	X&- xi- or 14th-
MU- mu- or 12th-	Y&- psi- or 23rd- or <u>pseudo</u> -
N&- nu- or 13th-	Z&- zeta- or 6th-

* Arranged in order of the Roman alphabet with an indication of position in the Greek alphabet.

A prime mark can be denoted by an asterisk; a pair of parentheses or brackets by a pair of slash marks polarized with hyphens (that is, - / / -), and hyphens can be used, if necessary, to separate locant letters and numbers in the same manner as commas are customarily employed. These usages are delineated in some of the examples presented.

We stand 100 yr from Frankland's paper in the Journal of the Chemical Society,¹¹ in which he recommended certain abbreviations and symbols for common radicals for the purposes of teaching and for convenience in the presentation of reaction sequences. As shown below in the left column, many of these old traditions can be utilized in the contraction of computer-oriented names by transposing all of the letters to capitals. In the right column, the extensions suggested by Wiswesser in 1953 are listed;¹ PN is now recommended for pentyl.²

<u>Frankland, 1866</u>		<u>Wiswesser, 1953</u>	
ME	METHYL	IPR	ISOPROPYL
ET	ETHYL	IBU	ISOBUTYL
PR	PROPYL	AM	AMYL)
BU	BUTYL	PE	PENTYL) ^{now PN}
HO	HYDROXY	HX	HEXLY
MEO	METHOXYL	HP	HEPTYL
ETO	ETHOXY	OC	OCTYL
	<u>also earlier</u>	NON	NONYL
	PH PHENYL	DEC	DECYL
	BZ BENZOYL	BZL	BENZYL

The examples shown below illustrate how the simple contraction for ethyl, ET, can be used to compact some simple chemical names. In the right column it is made clear how an Arabic numeral 2 can be suffixed to the abbreviation for ethyl to designate diethyl. It will be seen that a contraction is separated from what follows either by a hyphen or, where relevant, by a blank space. Note that the last example on the left column is ethyl dichloroacetate.

ET	IS ETHYL	ET2	IS DIETHYL
ET	ACETATE	ET2	ADIPATE
ET	ACRYLATE	ET2	AMINE
ET	AMINE	ET2	FUMARATE
ET	BENZENE	ET2	MALONATE
ET	BENZOATE	ET2	OXALATE
ET	CL2-ACETATE	ET2	PHOSPHITE

The examples given below illustrate the use of what may be termed a "packaging" operator; namely, the use of hyphens to set off contractions and their associated locants.

1-BR-55ME2-HYDANTOIN
 4-BR-35ME2-PYRAZOLE
 1-3-BR2-55ME2-HYDANTOIN
 3-5-CL2-4-HO-BENZOIC ACID
 1-CL-25MEO2-4-NITRO-BZN
 111555-F6-2-4-PENTANEDIONE
 F18-H10-NAPHTHALENE

With the information that the first example is 1-bromo-5,5-dimethylhydantoin, no difficulty should be experienced in reading the full systematic names for the other compounds. For example, the last compound is octadecafluorodecahydronaphthalene. These examples also delineate the recommended practice of providing nine columns for the variable length prefix to the main name. The latter is confined to 23 columns; consequently, the entire name field requires 32 columns. Some saving of space can be effected abandoning the nicety of an offset prefix;^{1,2} for general purposes, a field of 26 columns can be recommended.

The examples given below exemplify what might be termed the "acylating" or "oxygenating" operator. In brief, the ampersand in these examples oxygenates a contraction.

B&	BORATE	VL&	VALARATE	ADI&&	ADIPATE
C&	CARBONATE	IVL&	ISOVALARATE	AZE&&	AZELATE
I&	IODATE	PIV&	PIVALATE	FUM&&	FUMARATE
N&	NITRATE	HX&	HEXANOATE	MAL&&	MALATE
P&	PHOSPHATE	HP&	HEPTANOATE	MLE&&	MALEATE
S&	SULFATE	OC&	OCTANOATE	MLO&&	MALONATE
P-&	PHOSPHONATE	NON&	NONANOATE	TAR&&	TARTRATE
S-&	SULFONATE	DEC&	DECANOATE	SUC&&	SUCCINATE
		UND&	UNDECANOATE	OX&&	OXALATE
FO&	FORMATE	DODC&	DODECANOATE		
AC&	ACETATE	LAU&	LAURATE	BZ&	BENZOATE
PR&	PROPIONATE	MYR&	MYRISTATE	NAP&	NAPHTHOATE
BU&	BUTYRATE	PAL&	PALMATATE	PHTL&&	PHTHALATE
IBU&	ISOBUTYRATE	STE&	STEARATE	SAL&	SALICYLATE
PN&	PENTANOATE	OLE&	OLEATE	TOL&	TOLUATE

A distinction can be made for the "organo-bonding" of a stated element by the introduction of a hyphen between the symbol of the element and

the ampersand. Thus, P-& and S-& denote phosphonate and sulfonate, respectively. The use of a double ampersand with dibasic acid groups allows their difunctional character to be indicated. The names assigned to the contractions above are for the suffix position; however, these contractions are also applicable in a midname position (for example, FO&- = formyl, ADI&&- = adipoyl).

The next group of examples shows how the contraction for an alkyl or aryl group can be transformed by the addition of an ampersand to designate the corresponding ane or ene hydrocarbon in a midname position. The asterisk here can be termed the "ane/ene" operator.

ME*OL	METHANOL
ET*OL	ETHANOL
PR*DIOL	PROPANEDIOL
BU*ONE	BUTANONE
BU*DIAMIN	BUTANEDIAMINE
BZ*S-&	BENZENESULFONATE
NAP*S-&	NAPHTHALENESULFONATE
ET*S-&A	ETHANESULFONIC ACID

A slash as a suffix or prefix can serve to indicate a cation or anion, respectively. Thus, AMM/ designates the ammonium cation and /AZ the azide anion. The slash here can be described as an "ionizing" operator.

The selected examples below will serve to demonstrate how a pair of slash marks, polarized by hyphens, can represent a pair of parentheses or brackets to enclose a complex function.

2-/BR-ME/-TETRAHYDROFURAN	BIS-/CL-ME/-ETHER
1-3-BR2-2-2-BIS-/BR-ME/-PR	CL-/CL2-ME/-ME2-SILANE
-/2-BR-ET/-ME3-AMM/ BR	1-CL-4-/2-ME-ALLYL/-BZN
-/2-BR-2-ET-BUTYRYL/-UREA	BU3-/CL5-PH-O/-TIN

The last example is read as tributyl(pentachlorophenoxy)tin. The pair of polarized slashes here can be termed the "enclosing" operator.

Finally an attempt can be made to exemplify how a group of contractions can be developed systematically from the contraction for an alkyl group, namely, butyl.

<u>In Main Name</u>		<u>In Suffix</u>	
BU-	BUTYL-	-BU	-BUTANE
BU&	BUTYRYL-	BU&	-BUTYRATE
BU&M	BUTYRAMIDO-	BU&M	-BUTYRAMIDE
BUO	BUTOXY-	BU&A	-BUTYRIC ACID
BU&O	BUTRYLOXY-	BU&H	-BUTYRALDEHYDE
IBU	ISOBUTYL-	BU&I	-BUTYRYL IODIDE
BU*	BUTANE-	BU*OL	-SUTANOL
		BU/	BUTYL (free)
		/BU	BUTYLIDE

Note how the contraction for butyrate, BU&, can be transformed to represent the corresponding amine, acid, and aldehyde by the addition of M, A, and H, respectively. The resulting contractions are readily understood.

Extensive examples of the effective use of computer-oriented names in machine listings are to be found in the J. T. Baker BATCH directories.^{9, 12, 13}

SYNONYMS.

At this time, synonyms in the J. T. Baker system are in the form of hand-manipulated card files and their derived typeset equivalents in commercial catalogs. As the collection grows, these cross-references can be transcribed and contracted to their computer-oriented equivalents when the effort is warranted. It is noteworthy that on the average three to four synonyms have been assigned per compound; this number would be larger if certain types of synonyms were not limited to parent compounds only. For example, salicylic acid would have o-hydroxybenzoic acid as a synonym, but none of its derivatives would be assigned an o-hydroxybenzoic acid or o-hydroxybenzoate cross-reference.

CLASSIFICATION NUMBERS AND CODES.

The principal classification code employed so far in the J. T. Baker studies has been the BATCH number for organic compounds. An elemental classification code for inorganic compounds, the $\pm$ ABET number, also appears promising. These two classification codes and their uses will be described briefly. A direct polymer-use code has also been employed in the preparation of a BATCH directory for polymer-use chemicals.¹⁵

THE BATCH NUMBER.

In 1953, W. J. Wiswesser proposed a Formula Index Number consisting of five simple numeric measures, each having possible values from zero through nine.¹ This structural-atomic classification code was given the mnemonic designation BATCH number. Here, B corresponds to the basis of the compound (i. e., the nucleus), A to the atomic class, T the total heteroatoms, C the carbon count, and H the hydrogen count. The concept received little attention until 1964, when its use was implemented in the J. T. Baker studies. At that time, the assignment of two of the measures was revised, based on an expanded and updated statistical study.² The scheme for the assignment of a BATCH number to a compound of carbon is fully described in a research article² and summarized in the introduction to the J. T. Baker BATCH directories.^{9, 12, 13} No attempt will be made to explain the assignment of BATCH numbers in this article since the earlier publications are readily accessible. The relatively simple scheme can be readily mastered by chemistry-oriented personnel, including secretaries.

In addition to the machine application of the BATCH number, it has proven valuable in the manual organization of card and document files relating to individual organic compounds.⁴ With color-coding of cards for the B, A, and T digits, using color-ink markers, simple structural-atomic sorts can be effected merely by turning up relevant cards in the file.

BATCH numbers also could find application as locants or terms in coordinate indexing or optical-coincidence systems for the control of internal report literature. To distinguish isomers or other compounds having identical BATCH numbers, so that false drops are avoided, an extra digit or letter may be suffixed to the assigned BATCH number.

The BATCH number represents a classification code applicable to the management of a general collection of organic compounds. For special collections, where a high proportion of the compounds fall into a few classes, certain values for the digits in the BATCH number would be overloaded. Under such circumstances, the scheme for the assignment of a BATCH number can be modified. Feeman¹⁴ has described such an approach in the management of a collection of information on commercial dyestuffs, especially anthraquinone dyes.

THE #ABET NUMBER.

Wiswesser has recently elaborated systematic practices for the transcription of conventional line formulas for inorganic compounds into computer-oriented equivalents.⁷ This approach to inorganic formulas now has

been pilot-tested at Fort Detrick in punched-card sorting and listing routines for 5,000 compounds selected from relevant compendia. In the J. T. Baker studies, computer-oriented inorganic formulas now have been written for about 800 commodities, along with computer-oriented chemical names, and empirical formulas. For the inorganic structure cards, the general layout at the right of table II is employed. The name can be restricted to columns 23 through 45 since inorganic names do not have nonindexing prefixes. Some instructive examples of computer-oriented inorganic formulas are shown below:

BA	N-O3*2	KA	G-O4	NA	S-O3-Q
CA	C2 QH2	NA	B-O2 QH1/2	NI	Z&2 S-O4*2 QH6
CA	Q2	NA3	CO- N-O2*6	CA	CE/DY/LA C-O3*3 F2
H	F &AQ	NA2	S	BI	/CD/PB/SN
H3	P	NA2	S-O4 QH10		
H2	S-O4	NA2	S2-O3		

Note that Q is used for the hydroxyl group (as in acid salts of oxyacids) and for the hydroxide ion, and in QH for water (of hydration, crystallization, etc.). Observe that the use of an asterisk allows a following number to be recognized as a quantity multiplier, applying leftward until it is stopped by a blank space. Since the symbol for hydrogen has a single character, in an initial position it is displaced by one column (to column 3) to provide distinctive indexing and immediate recognition for acids and some hydrides. Water of hydration is indicated by QH followed by a multiplier that may be either an integer or a proper fraction. Aqueous solutions are distinguished from hydrates by an appended &AQ. The last two examples demonstrate what can be done for the representation of a mineral in which isomorphous substitution is possible and of an alloy.

Studies are under way on the use of such computer-oriented chemical formulas. It is clear that they are readily sortable into alphanumeric order and, therefore, offer indexing possibilities.

Such computer-oriented line formulas, written according to the established rules,⁷ allow the elaboration of a classification code for elements, inorganic compounds, and alloys. One such code, the *ABET number is under joint study at Fort Detrick and J. T. Baker. In structure cards, this elemental code can be punched in the same field as the BATCH number and is distinguished from it by the presence of a mark other than an integer in the first column of this field (see table II, right side).

Computer-oriented line formula	*ABET number	Computer-oriented line formula	*ABET number	Computer-oriented line formula	*ABET number
NA	*1--1	KA G-04	*1/66	H3 P	-15-1
NA Q	*16-2	NA3 CO- N-02*6	*1950	H F	-17-1
NA2 OO	*16-4	NA Q2	*26-3	H2 S-Q4	-1665
NA2 S	*16-3	UR--02 N-03*2 QH6	*6650	HG N-03*2 QH	-2569
NA S-03-Q	*1666	FE S	*86-2	PB B-02*2 QH	-4367
NA2 S-04	*1667	FE2 O3	*86-5	Z&2 CR2-07	-5660
NA G	*17-2	FE3 O4	*86-7	I F5	-77-6

The value of the *A digit corresponds to the periodic (sub)group into which the first cited element falls. Invariably this is the most electropositive species. The value of the B digit corresponds to the group into which the second element falls. The value of the E digit is the group into which the third (electronegative) element falls. The T digit value corresponds to the total nonhydrogen atoms present in the formula, omitting molecules of solvation (water of hydration). A zero is recorded for 10 or more such atoms. For a binary compound, the E digit, for example, is a hyphen; for an element, both the B and E digits are hyphens.

REGISTRY NUMBERS.

In the J. T. Baker studies, the J. T. Baker commodity number has been used as the registry number. For most organic compounds this consists of a letter and three digits and is assigned closely alphabetically. As a dividend, listings according to the registry number place the entries in alphabetic name order. The assignment of registry numbers based on alphabetized names has considerable merit when the names themselves are based on simple principles.

THE BATCH DIRECTORY CONCEPT.

As mentioned early in this paper, the feasibility of listing a portion of the information in classified arrangements is one prominent capability of small chemical data systems. This capability has been exploited in the J. T. Baker studies. For the retrieval of organic compounds by structural-atomic elements, the listing of the computer-oriented chemical name and other information (at least a registry number) according to the BATCH number has proved to be a most valuable tool.² So-called BATCH directories for the J. T. Baker offerings of organic compounds, prepared by low-cost offset printing of line-printer listings, have been made available to thousands of laboratory workers.^{9, 12, 13} The discrimination of the BATCH number as a retrieval tool

can be enhanced by multiple listings with its various digits treated as primary, secondary, etc. A listing with the BATCH number treated as a conventional five-digit number, that is with the B digit primary, and A digit secondary, etc., gives precedence to the nature (and number) of rings present over the type and number of atoms present. A listing with the A digit treated as primary, the T digit as secondary, the B digit as tertiary, etc., gives precedence to the type and number of atoms present over the nature (and number) of rings present.

The listing of compounds by BATCH number within use, property, or activity classes places on the desk of the researcher a simple but powerful correlation tool. This approach is exemplified by the J. T. Baker BATCH directory for polymer-use chemicals.¹³ Some 2,000 compounds from the J. T. Baker offerings were selected on the basis of functional groups and the knowledge of polymer chemists as having polymer-use possibilities and were assigned within 12 polymer-use classes. For example, this directory allows the polymer researcher to select from about 180 available compounds listed in the "catalyst" class those having structural-atomic aspects appropriate to the problem at hand.

REMARKS.

In summary, the J. T. Baker studies are exploring the effective machine management of small-to-moderate sized systems of chemical data. The efforts are to a great extent user-oriented. It is believed that many of the tools investigated so far and described in this paper are directly applicable to more diverse collections of chemical data. Such tools include the Wiswesser line notation and computer-oriented inorganic line formulas, easily mastered classification codes such as the BATCH number, computer-oriented chemical names, and machine-produced classified directories.

LITERATURE CITED.

1. Wiswesser, W. J. Literature Sources of Mammalian Toxicity Data, With Special Emphasis on Tabulating Machinery Applications. Advances in Chemistry Series No. 16. pp. 64-82. American Chemical Society. Washington, D. C. 1956.
2. Barnard, Jr., A. J., Kleppinger, C. T., and Wiswesser, W. J. Retrieval of Organic Structures From Small-to-Medium Sized Collections. J. Chem. Doc. 6, 41-48 (1966).
3. Barnard, Jr., A. J., Kleppinger, C. T., and Wiswesser, W. J. Computer-Oriented Chemical Names. J. Chem. Doc. 6, 48-57 (1966).

4. Barnard, Jr., A. J., and Wiswesser, W. J. Use of the BATCH Number With Hand-Manipulated Files. *J. Chem. Doc.* 6, 188-189 (1966).

5. Barnard, Jr., A. J., and Wiswesser, W. J. Some Innovations in Chemical Information Management. *Information Retrieval Letter* 2, 1-3 (1966).

6. Wiswesser, W. J. A Line-Formula Chemical Notation. Thomas Y. Crowell Co. New York. 1954.

7. Smith, E. G. Wiswesser's Line-Formula Chemical Notation. McGraw-Hill Book Co. New York. (In press.)

8. Sorter, P. F., Granito, C. E., Gilmer, J. C., Gelberg, A., and Metcalf, E. A. *J. Chem. Doc.* 4, 56-60 (1964); Granito, C. E., Gelberg, A., Schultz, J. E., Gibson, G. W., and Metcalf, E. A. *Ibid.* 5, 52-55 (1965); Granito, C. E., Schultz, J. E., Gibson, G. W., Gelberg, A., Williams, R. J., and Metcalf, E. A. *Ibid.* 5, 229-233 (1965).

9. J. T. Baker Special BATCH Directories for Halogen Compounds, Hydroxy Compounds, and Cyclic Nitrogen Compounds. J. T. Baker Chemical Co. Phillipsburg, New Jersey. January 1966.

10. Checklist of Transcriptions and Contractions for Computer-Oriented Chemical Names. Technical Information Service. J. T. Baker Chemical Co. Phillipsburg, New Jersey. July 1966.

11. Frankland, E. Contributions to the Notation of Organic and Inorganic Compounds. *J. Chem. Soc.* 19, 372-395 (1864).

12. J. T. Baker BATCH Directory. J. T. Baker Chemical Co. Phillipsburg, New Jersey. September 1965.

13. J. T. Baker Special BATCH Directory for Polymer-Use Chemicals. J. T. Baker Chemical Co. Phillipsburg, New Jersey. September 1966.

14. Feeman, J. F. A Novel Organizational Code for Organic Structures Based on Functional Groups. *J. Chem. Doc.* 6, 184-187 (1966).

THE "DOT-PLOT" COMPUTER PROGRAM

William J. Wiswesser
Fort Detrick

This oral presentation was especially designed by the Graphic Arts Section at Fort Detrick to show you in 20 min what a line-printer can do in less than 2 min, when it is directed by a standard computer with the "Dot-Plot" computer program.*

The "Dot-Plot" program obtains its name from the fact that the input for the printing coordinates consists of a rectangular grid in which the plotting positions are visible as holes or dots in the punched card. Thus, the hexagons (diagram 1 B**) or the pentagon (diagram 1 C) are instantly recognizable in the dot-plotting part of the card. As you will see, with examples that also show printing versatility, this program was developed for UNIVAC SS-90, Type 2 computers (diagram 1 E) to perform very high-speed printing of any graphical or tabular data (diagram 2 A) with general purpose standard equipment (diagram 2 B) from single cards holding all the information for a 6 by 30 grid (diagrams 2 C and 2 D). This diagram (2 D) shows the maximum limits of the dot-plot grid area for a single card. (A second card provides capacity for another six lines.)

Six rows or lines of print and 30 columns provided 180 plotting positions in which to place a maximum of 40 information marks (diagrams 2 E and 3 A). This flexibility is adequate to accommodate a graphical notation system for describing the chemical structures of some 90% of the commonly met synthesis and screening compounds.

The grid is scanned exactly like a TV picture: left to right, top to bottom (diagram 3 B). The computer completes its synthesis of a diagram or table by picking off the printing characters that are stored in another part of the card and dropping them one by one into the plotted positions. Thus, the dot-plots are like windows through which any characters, or blank spaces, may

* The "showing" in the oral presentation was done with 80 slides in a fast-switching Kodak "Carousal" projector. These are photographs of the 28 pages of diagrams that the computer delivered in a mere minute and forty seconds.

** Diagrams 1 to 28 are shown in appendix.

be displayed. The line-printer does its composing work at lightning speed, gushing out more than 16 pages of tables or diagrams (and five per page) in 1 min. This actually is faster than straight line-printing speed, because the paper feeder heaves the paper past the blank spaces that separate the diagrams. The secret of this high-speed performance is, as in many other mechanical devices, the straightforward simplicity of the program instructions (written by A. James Dukes, a clever programmer who has intimate familiarity with the hardware and software of this UNIVAC computer).

This left-to-right and top-to-bottom scanning, incidentally, explains how chemical line-formula descriptions started 100 yr ago (diagrams 3 D and 3 E): carbon skeletons stood erect, like human forms, and all of the attachments were shown to the right (never to the left). Thus, a typical 19th century structure diagram (4 A) shows all "appendages" drawn to the right of each vertically aligned carbon segment. The century-old line-formula notations are simple left-to-right and top-to-bottom scanning delineations of this vertical alignment. (Our computer-oriented line notations thus have no novelty except the use of single-mark symbols for the most frequently occurring structural units.)

The next two diagrams (4 B and 4 C) show such vertically drawn C-skeletons for the first 13 alkanes. This is just the first of five different ways these same dot-plot patterns can be labelled. Both of these diagrams (4 B and 4 C) use up the limit of six lines, and not quite all of the 40 columns of character storage.

In horizontal scanning, the far left was the beginning, just as it is in typewriting (diagrams 4 D and 4 E).

ALL GRID MARKS ARE STORED THIS WAY IN TOP OF CARD (diagram 5 A) without any spaces between the words. The card-punching instructions are very simple: first punch out the printing marks, then set places with the dot-plot (diagrams 5 B and 5 C). Each letter string can shift and swing, as shown in the next sequence of diagrams (5 E to 6 E). Nothing was changed in these duplicated cards except the positions of the dots.

Binary bits can be explained as a stacking of values in six horizontal "tape channels" (diagram 6 E), and this same orientation (diagrams 7 A, 7 B, and 7 C) shows the vertical summing of binary bits to make total counts that read stepwise from left to right. Binary arithmetic is a pertinent part of the numeric analysis of alkanes, as is shown in a later set of diagrams.*

* See diagrams 12 A to 12 E, appendix.

Next are shown some simple chemical diagrams wherein Q is OH, L is CH_2 , Y is a branched CH, and D is a double-bonded linear CH (diagrams 7 D and 7 E). These diagrams show, besides the figure in the center, a related (abbreviated) linear notation in the upper left, and a chemical name in the lower right. The related notation (diagrams 8 C to 8 E and 9 A to 9 C) uses an ampersand for the "dropped" YQ group and a slash or virgule for the "lifted" YQ group. These first two chemical diagrams (8 C and 8 D) are the two mirror-image forms of glycerose, the simplest sugar; and of ribose (diagrams 8 E and 9 A). Thus, D-glucose is related to D-galactose as shown in the aldose diagrams (9 B and 9 C).

The A-L-Y-X notation symbols used in these diagrams also can "fill in the dot-plot windows" of the first 13 alkanes previously shown with C-marks (diagrams 4 B and 4 C). These latter symbols or equivalent valence numbers (diagrams 9 D and 9 E) completely specify the associated H-atoms (letters in diagrams 10 A and 10 B, and numbers in diagrams 10 D and 10 E), thus, no additional marks are needed to show hypothetical postulations called "double bonds" (removal of one pair of H-atoms between connected groups) or "triple bonds" (removal of two pairs of H-atoms between connected groups). When the dot-plot "windows" are labelled with valence numbers (diagrams 10 D and 10 E), these strings of numbers provide all that is necessary for the mathematical connection tables. (Details are given later.)

All possible branching of alkanes is a binomial variation of just X or Y junctions. This elementary feature is emphasized by lettering only these junctions and merely "spotting" the unbranched C-groups with asterisks (diagrams 11 B and 11 C). This kind of alkane analysis, a routine permuting of CH_2 -links between the branches and their associated CH_3 terminals, led to new sets of binomial coefficients (diagram 12 A) that are natural extensions of the well-known "Pascal's Triangle" (diagram 11 E). This particular new set (diagram 12 A) predicts the number of possible X, Y-branching arrangements, from "all X" to "all Y", when different kinds of branching positions are added to the initial two that are identically equivalent. (A full explanation of these symmetric functions and their chemical application is a lecture in itself.)

New sets of partition-counting terms (diagrams 12 C and 12 D) then count the number of possible alkane isomers when CH_2 -groups are partitioned around and between the X, Y-branching points. Thus, the top line in diagram 12 C counts the disoalkane isomers, the next (YX) line, the isoneoalkane isomers, and the third (XX) line, the diisoalkane isomers, and so on for higher combinations (diagram 12 D).

Symmetry symbols, placed in the same original dot-plot "window" patterns for the first 13 alkanes (diagrams 13 A and 13 B) illustrate the elements of modern group theory in this alkane analysis, with different letters A, B, C, D, etc., now identifying different kinds of places. (Group theory is a basic part of the mathematical arsenal used by spectroscopists to resolve fine structural details from molecular absorption and emission spectra.)

These five different ways of labelling dot-plots of the alkanes now are illustrated with more complex examples. The first set (diagrams 13 E to 14 E) shows an indiscriminating C-letter at each printing place. The second set (diagrams 15 A to 16 B) shows the discriminating power that can be incorporated in the linear recitation of pieces when branches are identified as Y or X letters. The third set (diagrams 16 D to 17 D) shows complete discrimination with the A and L letters. The fourth set (diagrams 18 A to 19 A) matches these with corresponding valence numbers. The fifth set (diagrams 19 C to 20 C) explains the fundamental symmetry analysis by using different letters for different kinds of positions. These diagrams also have auxiliary information shown in the corners:

1. The lower left number in all cases is the number of C-atoms.
2. The lower right number in all cases is the specific isomer number.
3. The upper left marks are delineated definitions of the structure, either as line-formula (Y, X) notations or as connection-table numbers.
4. The upper right marks in the first two (C and ?) sets are Y/X identifications for the partitioning (isomer-counting) series and in the remaining three sets these are the line-formula notations.

All of these markings thus are different measures for various aspects of the simplest open-chain structures.

Branched branches have distinctive spacing patterns that emphasize the proper branching connections, as illustrated in the next diagrams (21 A to 22 A). Thus, the branching points (Y or 3 and X or 4) are separated and the terminal groups (I-marks) are brought closer to the junctions to avoid ambiguous implications about connections, regardless what symbols are used.

Two of the dot-plot symbols have two mnemonic aids (diagrams 22 B to 22 E): D is a Diatomic (as well as "double-bonded" or dehydrogenated) linear CH-group and T is a T-branched (tertiary) C-atom with a

fixed or floating double bond. The second associations appear with aldehydes and ketones, wherein D-O is alDo and T-O is keTo. (Experimental studies on remembering and forgetting, as reported in the March 1964 issue of the Scientific American, show that slow or fast learners improve their memory storage with multiple association.)

Aspirin and cyclooctatetraene illustrate monocyclic dot-plots (diagrams 23 A and 23 B). As noted earlier, the upper left marking is a related linear notation, and the lower right marking is the common or trivial name identification.

The bicyclic 5,7-ring frame in azulene and vetivazulene is recognized easily in its dot-plot (diagrams 23 C and 23 D). Other bicyclic examples with 5,6-ring patterns (diagrams 23 E and 24 A, piperonal and saccharin), or 6,6-ring patterns (diagrams 24 B and 24 C, naphthalene and tetralin, and diagrams 24 D and 24 E, quinoline and quinoxalone) show a negligible geometric distortion. In fact, the line-printer's pica type spacing provides geometrically regular hexagons when plotted as shown.

The symmetry of tricyclic structures such as carbazole and phenanthrene (diagrams 25 A and 25 B) is clearly evident in these dot-plots, so also is the hexagonal spacing of atomic groups in tetracyclic examples like pyrene and benzanthrene (diagrams 25 C and 25 D).

Many rather complex structures, such as cholesterol and morphine (diagrams 25 E and 26 A), or penicillin-G and yohimbine (diagrams 26 B and 26 C) can be dot-plotted in six lines of print with the A-L-Y-X letter expansions of the line-formula notation symbols that were introduced 16 yr ago.

Rectangular dot-plot diagrams (26 D and 26 E) make ideal data tables when the observed effects are recorded and tabulated as simple digits, and when most of the spaces can be left blank as inactive responses. Thus, in the given example (diagrams 27 A and 27 B), eight different effects from "A" to "H" are tabulated, from low (L) and high (H) dosage rates, on six different species identified here as the vertical row of letters, B to R, for item number 11788. The digits 2, 3, and 4 mean "slight," "moderate," and "strong" activity in any of the tabulating positions in the chart. The blank spaces obviously mean negligible activity, without using any symbol-storage capacity on the card. The overall low/high activity numbers also are dot-plotted under the item number; in this case the summarizing activity numbers are 7/14.

One of the most valuable applications of these single dot-plot cards was a numeric analysis of benzene-ring isomerism, made initially just as a memorial demonstration on Columbus (Explorer's) Day, 1965. Believe it or not, there are 828 different kinds of substituting patterns around this regular hexagon, too many to bother drawing by hand. All were gang-punched from a single dot-plot setting that provided for any symbol or none at the six substituting places, and this part had a lovely production rate of 100 per min. Then, after the variable marks were keypunched, in less than 30 min by a keypunch operator, this finished deck was reproduced twice (again at 100 per min) to yield three different sortings and listings. The computer generated the resulting 2484 diagrams in just 30 min. Subsequent leisurely inspection and analysis of these listings proved that one of the three methods of delineation was distinctly superior to the other two, so this led to a basic improvement of line-notations covering this important class of structures.

All of the illustrations shown thus far were made by photographing the actual computer-composed "dot-plot" diagrams, each from a single punched card carrying conventional punch-patterns in the 40 columns of character storage, and binary patterns (punched or not punched) for the 180 places in the 6 by 30 dot-plotting area. Here now (diagrams 27 C and 27 D) is a dot-plot diagram produced with two punched cards. The first 40 marks (or blank spaces to fill in any plotted "windows") must appear on the first card, and the next 40 marks must appear on the second card, in their respective character-storage columns, but now these 80 information marks can be shown anywhere in the full 12 by 30 grid. The top half of this grid is plotted on the first card, and the bottom half on the second card.

The number of possible dot-plot patterns for 80 or less punches ("dots" or windows) in the 360 positions, not even counting any differences because of the 63 available character marks, is enormous beyond human belief; there are vastly more dot-plot patterns in this grid than there are particles in the universe, so this number is quite beyond human comprehension, a magnitude of 10^{54} possibilities (diagram 27 E).

Perhaps the most impressive diagram in this entire report is the computer time taken to generate the dot-plot listing from which these illustrations were made (diagrams 28 B and 28 C), just 2 min. More precisely, just 100 sec.

Another potentially valuable application of the same equipment and cards is called the "double dot-plot" program because, in the simplest programming, the same long strip of diagrams is run through the line-printer a second time to print conventionally prepared name or number or descriptive

information alongside the diagrams. This "double dot-plot" utilizes nearly the full-width printing capacity of the line-printer, 120 columns in pica type. We hesitate to predict the many possible uses of this new tool for information scientists; however, we can predict the pending refinement on this particular "Double Dot-Plot" program that will simplify the input procedure and eliminate the necessity to run the paper through the line-printer a second time: control punches on the cards having the auxiliary data to be printed alongside the diagrams will permit these to be inserted in the card deck immediately after (or immediately before) the corresponding dot-plot card. Then any variable number of these "name and data" cards, or none at all, will be allocated by the program to the fields of available printing space (30 columns to the left and 60 to the right of the dot-plot diagram). The most attractive part of this merging of graphical information with nomenclatural or other delineated information is that all of the processing is done with widely available standard punched-card equipment: not a penny has to be spent buying any special attachments, and not a moment has to be spent in any preparations that are in effect special translations and departures from standard operating procedures.

CHEMICAL STRUCTURE DIAGRAMS

110

CHEMICAL STRUCTURE DIAGRAMS 5	CHEMICAL STRUCTURE DIAGRAMS 6	CHEMICAL STRUCTURE DIAGRAMS 7	CHEMICAL STRUCTURE DIAGRAMS 8
ALL GRU MARKS ARE STORED THIS WAY IN TOP OF CARD	E L S C S A S A E T A M N C T R N I D M T I F N E N T G R G	B C W L T I O O I M N U R K I A N K E S R Y S	RELATED NOTATION DOT-PLOT DIAGRAM NAME
FIRST PUNCH OUT THE PRINTING MARKS	E L S C S A S A E T A M N C T R N I D M T I F N E N T G R G	1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10	
THEN SET PLACES WITH DOTS TO PLOT	E L S C S A S A E T A M N C T R N I D M T I F N E N T G R G	1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10	Q1A VM Q1Y DO D-GLYCEROSE
EACH LETTER STRING CAN SHIFT AND SWING	E L S C S A S A E T A M N C T R N I D M T I F N E N T G R G	HERE ARE SOME SIMPLE CHEMICAL DIAGRAMS	Q1/VM Q1Y DO L-GLYCEROSE
E L S C S A S	BINARY BITS ARE STACKED LIKE THIS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	WHEREIN Q IS OH L IS CH2 Y IS -CH- I AND D IS -CH-	Q1B8A VM Q1Y Y DO D-RIBOSE

1

CHEMICAL STRUCTURE DIAGRAMS 13	CHEMICAL STRUCTURE DIAGRAMS 14	CHEMICAL STRUCTURE DIAGRAMS 15	CHEMICAL STRUCTURE DIAGRAMS 16
A A A A A A A A A B BA B ADA A B A C CO A A A B D 1 2 3 4 5	1V8X VX 22 C C C C 7	SAME DOT PLOTS NOW SHOWN WITH BRANCH SYMBOLS	1V8XY YXY 74 Y X Y • • Y X Y • • • • • 9
A A A A A A A A B B B OCO BA C C CO D A C DE B A B E A S MENAMES A	1A8BX XX 30 C C C C C C C C 8	1V81V 8VV 21 • V • Y • • • • 7	1V8V8A VYX 75 • V Y X • • • • • 9
ADDITIONAL EXAMPLES RELABELLED 5 WAYS	1V8VBV VVV 40 C C C C C C C C C 8	1V8X VX 22 • Y X • • • • 7	SAME DOT PLOTS NOW SHOWN WITH A-L-Y-X NOTATIONS
DELINEATIONS OR SERIES DOT- PLOT WO/C ITEM	1V8BY YXY 74 C C C C C C C C C 9	1A8BX XX 30 • X X • • • • 8	1312311 1V81V 21 A Y L V A A A A 7
1V81V YY 21 C C C C C C C 7	1V8V8X VYX 75 C C C C C C C C C 9	1V8V8Y VVV 40 • V Y V • • • • 8	1314111 1V8X 22 A Y X A A A A 7

A 13131311 1V8X 30 0 A A A A A X X A	1312311 1V81V 21 7 1 3 2 3 1 1 1 1	131314111 1V8V8X 75 0 1 3 3 4 1 1 1 1	13131311 1V8V8V 40 0 A Y Y Y A A A B A	A
B 13131311 1V8V8V 40 0 A Y Y Y A A A A A	1314111 1V8X 22 7 1 3 4 1 1 1 1	131411311 1V8V8V 74 0 A Y Y Y A A A B A	131411311 1V8V8V 74 0 A Y Y Y A A A B A	B
C 131411311 1V8V8V 74 0 A Y X Y A A A A A	1314111 1V8X 39 0 1 1 1 4 1 1 1 1	131314111 1V8V8X 75 0 A Y Y X A A A A A	131314111 1V8V8X 75 0 A Y Y X A A A A A	C
D 131314111 1V8V8X 75 0 A Y Y X A A A A A	13131311 1V8V8V 40 0 1 3 3 3 1 1 1 1	1314111 1V8X 22 7 B Y X A B A A	1314111 1V8X 22 7 B Y X A B A A	D
E 131411311 1V8V8V 74 0 A Y Y X A A A A A	131411311 1V8V8V 74 0 1 3 4 3 1 1 1 1	131411311 1V8X 39 0 A A A A A X X A	131411311 1V8X 39 0 A A A A A X X A	E

CHEMICAL STRUCTURE DIAGRAMS 21 CHEMICAL STRUCTURE DIAGRAMS 22 CHEMICAL STRUCTURE DIAGRAMS 23 CHEMICAL STRUCTURE DIAGRAMS 24

NOTATION	ATOMIC TRACING DIAGRAM	NAME	1X66 3X 2	QVR BOVI	ORSHV
1V6 3V	1 131V311311 31 1 3 V 3 1 1 1 1 1 1 1	NAME	1 1 1 141 141 1 14 X X 41 1 141 141 1 1 1 1 1	Q T O T A T T O O O O	O O D D T T D D M T D S O O
1V6 4X	1 131X311311 31 1 3 X 3 1 1 1 31 1 1 1	NAME	TWO DOT PLOT SYMBOLS HAVE TWO MNEMONIC AIDS	D O D O D D D O D	D O D T O D T O D D
1V6 5V	1 131V411411 41 1 4 V 4 1 1 1 1 1 1 1	NAME	D IS A DIATOMIC CM-GROUP	D T D D T D D O	L L L D T L D T L D C
1V6 6X	1 131X411411 41 1 4 X 4 1 1 1 1 1 1 1	NAME	T IS A T-BRANCHED C-ATOM	CY F J A O T D A T T D A O T A	D O D T D D T D D N
1V6 7X	1 131X411411 41 1 4 X 4 1 1 1 1 1 1 1	NAME	D-O IS ALSO AND T-O IS KETO	GVM O T D L T D O	D O N T O D T D T M O

CHEMICAL STRUCTURE DIAGRAMS 25			CHEMICAL STRUCTURE DIAGRAMS 26			CHEMICAL STRUCTURE DIAGRAMS 27			CHEMICAL STRUCTURE DIAGRAMS 28		
••••• M D T T O O T T O O O O O LARBAZOLE	L Y D T A K L Y O O T T K Y O O T T O O O MORPHINE	EFFECT-A B C D E F G H RATE - LM LM LM LM LM LM LM S E T	••••• O C D T T O O T T O O O O O PHENANTHRENE	S A O O Y K A O T L N Y N Y O O O T T O O O O O O PENICILLIN-G	11788 0 3 4 3 S 24 23 M 4 3 K 3 2 O 2 2 7/14 H	• TWO CARDS WITH NO MARKS STORED ON EACH CAN PUT MAXIMUM OF NO INFORMATION MARKS IN A 12 X 30 GRID	1000000 000000 000000 000000- 000000 000000 000000 000000- 000000 000 PLOTS	••••• L L L D L Y Y A Y A L O A P L L Y L L L L A A A CHOLESTEROL	ITEM H O S OF DATA ENTRIES ACTIVITY THIS CAN SHOW 34 10		
A			B								
C											
D											
E											

RUNNING TIME
FOR THIS DISPLAY
IS
JUST 2 MINUTES

THIS CAN SHOW

1000000 000000 000000 000000-
000000 000000 000000 000000-
000000 000 PLOTS

34
10

UPDATING PROGRAM FOR THE INDUSTRY LIAISON OFFICE

David E. Renard
Industry Liaison Office
Edgewood Arsenal

The Industry Liaison Office (ILO) of the Edgewood Arsenal Research Laboratories has successfully been using an index of permuted line notations for its file of chemical structures for several years.¹⁻⁴ The index has been thoroughly adequate for answering questions from the personnel of the Laboratories, and the use of the Wiswesser line notation has enabled the file to be kept complete and up to date.

This file of chemical structures numbers over 100,000 entries that have been donated from Industrial and Academic laboratories across the country in support of the military research programs at Edgewood Arsenal.

Because entries to the file are received from 500 to 600 sources, duplication is inevitable. This duplication can be beneficial to the Industry Liaison Program, for it gives multiple sources from which to receive further information or samples on interesting compounds. The place for this duplicate information, however, is in the ILO Log Book of Sources and not in the structure file.

Chemical structures donated to the file are divorced of Company identity and assigned an accession number. The structures are subsequently typed on 3 by 5 cards for use in circulation lists, reports, and searches. Much time and effort can thus be saved by checking for duplication prior to the assignment of accession numbers.

The computer program prepared at Data Processing Division enables the ILO to update its file with the truly new compounds while giving a record of those that are duplicates (see figure). Since the "CS" or "Commercial Source" number is needed only on the new structures, an automatic registry system has been incorporated.

As structures are received by the ILO, they are coded into line notations by a chemist and punched onto IBM cards with a temporary manufacturers' number (different from donor's identity number). At periodic intervals, these cards are taken to Data Processing Division where they are read onto tape and compared to the master file.

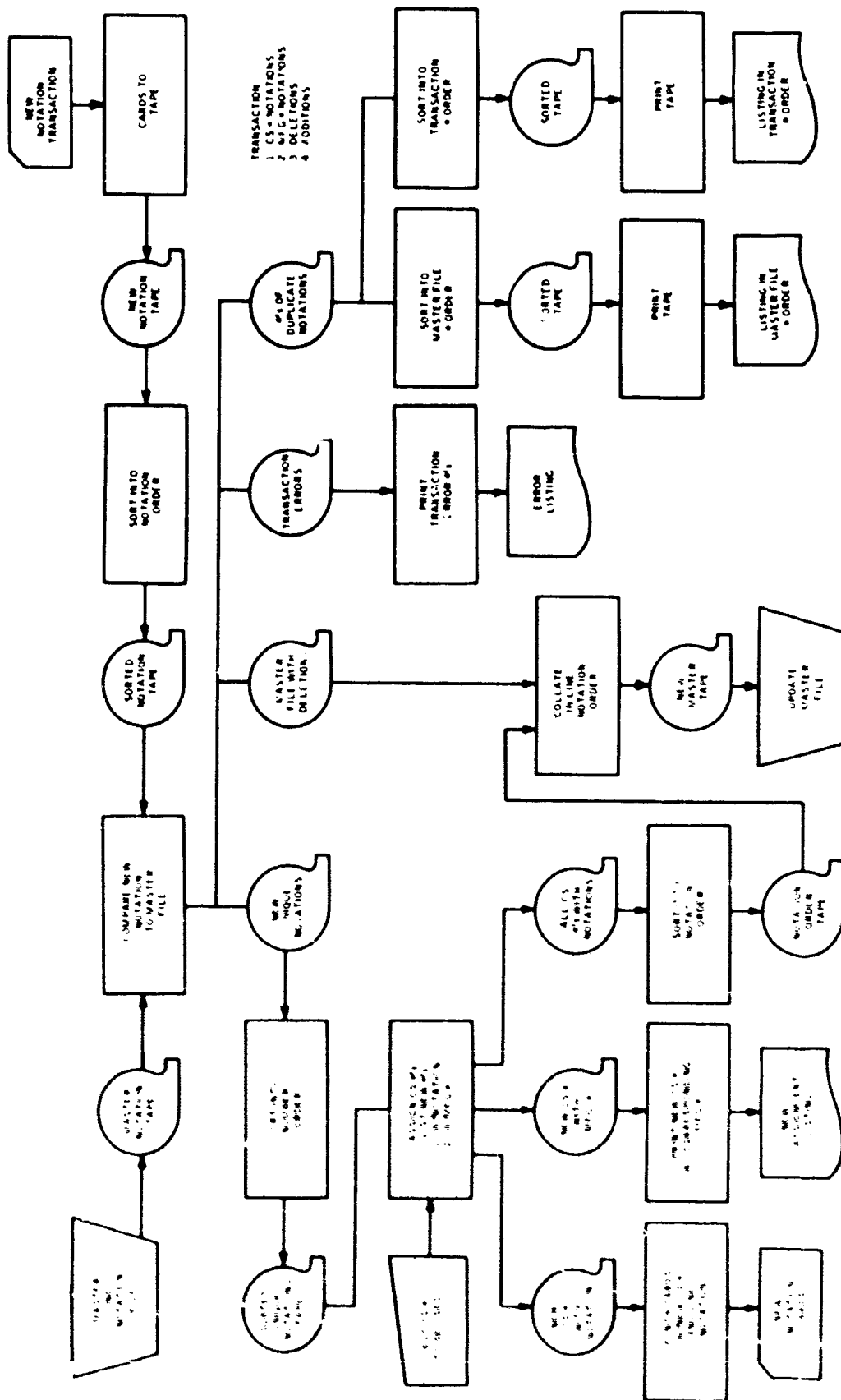


Figure. Updating Program for ILO Line Notation File

When the machine discovers duplicate line notations, the numbers are stored on another tape which in turn is used to generate two reports. One report has the duplicate number pairs listed in order of the notation from the master file while the other is ordered by the manufacturers' numbers of the cards. Together, these reports enable the ILO to update and cross-reference its log book of sources. A compound sought some time ago, but unavailable then, may now be available from a different company and will be requested.

Notations that are not found on the master file are stored on another tape. Some of these may have CS numbers, since the ILO retains the ability to assign CS numbers when necessary. Other notations with manufacturers' numbers are then assigned CS numbers beginning with a number determined by the ILO. From this step in the computer program, three tapes are generated. One will give a listing of the manufacturers' numbers and their newly assigned CS numbers so that this information can be entered in the log book. The second tape will be used to punch out new IBM cards containing the line notation with the new CS number. This will update the card file of line notations. The third tape contains all the new unique line notations which is next blended with the master file to bring it up to date.

The updated master file can now be used to produce a permuted listing of line notations which is the basis of the ILO's information retrieval system.

For a large and continually growing chemical structure file, the Wiswesser line notation provides a highly efficient method for updating the file without cluttering it with duplication.

LITERATURE CITED.

1. Gelberg, A., Nelson, W., Yee, G. S., and Metcalf, E. A. J. Chem. Doc. 2, 7 (1962).
2. Sorter, P. F., Granito, C. E., Gilmer, J. C., Gelberg, A., and Metcalf, E. A. Ibid. 4, 56 (1964).
3. Granito, C. E., Gelberg, A., Schultz, J. E., Gibson, G. W., and Metcalf, E. A. Ibid. 5, 52 (1965).
4. Granito, C. E., Schultz, J. E., Gibson, G. W., Gelberg, A., Williams, R. J., and Metcalf, E. A. Ibid. 5, 229 (1965).

UTILIZATION OF THE WISWESSER NOTATION IN CIDS

Clarence T. Van Meter
University of Pennsylvania

INTRODUCTION.

I feel somewhat out of place in the presence of this distinguished company highly experienced in the Wiswesserization of chemical structures, but, as a matter of fact, I am here to learn from the discussions rather than contribute to them. Rest assured that I have no intent to speak on the mechanics of the notation, i.e., the numerous rules and their application to encode the structures; these are matters for you experts and are of no immediate concern to CIDS. Rather, the CIDS concern is with utility of the notation as a source of information on structural features for retrieval purposes; and Mr. Mitchell and I felt that you might like to hear something about our visits with others who have expressed views on this aspect of the notation. We thought that you might like also to see an assortment of structural queries visualized as typical of those which would be addressed to an operational CIDS.

PROJECT CIDS AND ITS INTEREST IN THE NOTATION.

Prefatorily, I should explain that the unqualified acronym, CIDS, refers to the official Army program, Chemical Information and Data System, which was initiated a few years ago and is now in the latter stages of exploratory development. For the past 2-1/2 yr, a sizable portion of the program has been conducted under contract with the University of Pennsylvania and this contractual effort is known as Project CIDS. The responsibility for directing the work on this project has been entrusted to the speaker.

Now, it has been our understanding that, from the very inception of CIDS, the Army has consistently held the view that the Wiswesser notation might be a useful component of the system, but not necessarily as the sole structure probing technique. Until recently, most structure probing efforts have involved perfecting techniques for extracting this information from node/connector tables, and the time has now come to investigate accomplishment of this via the notation.

I must reiterate that it is not within the scope of our contract to probe-in-depth the intimacies of the notation, although I understand that such studies are in progress under other contractual efforts. Project CIDS is

concerned however, with all facets of the broad problem of satisfying the user's needs with acceptable economy and dispatch, and at some time in the future we expect to be asked for documented opinion with respect to the utility of the notation in handling some of these problems.

SOME VIEWS ON UTILITY OF THE NOTATION.

During the past 2 yr, we have therefore made it our business, while visiting various institutions, to solicit their views on the retrieval capabilities of the notation, and the wide diversion of these views is rather surprising. Without further elaboration, off-the-cuff estimates ranged from 30% "effective" or "good" to 100%, and one institution is said to have "mathematical proof" that the notation will not work (but I have not encountered anyone who has actually seen this "proof").

Naturally, such casual expressions of utility are of little value to CIDS. Disregarding economic features for the moment, what CIDS must know are: (1) the specific structural areas, if any, in which the notation is unreliable, and (2) the direction of the unreliability. With respect to (2) above, I might point out here that, for reasons to be disclosed later, CIDS can live with what I call "hyperretrieval" (I believe this is commonly termed "false drops"),* but it cannot tolerate "hyporetrieval" (misses).

With regard to the "mathematical proof," any such undocumented statement must be taken "cum grano salis." It is reminiscent of the mathematical proof that one can never reach a given destination because it requires an infinite number of half-journeys to get there while man's life span, at least as we know it, is finite. Many of man's "scientific proofs" turn out to be vulnerable; e.g., the bumble bee continues to fly in blithe ignorance of the fact that this is aerodynamically "verboten," and certain of the inert (group O) elements do form some stable compounds despite the "loner" characteristic emphasized in the Lewis octet depiction.

I strongly suspect that what the "mathematical proof" shows is that the notation will not work in 100% of the cases, and I hasten to add that such a finding would not be in the least surprising to me. But this does not spell zero utility to CIDS provided the structural areas in which it will not work are known.

* Indeed, false drops often provide serendipitous research leads that prove very useful.

THE "PERFECT STATE" OF CHEMICAL "SYSTEMS."

It seems important to remember that many of the chemist's so-called systems are highly useful even though they are not 100% perfect. We all recognize that this is abundantly true for systematic nomenclature and, perhaps more fundamentally, for the diverse systems used in drawing structural formulas. Furthermore, as long as man continues to exercise his scientific curiosity, systems such as these are bound to undergo refinement if we want to continually update them to incorporate the findings which result from probing the curiosity. Stated simply: A system which is 100% perfect today will not be 100% perfect tomorrow!

I have heard the Wiswesser notation criticized on the grounds that a given notation is not always unique. On reflection, I must say that this criticism seems rather strange to me — indeed, I would be somewhat alarmed if this were not the case — and this for the simple reason that: If man's known universe of chemical compounds were Wiswesserized and the resulting file restricted to completely unique entries, a very significant percentage of known compounds would not be in the file. This is exemplified in several of the typical queries I will leave with you.

Rather than a fault then, this feature becomes a virtue, whether it characterizes the Wiswesser notation, systematic nomenclature, or whatnot, provided that: the degree of nonuniqueness inherent to the system does not exceed that which is necessary to reflect truly man's state of knowledge at the time.

PROJECT CIDS VIEWS ON UTILITY OF THE NOTATION.

Naturally, we have found it necessary to record the results of the above visits in our formal contract reports, and I fear that here I have blundered rather badly. It appears that, although I tried to guard against it, I have somehow engendered the belief that we are anti-Wiswesser. Now I assure you that nothing could be further from the truth. The actual fact is that no one on our project is sufficiently cognizant of the notation to be either "for it or agin it." And it so happens that my personal activities, both at the University and in my consultant practice, require approaches through structural formulas and systematic nomenclature. Thus, our project has a completely open mind on the subject.

My personal prognosis, based on my very limited knowledge at present, is that there will be a useful niche for the notation in CIDS, even though it may be found not to be completely satisfactory in all structural areas.

In this regard, I should also mention that CIDS will have a built-in capability for the rapid display of actual structures. The busy researcher does not want a list of notations retrieved - in the parlance of the computer man, he wants the "pictures," for these constitute the common language of his worldwide scientific community.

EXPERIMENTAL EVALUATION OF THE NOTATION.

In essence, our experience suggests the need for an evaluation of the notation from a CIDS-utility viewpoint. It further suggests that the evaluation be of the direct experimentation type. Theoretical excursions into the problem may be feasible - we do not know - but it appears doubtful that the results would be nearly as convincing as those resulting from direct experimentation.

We visualize a four-phase experiment:

Phase I - Construction of a suitable file of test queries.

Phase II - Construction of a suitable file of test compounds.

Phase III - Wiswesserization of the compounds of (II) and running the queries against the notated compounds.

Phase IV - Assessment of the results of III from the standpoint of (a) chemical accuracy, (b) time, and (c) cost.

OBSERVATIONS ON INDIVIDUAL PHASES OF THE EVALUATION.

Phase I.

The file of test queries should be comprehensive and fully representative of:

1. Structural characteristics.
2. The Army's structural retrieval needs.

It is preferable that this file be constructed by persons other than those to be engaged in Phase III (Wiswesserization of compounds). As a further guarantee against inadvertent bias, it is preferable that the persons engaged in construction of the query file be unknowledgeable Wiswesserwise.

Phase II.

The file of test compounds should:

1. Be statistically compatible with the file of test queries.
2. Contain a generous percentage of "dummy compounds," i.e., compounds which should not be retrieved from any of the queries.
3. Provide for adequate representation of tautomeric forms.
4. Provide, through multiple entries of the same compound, inquiry into effectiveness of retrieval regardless of how the structures are presented by the querist.

If constructed expressly for the evaluation, a suitable file may contain as few as 10,000 compounds and in all probability not more than 25,000 compounds.

It is possible that a suitable file may result from combining selected portions of existing files, but it seems likely that the effort required to do this would be greater than that required to construct the brand new file mentioned previously.

As in Phase I, persons other than those engaged in Phase III should construct the file of test compounds.

Phase III.

This phase of the evaluation is the specific task of the Wiswesser specialist. Using the most recent set of rules, each compound in the test file is to be baptized with its correct linear notation. Only highly experienced encoders, as agreed upon by the sponsor and Mr. Wiswesser, should be entrusted with this responsibility.

Phase IV.

Overall assessment of the results involves (1) retrieving the information demanded by the queries by the two techniques, one utilizing the notation and the other not, and (2) comparing the findings by the two techniques. The nonnotation technique could be adapted from either of the following:

Method A.

Automated retrieval utilizing molecular formulas, structural fragments, and atom-by-atom probing.

Method B.

Manual inventory of the file in terms of the test queries.

Method A is open to the objection that it is not conclusive. Method B, on the other hand, requires a much greater technical effort. Whichever method is elected, however, the ultimate procedure is the same, viz., compare retrievals via the notation technique with the retrievals via the nonnotation technique from the standpoints of:

1. Chemical accuracy and completeness.
2. Time rate of query processing.
3. Cost.

A SPECTRUM OF TYPICAL CIDS QUERIES.

By way of conclusion, I have selected from the project files a few structural queries, each typical of one which might be addressed to an operational CIDS. Collectively, we, along with others, have assembled some 400 such queries, and I have tried here to select a "handful" and order them roughly in increasing complexity.

I shall refrain from taking up your valuable time by discussing each of the 17 queries on the list. Copies have been distributed which, if you wish, may be examined later at your leisure. I might add that we will be most happy to receive any comments you may wish to make, indeed, we solicit them. Send them either to Mr. Mitchell or to me; each of us will see that the other gets copies. I would like, however, to make a few comments now on some of the queries.

You understand, of course, that querists will usually want some additional information/data as well as retrieving the actual compounds, but this aspect of querying does not involve the structurally descriptive characteristics of the notation, and is therefore not included in the presentation.

It is fully expected that many of these will present no problems whatsoever with respect to retrieval via the notation. It is hoped that none of them will.

How satisfactory the economics of search will turn out to be is, of course, unknown. Perhaps I should mention here that it is our expectation that a practical CIDS will take full advantage of molecular formulas, in whole or in part, in order to limit the number of compounds which must be searched further.

I should mention also that we visualize a technical information specialist functioning in CIDS as a intermediary between the querist and the system operator. The busy man at the bench will then be allowed to ask his query in any way he wants, and it will be the job of the specialist to translate the querist's language into the language of the system. We have consistently held the view that to restrict the querist in any way will be to reduce his appetite for the system.

A few of the queries are worthy of brief specific comment. Query No. 4 illustrates nicely one of the kinds of differences between the theoretical and practical demands of a system. An actual search of the literature reveals that only about 50 C₂₀-alkanols are known and thus to become deeply involved in discriminating among the 5.6×10^6 theoretical isomers is to depart from reality.

Query No. 12 actually arose some time ago. A substance on hand was thought to be a mixture of two or more of the eight stereoisomeric forms of the structure shown, and the researcher wanted to know which, if any, of these stereoforms had been synthesized and studied.

Query No. 14 actually represents 13 different queries, each with a little "twist" of its own, which collectively read on perhaps the most troublesome area of structural search, i.e.; queries which, regardless of any functional groups that may or may not be present, require the ability to retrieve solely on characteristics of the heteronuclei. During the last quarter century, we have witnessed a remarkable increase in the chemist's ability to synthesize these structures with the result that new types of compounds are providing vistas for application to numerous chemical technologies. Thus, queries based on the nuclei are becoming increasingly frequent.

Query No. 15 is designed to disclose how the system responds to queries involving (a) tautomers, and (b) identical structures drawn in diverse orientation.

Query No. 16 merely illustrates the large number of actual compounds which may be encompassed by a seemingly simple query.

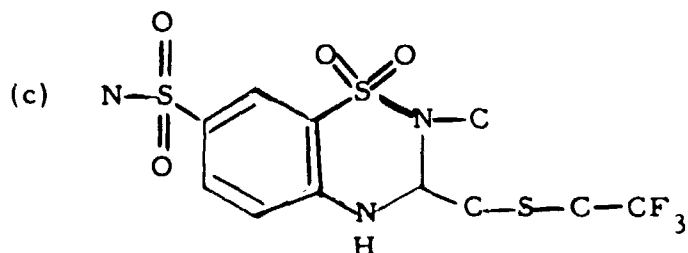
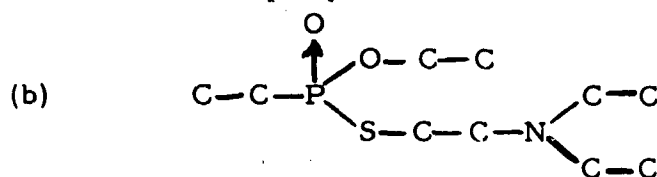
Query No. 17 is another example of a situation which has actually been encountered. A compound of the type shown was inadvertently discovered to have an interesting property and the search was among the family of related compounds for a member in which the desired property might be more pronounced. A brief examination of the problem soon reveals its enormity by the ordinary methods of search, but it is our expectation that a practical CIDS could prosecute the search in a matter of minutes.

TYPICAL CIDS STRUCTURAL QUERIES.

Retrieve:

1. The following specific compounds:

- (a) Perfluorocyclopentane



2. All acyclic compounds which contain any kind of a guanidine

residue and a formyl group.

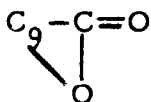
3. All of the 17 isomeric forms of the C_6 -alkanols on file and isolate those which are secondary alcohols along with their corresponding ketones

$C_6H_{13}OH$	$-CH(OH)-$	$-CO-$
theory: (17)	(6)	(6)

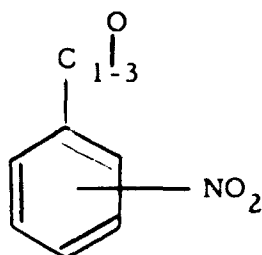
4. All C_{20} -alkanols on file and isolate those which, by IUPAC nomenclature, are alkylated hexadecanols.

	$C_{20}H_{41}OH$	$C-(C_{14})-C$ (parent)
theory:	5.6×10^6	?

5. All $C_{10}H_{16}O_2$ compounds on file that (a) are lactones and (b) contain C_3 as the only cyclic nucleus in the molecule.

	(a)	(b)	
	$C_{10}H_{16}O_2$		
theory:	$n \times 10^3 (?)$	$< 10^2 (?)$	$< 10^2 (?)$
total file:	few hundred (?)	(?)	(?)
CA vol'62	78	7	4

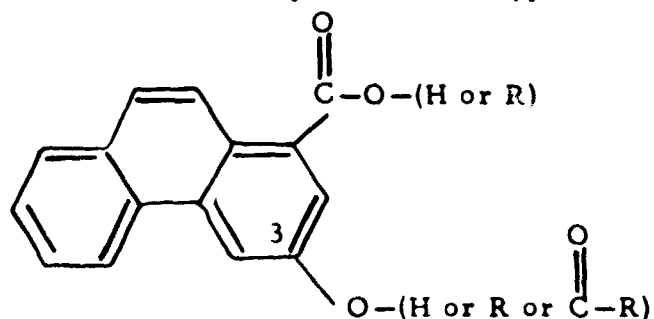
6. All monohydric mononitrophenylalkanols containing from 7 to 9 C atoms.



Mol. form. $C_nH_{2n-7}NO_3$ where $n = 7, 8, 9$

theory:	N =	7	8	9
No. isomers =		3	6	18 = 27

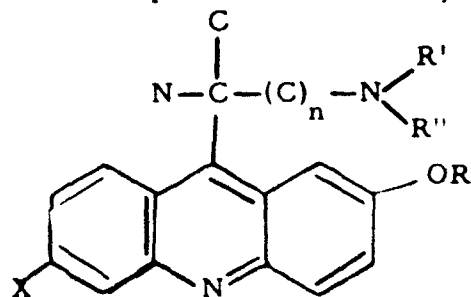
7. All compounds of the type:



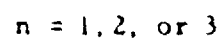
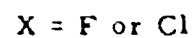
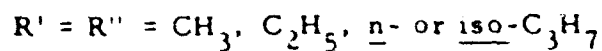
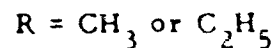
in which R is any hydrocarbon radical

- (1) with or without any other substituents on the nucleus
- (2) (a) without any other substituents on the nucleus
- (3) (b) restricted to 3-OH compounds.

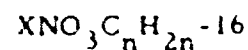
8. Compounds of the family:



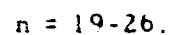
wherein:



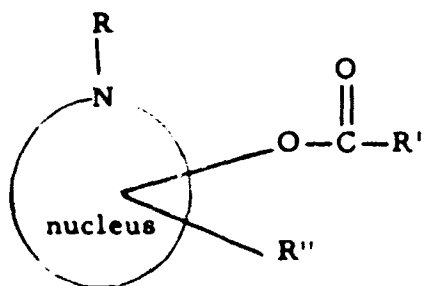
Mol. forms:



wherein:



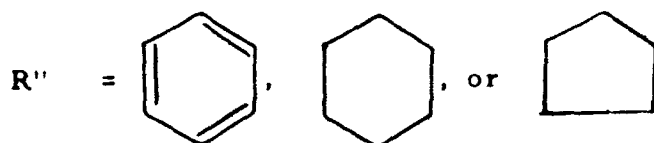
9.



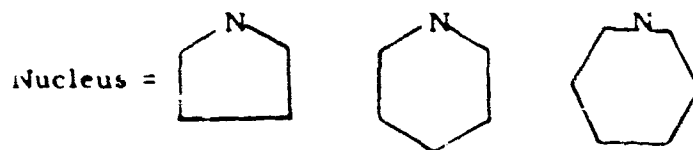
Restrictions:

R = CH₃ or C₂H₅

R' = any carboxylic acid residue



-O-C(=O)-R' and R'' attached to same C atom of nucleus



10. L-Xylose

D-Mannose

Furmaronitrile and Maleonitrile

d-Epinephrine

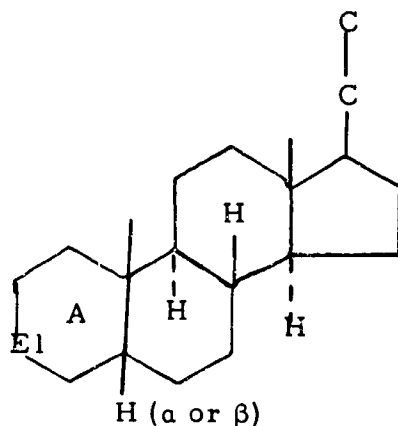
meso-Tartaric acid

anti-2-Nitro-2-furaldehyde oxime

9-Fluoro-11 β , 17-dihydroxy-6 α -methylpregna-1, 4-diene-3, 20-

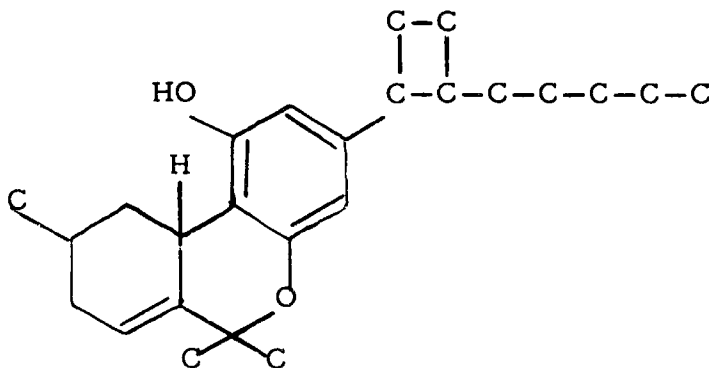
dione

11. All 5α - and 5β -pregnane derivatives containing one heteroatom in ring A.



position of El variable within A
no generalized mol. form.

12. All stereoisomeric forms of



13. All glycine or glycine derivatives which contain any of the following isotopes: (a) deuterium, (b) C^{14} , (c) N^{15} .

14. Queries Involving Heteronuclei.

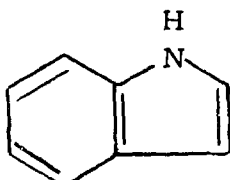
- Identify all known four-ring nuclei which contain 2 S and 1 O atoms and no other heteroatoms.
- Identify all known nuclei of the 6,6 system which contain a total of 8 C and 2 N atoms.

c. Which of the nuclei identified in b., above, are of the types:

(1) $C_4N_2-C_6$; (2) $C_4N_2-C_5N$?

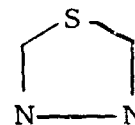
d. Identify all three-ring nuclei represented in the file which consist of either quinoline or isoquinoline in ortho or ortho-peri fusion with benzene.

(Note: Preliminary analysis discloses that all wanted nuclei are of the $C_5N-C_6-C_6$ type and of the two skeleton molecular formulas, $C_{13}N$ and $C_{12}N$.)

e. Which isomers of  are known which differ only in the location of the indicated hydrogen?

f. What naphthofurans, if any, are known which contain five double bonds?

g. What compounds, if any, are on file which contain the nucleus, the nucleus having (1) two double bonds and (2) one double bond?

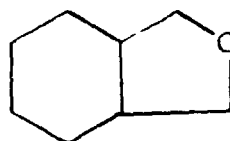
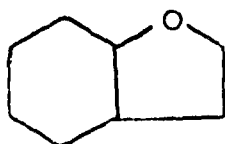


h. What four-ring nuclei are known which have the skeleton molecular formula $C_{17}N$?

i. Which of those in (h) belong to the $C_5N-C_5N-C_6-C_6$ group?

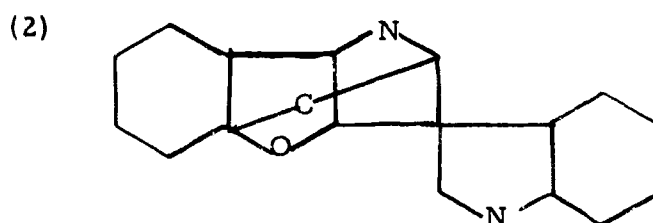
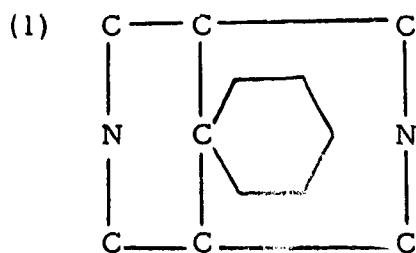
j. Which of those in (i) are ionic forms?

k. What compounds are on file which contain any form of the C_4O-C_6 nucleus other than the following (any stage of hydrogenation):

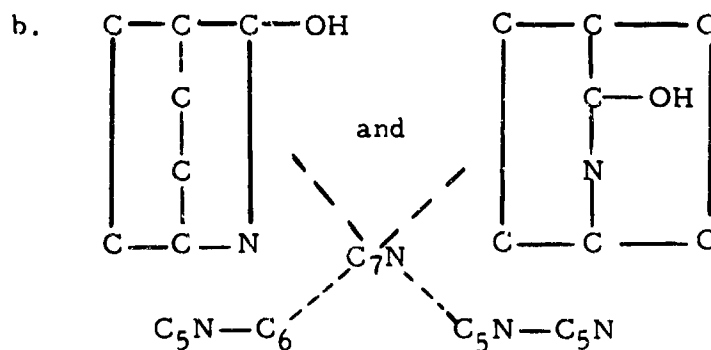
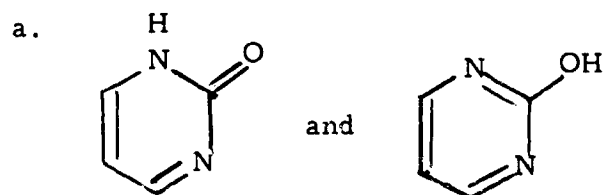


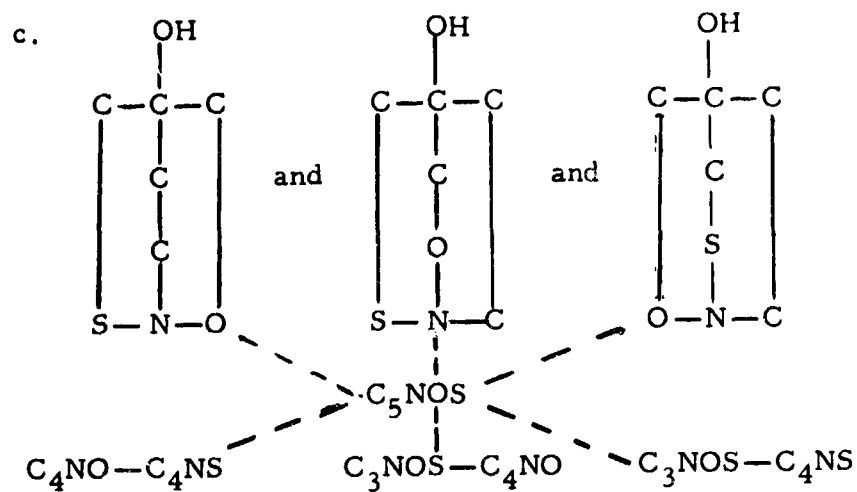
l. Which of the oxadiazoles is (are) represented by polymers in the file?

m. Retrieve all compounds on file which contain the following in any stage of hydrogenation:

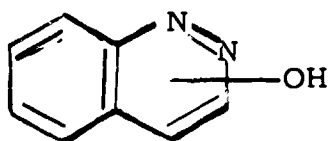


15. Give registry numbers of the following:





16. Retrieve all registered compounds of the type:



Restrictions:

6,6 systems only

LSHF's only

1 OH only

OH to C only

N's anywhere in nucleus

Heteroparents

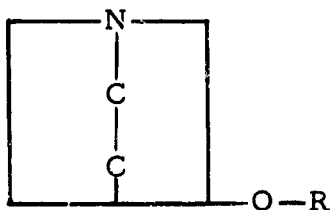
Type	Number
$C_4N_2-C_4N_2$	1
$C_4N_2-C_5N$	20*
$C_4N_2-C_6$	4
C_5N-C_5N	<u>6</u>
	31

Monohydroxy Derivatives

Mol. Form.	Number
$C_8H_8N_2O$	2
$C_8H_8N_2O$	140
$C_8H_6N_2O$	18
$C_8H_6N_2O$	<u>24</u>
	184

* Large number due to fact that each LSHF contains an "extra" ("indicated") hydrogen.

17. Azabicycloalkanol/esters problem:

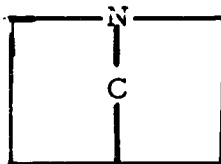


Preliminary study discloses restrictions:

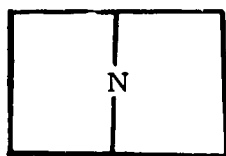
- a. Heteroparents saturated only
 only 1 N atom
 N atom common to both rings
 no spiro forms permitted
 total ring₁ plus total ring₂ atoms = 6-16
 = C_nN-C_nN through $C_nN-C_{14-n}N$ where $n = 2-7$

Three general types (examples):

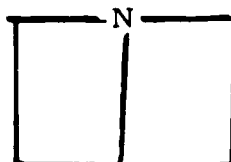
N-Bridgehead



C-Bridgehead



Fused (zero bridge)



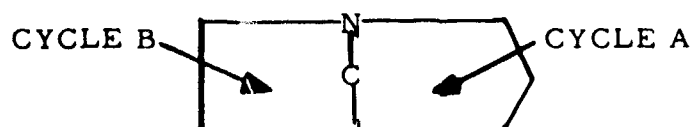
- b. $R = H$ or acyl of any monocarboxylic acid.

Table. Numbers of Theoretical Parents in the Retroarea of Interest

Cycle A <u>b/</u>	Numbers of atoms <u>a/</u>					
	Cycle B <u>b/</u>					
	3	4	5	6	7	8
3	1					
4	1	3				
5	1	3	3			
6	1	3	3	5		
7	1	3	3	5	5	
8	1	3	3	5	5	8
9	1	3	3	5	5	
10	1	3	3	5		
11	1	3	3			
12	1	3				
13	1					

a/ Number of atoms in each cycle of a bicycle includes atoms shared with other cycle.

b/ Cycle designations:



CONNECTION TABLES FROM WISWESSER LINE NOTATION: A PARTIAL ALGORITHM

George F. Fraction
Eli Lilly and Company

and

Justin C. Walker
Stephen J. Tauber
National Bureau of Standards

An algorithm has been developed for transforming certain types of Wiswesser organic structure notations into connection tables. Acyclic and benzene structures are treated, and provision has been made for all of the types of contractions used by the Wiswesser notation system. A separate algorithm is presented for treating linearly fused ring aggregates. A syntax has been developed to describe those portions of Wiswesser notations that refer to nonbenzene ring systems.

INTRODUCTION.

The purpose of our work with the Wiswesser notation was to make generally available in algorithm for generating connection tables from such structure notations. This work has had to be terminated before reaching its final objective and will not be resumed in the foreseeable future. We, therefore, wish to make our results available, although they are admittedly incomplete.

The algorithm here described will provide connection tables from the acyclic and benzene portions of a notation and process all forms of contraction demanded by the rules. The known bugs in the algorithm are indicated, with suggestions for their elimination. Finally, our approach to handling nonbenzene cyclic notations is described. This comprises a syntactic description of the portions of a notation enclosed in the L...J or T...J symbols. In addition, a separate program has been written to construct connection tables for linearly fused ring aggregates from the syntactically analyzed notations.

Hyde et al. have recently developed a program for a similar analysis of Wiswesser notations.¹ The two approaches differ in several respects: Hyde et al. use notational symbols as the basic structural units (e.g., Q, V, 3), whereas, we use the individual atoms, bar hydrogen attached to carbon. They in turn indicate the number of hydrogen atoms attached to

ring carbon atoms. The program they describe does not interpret substituent locants of benzene rings in terms of the individual ring atoms. They do not describe a method to handle the systematic contractions which the rules require. Another program for transforming Wiswesser notations also exists² but it is not publicly available.

The programs herein described, with the exception of the one written in ALGOL-60, were implemented on the NBS PILOT computer facility, which has been decommissioned since this work. The language used was the assembly language, PEAP, providing for three-address instructions with operation code modifiers, sequence control register digits, and breakpoint control digit. The configuration of the system was 64K of 68-bit word core memory, magnetic tape, magnetic wire, punched tape, typewriter, and Data-phone input and output and punched card and raster scanner input.

THE NOTATION.

The notation rules underwent change during the time that this laboratory was working with them and, therefore, specific note had to be taken of the modifications.

The versions of the wiswesser notation manual of October 1964 and of April 1965 were compared and, in addition, members of the Wiswesser notation committee were contacted directly on specific points.

The following specific changes were noted in the manual:^{3, 4}

1. The rank of the slash (/) was shifted from between R and S to between hyphen (-) and (Ø). Cf. Rule 2.
2. The concept of "unshared bridge atoms" and a rule for their citation was added. Cf. Rule 31C.
3. A section on ring analysis was added. This does not affect the notation rules but does give suggestions useful in error checking.
4. The section on chelate rings was eliminated.
5. A section was added which comments at length on redrawing structural formulae for easier encoding, with particular attention to uncrossing valence bonds.
6. Rules were added for enciphering ions, free radicals, and isotopic species. The specification of the location of charge, unpaired electron, or isotope relies on citing the number of the position within the symbol

string of the symbol which refers to the atom in question. Cf. Rules 50-52.

7. Rules were added for citing several types of indeterminacies. Substituents can be cited as attached at an unknown point, to a cyclic atom or to an acyclic atom not otherwise specified, to a specific ring system, to a ring system between limits of two locants, to a specific side chain, or to a specific alkyl or alkylene chain. Symbols are defined for unspecified hydrocarbon moieties, for alkyl groups of specified size but unspecified structure, for generic halogen, and for generic metal. Monocoordinated Markush groups can be cited. Cf. Rules 53-63.

Correspondence with members of the Wiswesser notation committee disclosed no further extensions to the Wiswesser notation in the areas of Markush structures or partially indeterminate structures nor any rules to cover stereoisomerism, coordination complexes, or inorganic structures. There have, however, been proposals, not yet formalized in rules, which provide for:

1. The use of "D" in initiating the notation for a ring system that contains a donor-acceptor bond.*
2. The use of "0" to signify a metallocene.**
3. The use of "C" for cis and "T" for trans.†

Rules have also recently been adopted to resolve alternatives among possible multiplier contractions.‡ There evidently is still no provision to distinguish among cyclic double and triple bonds in certain circumstances. Thus, according to Rule 31,⁴ both cycloocta-1,3,5,7-tetraene (structure I) and cycloocta-1,3,5-triene-7-yne (structure II) would be enciphered as L8J.

* Landee, F. A., Bowman, C., and Reslock, M. H. Private communication. November 1965.

** Granito, C. Private communication.

† Smith, E. G. Private communication. June 1964.

‡ Smith, E. G. Private communication. October 1965.

TRANSFORMATION ALGORITHM.

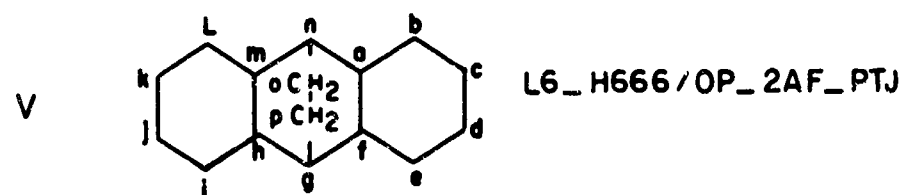
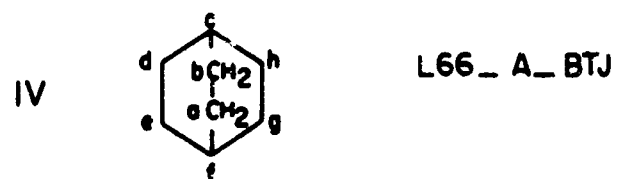
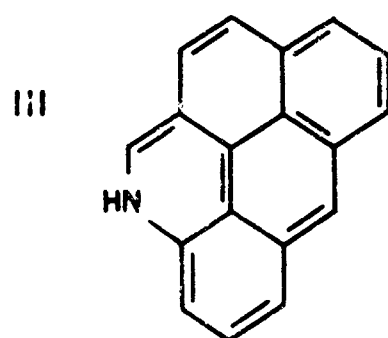
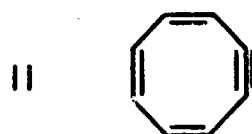
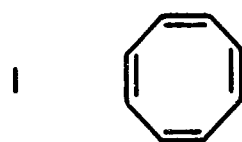
The algorithm under development for transforming Wiswesser notations (WISCO) was being so designed as to yield connection tables compatible with those resulting from the computer routine developed for transforming Hayward notations.⁵ The atoms would generally be listed in a different sequence depending on whether a Wiswesser or Hayward notation had provided the input, but the format of the connection tables resulting from the two transformations would be identical. An algorithm was essentially completed, programmed for, and partially debugged on the NBS PILOT computer facility to process benzene rings, acyclic portions, and multiplier contractions. The algorithm is shown at the stage it reached in figures 1 through 19.* The operation of this algorithm is highlighted in sections 2 and 3, below.

No algorithm usable for processing the general cyclic portions of Wiswesser notations has been devised; however, a linguistic structure of such cyclic portions was worked out as a basis for a transformation algorithm. Furthermore, an algorithm for analyzing a limited class of nonbenzene cyclic notation portions was devised. The syntax for general cyclic portions and this limited algorithm are presented in section 4, below.

No attempt was made in this phase of the work to deal with notations for salts, ions, free radicals, isotopic species, indeterminate locations, or generic substructures. For the present work a maximum coordination number of four has been assumed. Furthermore, it is suspected, although the algorithm was not developed far enough to verify this suspicion, that the processing of complex combinations of branched locants in ring system notations would require an effort disproportionate to the benefit to be derived from being able to deal with such rare cases. It appears to us sufficient to be able to construct connection tables from notations containing simple branched locants, and merely to shunt out those which contain either two branched locants from the same locant or a branched locant from a branched locant; i.e., those covered by Rule 39.⁴ We also believe that it is sufficient to be able to process notations for a single ring of rings (Cf. Rule 44);⁴ it is in any case not clear how Rule 44 is to be applied to structures with two or more rings of rings.

WISCO incorporates error checks both on the input notation and on the program itself. The errors are detected by looking for invalid situations and printing out an appropriate error message. For instance, if the character currently being examined is an "H" (hydrogen) or a "/" (slash

* Figures 1 through 29 are in the appendix.



mark) and the atom counter J has value 1, this indicates that the character initiates the notation and the cipher is rejected (except when the notation is HH). Another instance is a symbol for a multiple bond in an illegal context; e.g., UZ (i.e., $\equiv \text{NH}_2$, which, though chemically conceivable, would receive a different coding). When a symbol which stands for an atom set that demands a single bond connection to the rest of the compound occurs, the bond indicator (set to 2 or 3 when U or UU is encountered) is checked, and if the indicator is nonzero the notation is rejected.

Several ancillary routines were developed for use with WISCO. These were for convenience in implementation; they are not direct consequences of the logic evolved for transforming Wiswesser notations. Subroutine DEBUG (cf. figures 21 through 25) allows examination of system storage when a notation has been fully processed or when an error exit occurs. The programmer may then either look at storage or modify storage and re-enter WISCO at the beginning. The entry to WISCO is via subroutine INPUT, which reads a cipher one character at a time, checking each character for acceptability and storing it in a separate word of memory. (The 6-channel Flexowriter codes are converted upon input to 8-bit ASCII.) Two spaces in succession signify the end of the cipher. After a cipher has been read in, control passes to WISCO proper. Storage tables are set up not at assembly time but rather via subroutine GIMME (cf. figures 26 through 28). Storage is allocated pseudo-dynamically; i.e., storage is provided at run time, but if an overflow occurs then the storage is automatically augmented and the cipher is reprocessed afresh. GIMME is called with an argument SIZE. A value of zero for SIZE signals initialization for a new cipher. If SIZE has a "small" value, GIMME creates table storage of that size up to memory capacity and tells WISCO the table limits via SIZE. A "large" value of SIZE represents a table address and tells GIMME that a certain block of storage is being returned and causes this table to be erased from GIMME's records. "Large" and "small" values are defined by a constant in GIMME. GIMME will provide table storage of up to several thousand 68-bit words.

In the course of processing a notation there are several syntactic structures which require WISCO to employ recursive techniques. The recursion is facilitated by the use of pushdown stacks; i.e., last-in first-out storage, which are used for processing multiplicative contractions and branch symbols, as described in section 1, below.

1. Multiplicative Contractions and Branch Symbols.

Multiplicative contractions and chains of benzene rings necessitate recursive processing because multiplied strings may be nested (but not overlapped) and because the routine must be able to find the proper ring to which to attach a given substituent. When the algorithm is extended to handle general

cyclic notations all ring systems, including benzene rings, must be processed with the same push-down stacks. Similarly, all branch symbols must be processed recursively. As each layer of a nested sequence of multiplied strings is encountered, the relevant information is pushed onto a stack. The expansion of the notation (i.e., duplication of portions of the connection table) then begins with the innermost layer of the nest, and each recursion expands everything from a given layer in. For example, in notation VI, first the portion "/NCU1/_4" is expanded; then, at the same level, "/SWONUM_3"; next, the entire portion from the first "/" to the second "/"; finally between the triple slashes.

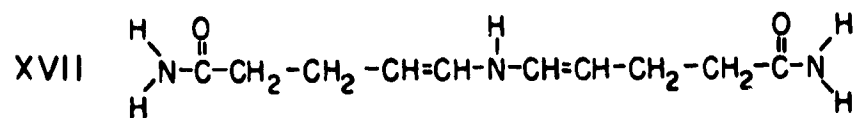
WISCO recognizes and processes the following types of multiplicative contraction, an example of each of which is indicated:

- a. Multiplication of initial strings (notation VII).
- b. Multiplication of terminal strings and side strings (notations VIII and IX).
- c. Multiplication on a central asymmetric unit (notations X and XI).
- d. Polymeric multiplication (notation XII).
- e. Multiplication of strings cited prior to the citation of the ring to which they are attached (notation XIII).
- f. Multiplication of strings cited after the citation of the ring to which they are attached (notation XIV).
- g. Multiplication of strings attached to rings but with implicit locants (notations XV and XVI).

Push-down stacks are also used for processing:

- h. Rings--the correspondence between atom number and locant on each ring is stored together with the number of possible substituents at each locant.
- i. Branched atom symbols--the atom number of that atom is stored once for each branch attached.
- j. Variable valence branching symbols--the atom number is stored for each variable valence atom; a flag is stored for each ring. Each atom with variable valence is pushed onto the branched atom stack four times,

- VI ZVXQ&M1Y&VOYUS&YGUIX///1UNY//O/NCU1/_4
X/SWONUM_3//_2 ///_3
- VII ZYUS&O_-2N101U1M_-2Y3UYZSONNUM&X
- VIII ZYSHYVQ2MIUIY/ SO1XZ&NW_2
- IX Z5X/ MNUNNW_-26YQ
- X QVYQ1U1_-2/PQO/
- XI Z1U1O_-3/YSWM/
- XII ZSW/2VM/_-4SWQ
- XIII WN_-4_-B_-C_-DR_-EZ
- XIV ZOVR_-B_-C_-E_-F-/ VOQ_-4
- XV WN_-5- R_-FZ
- XVI WNR- / Z_-5



which exceeds the number of branches permitted by the present algorithm. Whenever the routine has finished with the top entry in the variable valence stack, it erases the record of the corresponding atom from the top of the branched atom stack or of the corresponding ring from the top of the ring stack.

Not all of the above features were successfully implemented. The subroutine which processed the locants used in multiplication type d, above, did not give the proper results; it was removed and has not yet been rewritten. A single atom number for the last atom in a string of type a, above, or the first atom in a string of type b, above, is stored for reference in determining the limits for multiplication. This provision should be modified also to allow for recursion because, e.g., in a nested sequence of type a multiplication only the atom number needed for the last-encountered multiplier is now retained.

2. Acyclic Notations.

With these conventions in mind, we turn now to the processing of a cipher. Consider structure XVII, which is coded as ZV3U1_2M. The fact that the first character is a Z calls a routine which stores the NH_2 group as follows:*

1	H	(0, 2)	
2	N	(0, 1)	(0, 3)
3	H	(0, 2)	

where for each (i, j) couple i represents the bond type for the connection between the current atom and atom j. A single bond is indicated by $i = 0$; double, triple, and nonlocalized multiple bonds are represented by $i = 2, 3$, and 1, respectively. If the NH_2 group had not begun the notation, it would be stored in the order N, H, H, thus terminating a path. But in this case the path must continue from Z with the next atom, and a back connection of 2 (the atom number of the N) is stored. In a similar way, the symbol V is translated as $\text{C}=\text{O}$ and the connection table becomes:

1	H	(0, 2)		
2	N	(0, 1)	(0, 3)	(0, 4)
3	H	(0, 2)		
4	C	(0, 2)	(2, 5)	
5	O	(2, 4)		

* Actually atomic numbers are stored, not atomic symbols. The numbers of the atoms on the left are for ease of reading only; they do not actually appear in the machine representation.

The symbol 3 calls a subroutine which stores in sequential order as many carbons as indicated by an alkyl numeral. The symbol U causes a bond indicator to be set equal to 2 and processing continues. As each atom is encountered in the symbol string, the back connection is stored in its row of the connection table and its number is stored in the row of the atom so connected to it. Thus the bond indicator is set before the bond to which it refers has been stored. When atom 9 has been reached, the connection table appears thus:

1	H	(0, 2)		
2	N	(0, 1)	(0, 3)	(0, 4)
3	H	(0, 2)		
4	C	(0, 2)	(2, 5)	(0, 6)
5	O	(2, 4)		
6	C	(0, 4)	(0, 7)	
7	C	(0, 6)	(0, 8)	
8	C	(0, 7)	(2, 9)	
9	C	(2, 8)		

Next, the routine encounters " 2", a space followed by a number, which must be a multiplier. Several possibilities are consequently checked for immediately following:

- (1) A space followed by a letter, i.e., a locant.
- (2) "_R", i.e., implicit locants.
- (3) "/", i.e., possible central asymmetric unit.
- (4) Two spaces, i.e., the end of the cipher.

Any one of these situations would cause specific action. The present example, however, requires further analysis, since the local context of the multiplier admits either a multiplicative contraction of type a, above, or the multiplier side chains of type b, above. The routine as written pushes the multiplier (2) onto the top of DECML and the most recent atom number (9) onto JKEEP.

A modification of the routine would permit resolution of this situation, since, if type b multiplicative contraction were the case, a slash would have been noted on the slash stack. A slash between the beginning of the notation and the multiplier must be paired with the latter, for the slash delimits the range of the multiplier. If the slash stack is empty the multiplier covers everything back to the beginning of the cipher.

The next character examined is "M," and the connection table looks as follows after it has been processed:

9	C	(2, 8)	(0, 10)
10	N	(0, 9)	(0, 11)
11	H	(0, 10)	

Now we have reached the end of the cipher. At this point a check of the various stacks is made to see whether there is any outstanding business, in particular whether any multiplication needs to be effected or whether any methyl contractions need to be expanded. If the branched atom stack is not empty, the atom indicated has an implied methyl attached. Once the branched atom stack is empty (as it is immediately in this case), JKEEP is tested. If JKEEP = 0, ring locant storage is checked for ring multiplication. In our example, JKEEP $\neq$ 0, which means that multiplication of type a has been defined. (WISCO as presently implemented will not detect type b multiplication since it assumes at this point that type a is specified. This defect could be eliminated as indicated previously.) A subroutine then copies the portion of the connection table indicated, in our case from atom 1 to atom 9, indicated by JKEEP. A switch indicates that the sequence is to be copied in the reverse direction; i.e., from the end to the beginning. The values of j in the (i, j) couples must be suitably modified during copying. (This routine will work satisfactorily with a slight modification for "forward" copying, but a bug as yet uncorrected prevents proper "reverse" copying.) The end result of the algorithm will be:

1	H	(0, 2)		
2	N	(0, 1)	(0, 3)	(0, 4)
3	H	(0, 2)		
4	C	(0, 2)	(2, 5)	(0, 6)
5	O	(2, 4)		
6	C	(0, 4)	(0, 7)	
7	C	(0, 6)	(0, 8)	
8	C	(0, 7)	(2, 9)	
9	C	(2, 8)	(0, 10)	
10	N	(0, 9)	(0, 11)	(0, 12)
11	H	(0, 10)		
12	C	(0, 10)	(2, 13)	
13	C	(2, 12)	(0, 14)	
14	C	(0, 13)	(0, 15)	
15	C	(0, 14)	(0, 16)	
16	C	(0, 15)	(2, 17)	(0, 19)
17	O	(2, 16)		

18	H	(0, 19)		
19	N	(0, 16)	(0, 18)	(0, 20)
20	H	(0, 18)		

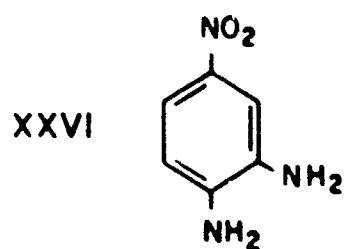
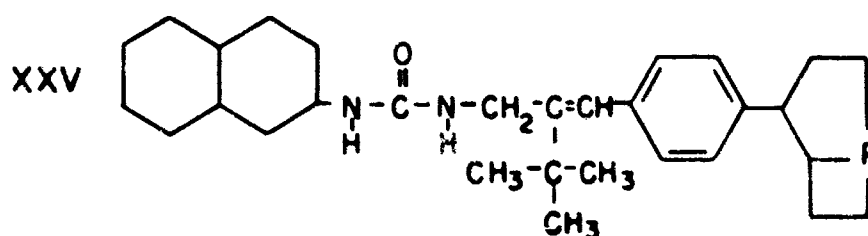
Type b multiplications are handled in an analogous way, with the copy switch set for forward copying.

To see how the routine handles type c, above, multiplication, we can generalize from the foregoing example. This type is recognized by a multiplier followed by a slash when the slash stack is empty. The atom number for the last atom symbolized before the multiplier is stored in JKEEP. When the terminal slash is encountered, the last nonterminal atom within the slashes is under current consideration and is joined to a duplicate of the last atom of the repeated string. The copying routine is called with the "reverse" switch set. If the multiplier is greater than 2, then the branched atom stack will indicate the additional points of attachment within the asymmetric central unit (cf. structure XI). Any residual connections within the slash marks then imply methyl groups.

Polymer-type multiplication, type d, is handled straightforwardly by "forward" copying of all of the atoms symbolized between the pair of slashes.

Since there is at least a superficial reason to expect confusion between the "&" for an explicitly cited methyl group and the "&" ending a branch containing either a variable valence atom or a ring, we shall consider the relevant contexts. When an "&" (which is not immediately preceded by another "&") is reached, there are three possibilities -- the preceding atom symbol may be "X," "Y," "K," or "N"; it may be strictly terminal; or it may be a nonterminal symbol other than "X," "Y," "K," or "N." If the preceding atom symbol is one of these branching symbols, then the "&" signals methyl contraction, as do any succeeding ampersands until the branching requirements of the atom have been satisfied. (Cf. notation XVIII.) If the atom symbol preceding the "&" is a nonterminal symbol other than one of these branching symbols, then the first "&" signals the end of a chain. If there is a branching carbon or nitrogen at the top of the branched atom stack, and further "&'s" follow, then the succeeding ampersands signal methyl contractions until the branching requirements of the indicated atom have been satisfied. (Cf. notation XIX.) For either case of a nonterminal atom symbol preceding the first "&," each further "&" then indicates that all branches attached to one variable valence atom have been cited. (Cf. notations XX and XXI.) For each such succeeding "&" the top of the variable valence stack is fetched; if it is a ring flag, then the corresponding ring is discarded from further consideration; if it is an atom number, then the top of the branch symbol stack

- XVIII ZX&&1-SI-Q00
 XIX ZX2&&-SI-Q00
 XX Z1YPGY&&&YG-SI-Q00
 XXI Z1YPGY2&&&YG-SI-Q00
 XXII QYGVR_ER_DG&&_CY
 XXIII ZY&2Y1-SE- *2Q2Q2Q&Y&Y1Z
 XXIV ZY&2Y1-SE- W2Q&Y&Y1Z



is discarded as long as this equals the top of the variable valence stack; then the top of the variable valence stack is discarded.

If the atom symbol preceding the first "&" is strictly terminal, then this "&" does not end the chain and it must therefore either represent a methyl group or signal the end of a branch containing either a variable valence atom or a ring (cf. notation XXII). Note that a methyl contraction must refer to the branch symbol immediately preceding it and that the implied methyl contraction is limited to the last branch symbol (and to "X," "Y," "N," or "K") so that there can be no ambiguity in the meaning of the top of the branched symbol stack.

Rule 8b⁴ appears to require one "&" for each branch point encountered while backing down the branch to the point where ciphering resumes. This complicates the algorithm and appears to be unnecessary to the notation system. It is agreed that one "&" per variable valence branch point would suffice;* however, this modification has not (yet) been adopted.

There is possible ambiguity due to symbols which may or may not be branching symbols depending on valence. We have adopted the asterisk as a flag to follow such a symbol when it is a branching symbol (cf. notation XXIII), except when the only branches attached are represented by "W" (cf. notation XXIV).

The existing algorithm has a known bug in its handling of branches. It will decode the notation T45PTJ_ER_DIUYXIMVM_CL66TJ, for structure XXV, in such a way as to yield the connections Y-X-I-M, i.e., it does not in this instance take cognizance of Rule 10,⁴ which demands deletion of the "&'s" in a methyl contraction on the only or last branch symbol in the (uncontracted) notation.

In order to remove this bug, a back connection to a branch point must not be made until the notation has been completely processed; i.e., until the end of the cipher has been reached. From the above example, it is evident that the last branch point in the notation can be determined only after all characters have been examined. This suggests to us that rules other than Rule 10 may lead to similar situations, and that similar bugs remain to be discovered.

3. Benzene Notations.

We begin by considering a simple example: Structure XXVI is ciphered WNR_CZ_DZ.

* Smith, E. G. Private communication. October 1966.

The first two characters are processed as indicated in the preceding section, a subroutine handling "WN" in a manner similar to the handling of an initial "Z." When the "R" is encountered, six carbons are entered into the connection table with the first and last atoms connected to each other and with nonlocalized multiple bonds indicated:

1	O	(2, 2)		
2	N	(2, 1)	(2, 3)	(0, 4)
3	O	(2, 2)		
4	C	(0, 2)	(1, 5)	(1, 9)
5	C	(1, 4)	(1, 6)	
6	C	(1, 5)	(1, 7)	
7	C	(1, 6)	(1, 8)	
8	C	(1, 7)	(1, 9)	
9	C	(1, 8)	(1, 4)	

Next, a temporary block of six words is pushed onto the ring stack. For our example this block would look as follows:

1	0	4	A
0	1	5	B
0	1	6	C
0	1	7	D
0	1	8	E
0	1	9	F

The 1 in the left-hand position of the first word indicates a benzene ring, in anticipation of the routine enlarged for processing general cyclic structures. The next field contains the number of substitutable positions left for each ring atom, decremented each time a substituent is attached. The third field has the atom number corresponding to the locant given in the right-most field. When, in our example, the locant "C" is encountered, the storage block for the current ring is searched for the locant and the corresponding atom number (here 6) is stored as the back connection, the second field is decremented, and processing continues with the substituent symbol following the locant. After the first such substituent has been processed, the connection table appears in part as:

6	C	(1, 5)	(1, 7)	(0, 10)
:				
:				
9	C	(1, 8)	(1, 4)	
10	N	(0, 6)	(0, 11)	(0, 12)
11	H	(0, 10)		
12	H	(0, 10)		

When an "&" is encountered that finishes a branch containing a ring (cf. cipher XXII), the current ring block is discarded and processing returns to the previous ring. In this example, a block is set up for the first "R" and the locant "E" is interpreted with its aid; then a second block is pushed onto the stack for the second "R." The locant "D" is interpreted from the second block. The "&" causes the second block to be erased, and the locant "C" is interpreted again from the first block.

For a cipher with type e multiplication (cf. notation XIII), the locants cited before the ring are stored in a table. When the ring block has been created for the "R," these locants provide the back connections for the repeated string. Type f multiplication (cf. notation XIV) is handled in much the same way except that it is the hyphenated locants that are packed into a table.

The analysis of type g multiplication is not debugged in the current version of WISCO. For either of the two cases, expansion of the multiplied group (as intended) must be the last item in processing the notation, because the counts are implicit. When either "R" or "/" is encountered, an appropriate flag corresponding to the proper ring can be pushed onto the top of a stack. When the end of the cipher is reached (and all other matters have been attended to), the residual locants on the ring, as indicated by l's in the second field of the atom words of the ring stack, provide the back connections for the multiplied strings.

The copying routine requires knowing the atom number for the back connection and the limits of the connection table to be copied. In "forward" copying, the replicate of the first atom of the repeated portion is back-connected to the atom indicated above; all other atom numbers in the (i, j) couples are edited during copying by adding a constant to correspond to the position in the connection table of this portion after replication. For "backward" copying, the editing of atom numbers requires subtraction from a constant.

4. General Cyclic Notations.

The basis devised for an algorithm to effect the complete analysis of cyclic notations is the separation of those portions of the notation which lie between the delimiters L and J or T and J into distinct fields. Each of these fields deal with one of the several types of features which may be cited as parts of the cyclic system. The results of such an analysis would be used as input to an algorithm for creating an atom-by-atom listing of each cited ring, such as is shown for the special case of linearly fused rings (cf. below), and for identifying the nature of the bond between each pair of atoms.

The fields which may be present are initial character, ring declaration, shared bridge locants, unshared bridge locants, multicyclic points, last locant, hetero symbols, unsaturation codes, ring saturation code, and final character. For example, in the notation T6(6_B6_C6_ABC_S_KMJ for the cyclic system shown as structure III, the fields present are: initial character, ring declaration, multicyclic points, last locant, hetero atom, and final character.

The initial character must be an L or T, which is retained for later error checking. The routine then looks for ring information, filling a table with the ring size and the atom number corresponding to the initial (fusion) locant of each ring, in the order of citation. Thus, for our example, the ring declaration would be as follows:

<u>Initial locant</u>	<u>Ring size</u>
1	6
1	6
1	6
2	6
3	6

The end of this portion of the routine is signaled by any one of the following:

- a. Locant followed by a nonnumeric character.
- b. "/".
- c. Space followed by a number.
- d. "K," "M," "N," "O," "P," "S," "V," "X," "Y," or hyphen followed by a nonnumeric character.
- e. "&" or "T."
- f. Final character "J."

If the ring declaration is followed by a locant and a nonnumeric character (cf. notation IV), then a shared bridge atom, a ring segment symbol, a hydrogen saturation code, or an unsaturation code follows. Such locants are converted to the corresponding number and stored in another table. Similarly, a slash indicates at least one unshared bridge atom (cf.

notation V). A space followed by a number indicates that multicyclic point atoms are cited, and these are similarly stored in a table. For the above example (structure III) the table of such atoms would appear as follows:

1
2
3

As with the ring declaration field, these fields are well-defined and their terminations can be detected immediately. If multicyclic point atoms are present, the last locant pertaining to the cyclic system must also be cited. This is similarly converted to a number and stored for future reference. Any ring segment symbols are then picked up and stored along with the numerical equivalent of their locants; in the example on the previous page

11(= K) M

is the only such entry. Similarly citations of H and U or UU are stored with the (converted) locants.

"K" or one of the other symbols listed under possibility d indicates a ring segment symbol with implied locant A in a monocycle. A hyphen followed by a nonnumeric character presumably introduces a two-letter atomic symbol.

The "T" and "&" symbols are stored and assigned a number according to the order of citation so that they may be paired with the corresponding rings. The occurrence of "J" signals the end of the cyclic portion of the notation, and control must then revert to the calling routine for processing of the tables here generated and of portions of the notation referring to attachments to the cyclic systems, including interpretation of substituent locants. As in the case of benzene rings (cf. p. 149, above), a temporary storage block would be created and pushed onto the ring-stack for each ring system encountered. The number of words required equals the numeric value of the last locant occurring in the ring system plus one word per branched locant. The last locant may be stated explicitly in the notation, or else the number of atoms in the ring system may be computed via the Landee-Bowman equation.⁴

A routine has been written in ALGOL-60 (cf. figure 20) to decode Wiswesser line notations for linearly fused ring aggregates (i.e., ones with neither multicyclic nor enclosed atoms). It assumes that the notation has been analyzed and tabulated as above. On the basis of this information a

two-dimensional array (with its size determined by the number of rings and the maximum ring size) is created which lists each ring and the enumeration of the atoms therein.

If the notation were L_B666J, then the ring declaration table would be:

<u>Initial locant</u>	<u>Ring size</u>
2	6
1	6
1	6

This routine would then fill up the array as follows: The first ring starts with atom number 2 and has size 6:

2 3 4 5 6 7

The second ring starts with atom number 1 so the array would become

2 3 4 5 6 7

1 2 _ _ _ _

Since atom number 2 has been used in a previous ring, we must get the last atom in that ring and proceed (to six atoms) thus:

2 3 4 5 6 7

1 2 7 8 9 10

Again, the next ring starts with atom number 1, but atom number 2 has already been twice used, so we must get the last atom in the last previous ring cited and continue:

2 3 4 5 6 7

1 2 7 8 9 10

1 10 11 12 13 14

A preliminary syntax for this analysis has been developed. This syntax, as given in the table, will apparently generate (or, equivalently, accept) that set of correct Wiswesser notations which corresponds

Table. Preliminary Syntax for Analysis of
Cyclic Portions of Wiswesser Notations

Wiswesser Notation String:

$\langle \text{WLN} \rangle ::= \langle \text{IC} \rangle \langle \text{RD} \rangle \langle \text{ERA} \rangle \langle \text{MCP} \rangle \langle \text{RSS} \rangle \langle \text{HSAT} \rangle \langle \text{UNSAT} \rangle \langle \text{ANTSAT} \rangle \langle \text{TC} \rangle ;$

Initial Character Code:

$\langle \text{IC} \rangle ::= \text{L} | \text{T} ;$

Ring Declaration:

$\langle \text{RD} \rangle ::= \langle \text{LOC} \rangle \langle \text{RSIZE} \rangle | \langle \text{LOC} \rangle \langle \text{RSIZE} \rangle \langle \text{RD} \rangle ;$

Locant:

$\langle \text{LOC} \rangle ::= \langle \text{NULL} \rangle | \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle ;$

$\langle \text{NULL} \rangle ::= ;$

$\langle \text{SPACE} \rangle ::= _ ;$

$\langle \text{LOCMAG} \rangle ::= \langle \text{LETTER} \rangle | \langle \text{LETTER} \rangle \langle \text{MOD} \rangle ;$

$\langle \text{LETTER} \rangle ::= \text{A} | \text{B} | \text{C} | \text{D} | \text{E} | \text{F} | \text{G} | \text{H} | \text{I} | \text{J} | \text{K} | \text{L} | \text{M} | \text{N} | \text{O} | \text{P} | \text{Q} | \text{R} | \text{S} | \text{T} | \text{U} | \text{V} | \text{W} ;$

$\langle \text{MOD} \rangle ::= \langle \text{AMP} \rangle | \langle \text{HYP} \rangle | \langle \text{NULL} \rangle ;$

$\langle \text{AMP} \rangle ::= \& | \& \langle \text{MOD} \rangle ;$

$\langle \text{HYP} \rangle ::= - | - \langle \text{HYP} \rangle ;$

Ring Size:

$\langle \text{RSIZE} \rangle ::= \langle \text{RNGMAG} \rangle | - \langle \text{NONZERO} \rangle \langle \text{DIGIT} \rangle - ;$

$\langle \text{RNGMAG} \rangle ::= 3 | 4 | 5 | 6 | 7 | 8 | 9 ;$

$\langle \text{NONZERO} \rangle ::= 1 | 2 | \langle \text{RNGMAG} \rangle ;$

$\langle \text{DIGIT} \rangle ::= \emptyset | \langle \text{NONZERO} \rangle ;$

Enclosed Ring Atom:

$\langle \text{ERA} \rangle ::= \langle \text{NULL} \rangle | \langle \text{UBA} \rangle | \langle \text{SBA} \rangle | \langle \text{UBA} \rangle \langle \text{SBA} \rangle ;$

Table (Contd)

Unshared Bridge Atoms:

<UBA> ::= /<ATLOC> ;

<ATLOC> ::= <LOCMAG> | <LOCMAG> <ATLOC> ;

Shared Bridge Atoms:

<SBA> ::= <LOCMAG> | <LOCMAG> <SBA> ;

Multicyclic Point Atom:

<MCP> ::= <NULL> | <MCP2> <SPACE> <LOCMAG> ;

<MCP2> ::= +1<LOCMAG> | +1<MCP2> <LOCMAG> ;

[Note: n successive occurrences of +1 are written as the number n.]

Ring Segment Symbols:

<RSS> ::= <NULL> | <SPACE> <LOCMAG> <SYMBOL> <RSS> | <SYMBOL> <RSS> ;

<SYMBOL> ::= <WISLET> | - <ABC> <ABC> - ;

<WISLET> ::= A | B | H | K | M | N | O | P | S | V | W | X | Y | <HAL> ;

<HAL> ::= E | F | G | I ;

<ABC> ::= <LETTER> | X | Y | Z ;

Hydrogen Saturation:

<HSAT> ::= <NULL> | <SPACE> <LOCMAG> H <HSAT> ;

Unsaturation Code:

<UNSAT> ::= <NULL> | <SPACE> <LOCMAG> <USYM> <EXLOC> <UNSAT> ;

<USYM> ::= U | UU ;

<EXLOC> ::= <NULL> | - <LOCMAG> ;

Ampersand and T unsaturation:

<ANTSAT> ::= <NULL> | & <ANTSAT> | T <ANTSAT> ;

Terminal Character Code:

<TC> ::= J

to the set of possible nonbenzene cyclic structures excluding the so-called ring-of-rings structures, whose notations begin with L- or T-. It can be used to formulate an algorithm either to generate or to accept individual notations. The syntax distinguishes in those portions of the notation beginning with L or T and ending with J the following fields, each of which contains a specified type of information:

Initial character (L or T)

Ring declaration

Fusion locant (may be null)

Ring size

Enclosed ring atoms

Unshared bridge atoms

Shared bridge atoms

Multicyclic points and last locant

Ring segment symbol (required if initial character is T)

Carbon saturation (H)

Unsaturation (U or UU)

Ring saturation (& and/or T)

Final character (J)

The first two fields and the last must be present in any nonbenzene cyclic notation. The others may or may not be present, depending entirely on the structure under consideration. The symbol D signaling a ring structure with a donor-acceptor bond* is not accounted for by this syntax.

Note, however, that this syntax is too powerful, in the sense that it will generate strings of symbols which are not acceptable Wiswesser notations. For instance, any series of ring declarations can be generated, including invalid ones such as in L B6J. Implementation of restrictions such as that which requires either <RSS> (see below) to be <NULL> or else <SYMBOL> to be V, X, or Y whenever <IC> is L will require a considerable increase in the complexity of the rules.

* Landee, F. A., Bowman, C., and Reslock, M. H. Private communication. November 1965.

The syntax is presented in the so-called Backus Normal Form.^{6,7} The metanotational terms (e.g., <IC> for initial character) are enclosed in pointed brackets, and each such appears exactly once on the left-hand side of a rule (i.e., a string of the form ... ::= ...). The symbol ::= is read "is replaced by" or "is written as." The vertical line indicates alternatives; concatenation means "and." Thus the rule

$$\langle \text{IC} \rangle ::= \text{L T}$$

means that any occurrence of the symbol <IC> is replaced by L or T; the rule

$$\langle \text{UBA} \rangle ::= / \langle \text{ATLOC} \rangle$$

indicates that <UBA> is rewritten as / followed by <ATLOC>.

Note that some rules are recursive: the symbol on the left of a given rule may also appear in the right-hand side thereof, as in

$$\langle \text{HYP} \rangle ::= - - \langle \text{HYP} \rangle .$$

This rule gives rise to the string of hyphens which indicate a branched locant at a given depth. To apply the rule, rewrite <HYP> as either - or -<HYP>. If the former alternative is chosen, the recursion ends; if not, reapply <HYP>. This sequence is continued until a string of hyphens of the desired length is obtained. Finally, for the fields which are optionally present the alternative <NULL> is given, as in the case of <ERA>, to allow for their nonoccurrence.

As indicated above, this syntax can be used in an algorithm to generate individual notations. The algorithm used to parse a given string will in effect be the inverse of the rule-application procedure outlined above, so that the notation will be accepted as input and will be parsed (i.e., accepted) if and only if a "path" through the syntax can be found such that the path terminates with the rule

$$\langle \text{WLN} \rangle ::= \langle \text{IC} \rangle \dots \langle \text{TC} \rangle$$

An algorithm to "turn around" a procedure of this kind has been discussed by Schneider.⁸

The workings of the syntax will be illustrated by tracing the path which would be taken to generate the string T66_BMT&J. Starting at <WLN>, one is led first to <IC> : T is generated as the terminal symbol. The next symbol in the definition of <WLN> is <RD> : the <LOC> <RSIZE> <RD>

option must be generated in order to get the substring of two ring sizes. <LOC> must generate <NULL> to account for the suppressed A locant, and <RSIZE> must generate <RNGMAG> to get the single digit 6 as a ring size. Going through <RD> again, generating <LOC> <RSIZE> , and proceeding as above generates the second 6 with a suppressed A locant. Since no enclosed ring atoms nor multicyclic points are cited, <ERA> and <MCP> terminate via the <NULL> options in each case. In the case of <RSS> the first option allowing recursion is required in order to cite the locant for the M. The symbol <SPACE> terminates immediately with _ ; <LOCMAC> generates <LETTER> , which terminates with B for the locant. The symbol <SYMBOL> generates <WISLET> which terminates with M. Upon recursion through <RSS> the <NULL> option is generated. Since there are no H-saturations or explicit unsaturations in the cipher the <NULL> options are generated for <HSAT> and <UNSAT> . <ANTSAT> generates first the string T<ANTSAT> ; upon recursion the string & <ANSAT> is generated. This part of the generation is stopped by generating the <NULL> option for <ANTSAT> . The generation is completed by applying the rule <TC> ::=J. For the tree of this generation, see figure 29.

LITERATURE CITED.

1. Hyde, E., Matthews, F. W., Thomson, L. H., and Wiswesser, W. J. Conversion of Wiswesser Notation to a Connectivity Matrix for Organic Compounds. Paper F13. In Abstracts of Papers, American Chemical Society, 153rd National Meeting. Miami Beach, Florida. 9-14 April 1967.
2. Landee, F. A. Computer Programs for Handling Chemical Structures Expressed in the Wiswesser Notation. Paper 3F. In Abstracts of Papers, American Chemical Society, 147th National Meeting. Philadelphia, Pennsylvania. 5-10 April 1964.
3. Wiswesser, W. J. Line-Formula Chemical Notation. Revision of October 1964 by E. G. Smith et al. of Wiswesser's Monograph. Thomas Y. Crowell Co. New York. 1954.
4. Ibid. Revision of April 1965.
5. Tauber, S. J., Bolotsky, G. R., Fraction, G. F., Kirby, C. L., and Reed, G. R. Algorithms for Utilizing Hayward Chemical Structure Notations. In H. Pfeffer, ed. Information Retrieval Among Examining Patent Offices. pp 351-394. Proceedings of the Fifth Annual Meeting (1965) of the Committee for International Cooperation in Information Retrieval Among Examining Patent Offices. Thompson Books, Inc. Washington, D. C. 1966.

6. Naur, P., et al. Report on the Algorithmic Language ALCOL-60. Comm. ACM 3, 299-314 (May 1960).

7. Backus, J. W. The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference. In Information Processing. pp. 125-132. Proceedings of the International Conference 1959. UNESCO, Paris. 1960.

8. Schneider, F. B. Pushdown-Store Processors of Context-Free Languages. Ph.D. Dissertation. Northwestern University. June 1966.

FIGURES

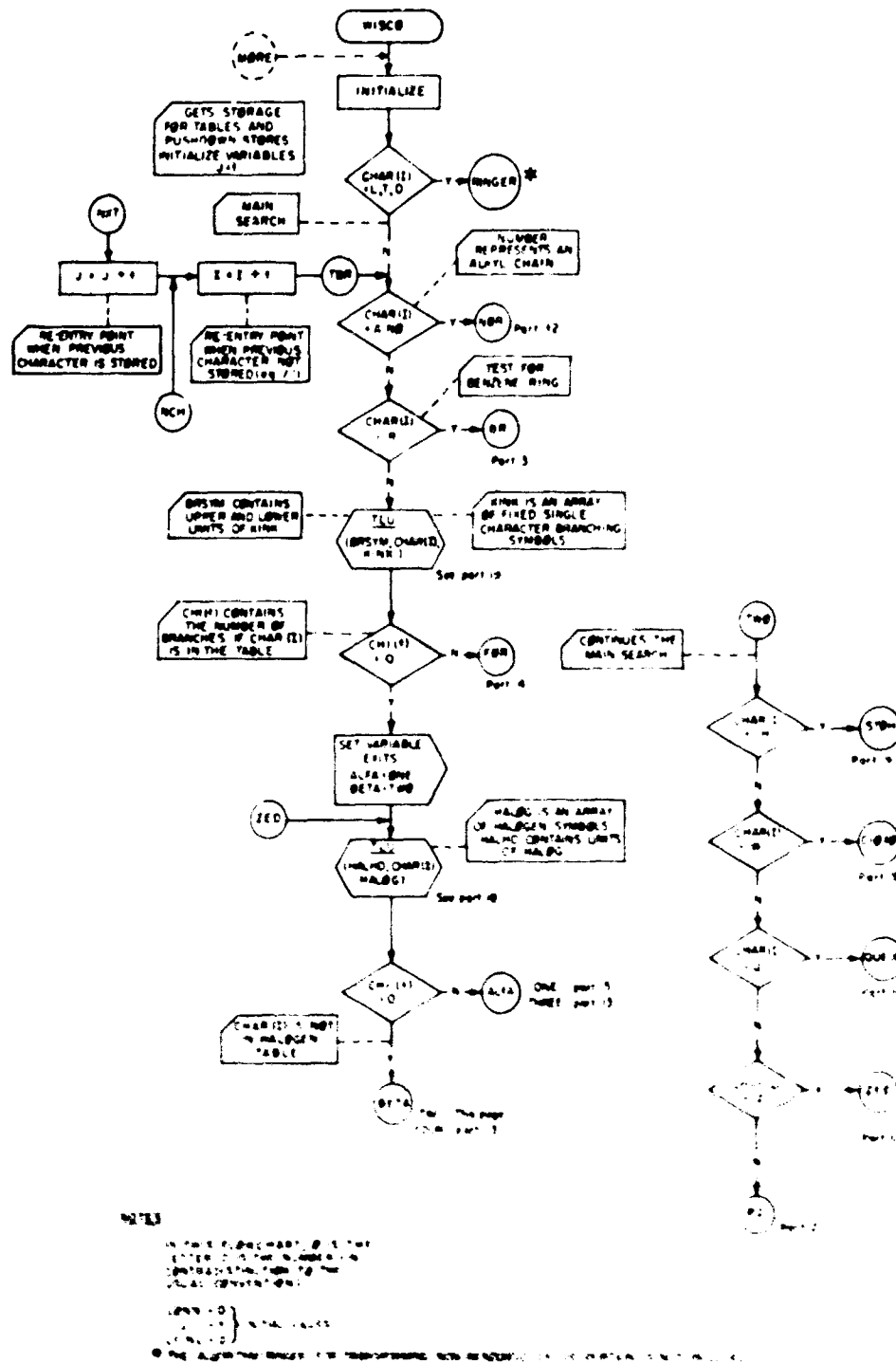


Figure 1. Flow-Chart for Transforming Wiswesser Notations Into Connection Tables - Part 1

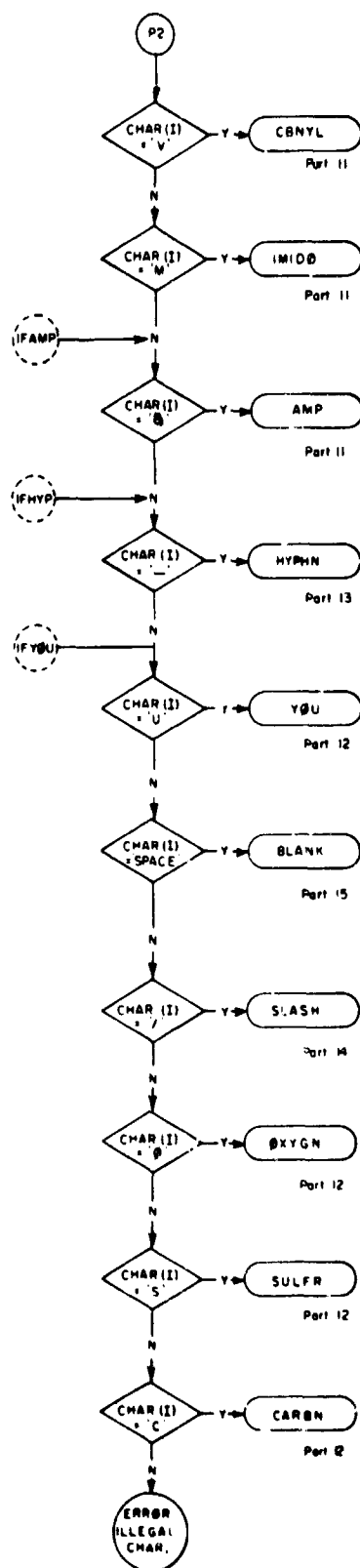


Figure 2. Flow-Chart for Transforming Wiswesser Notations - Part 2

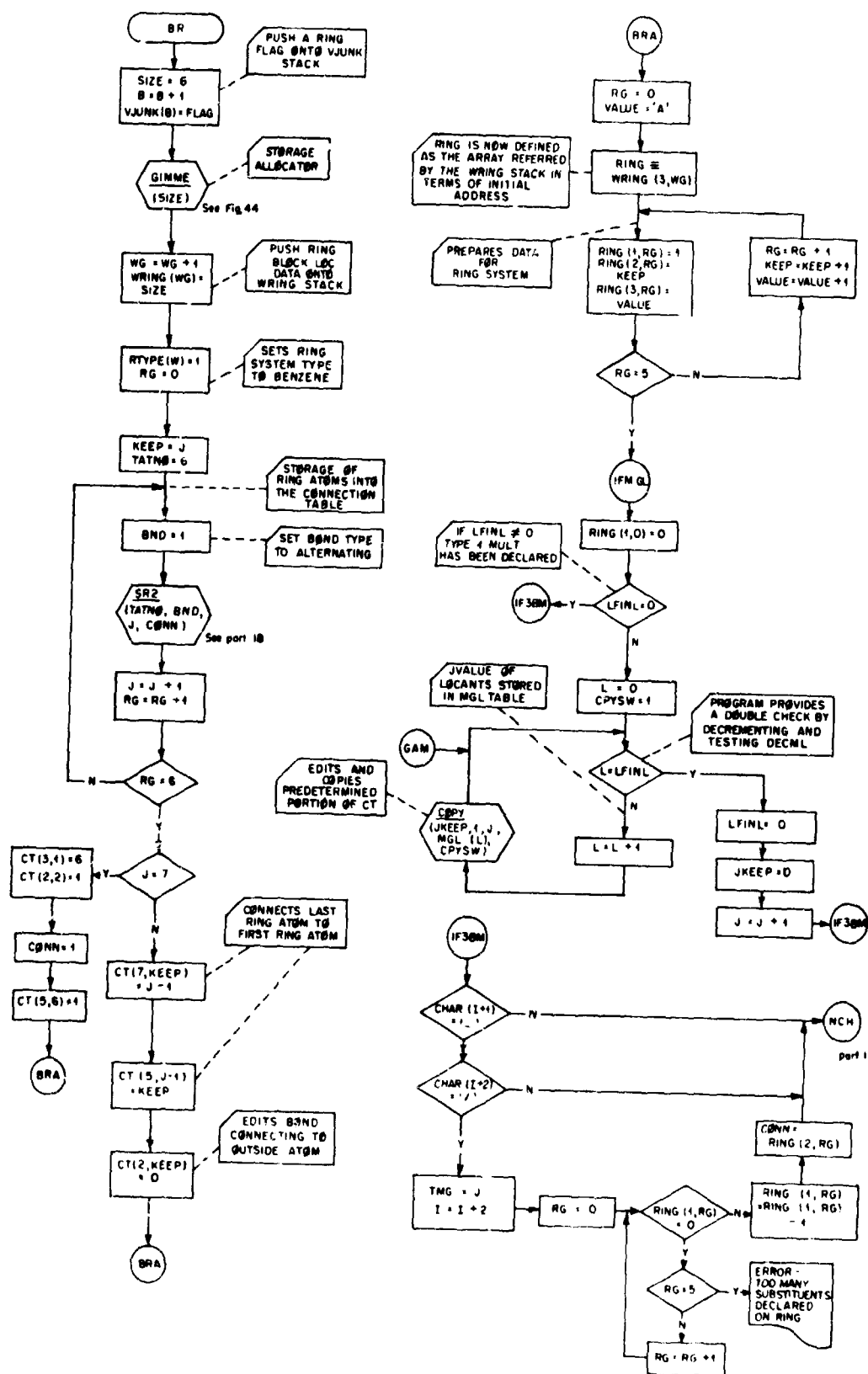


Figure 3. Flow-Chart for Transforming Wiswesser Notations - Part 3

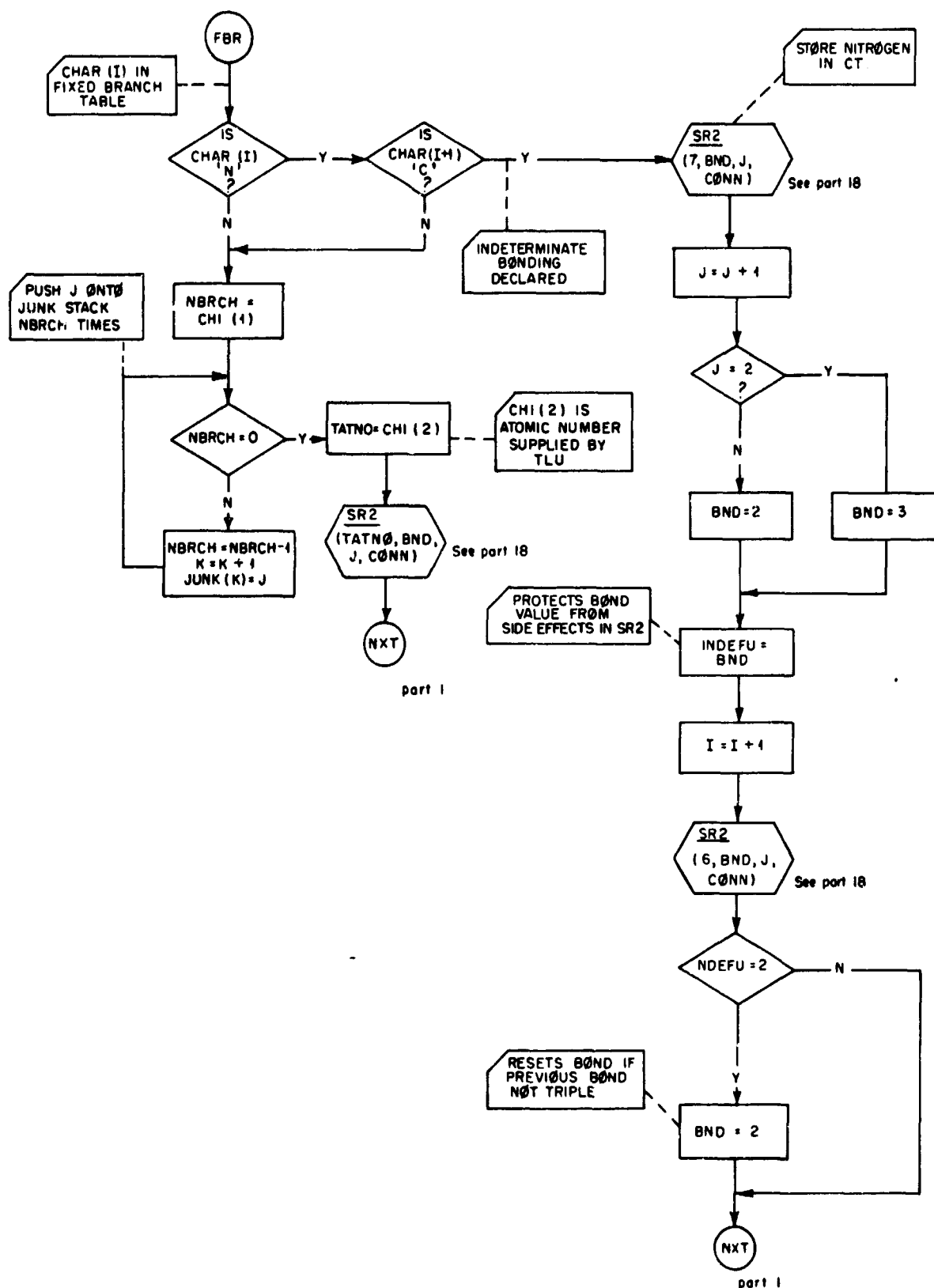
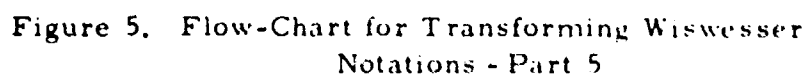


Figure 4. Flow-Chart for Transforming Wiswesser Notations - Part 4



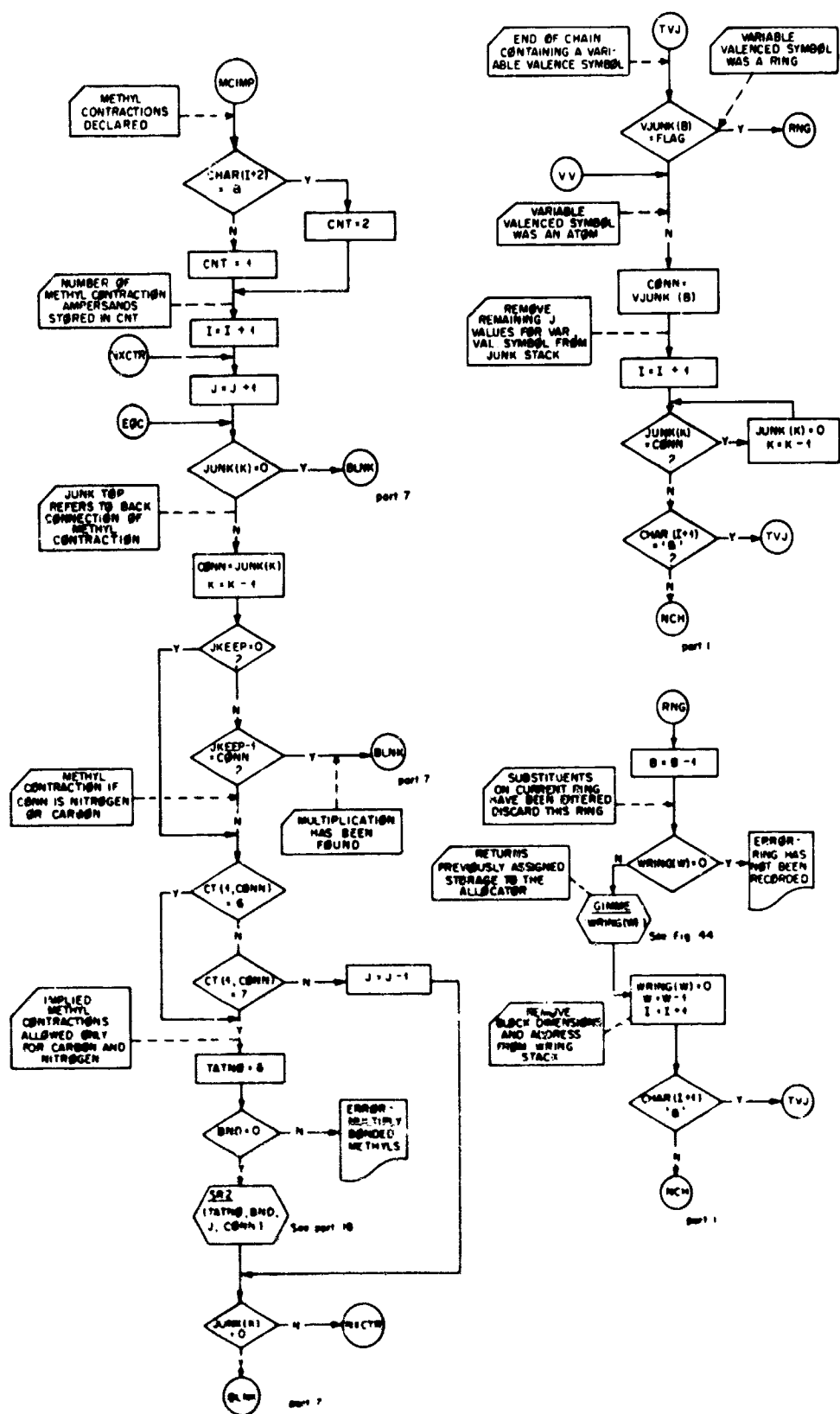


Figure 6. Flow-Chart for Transforming Wiswesser Notations - Part 6

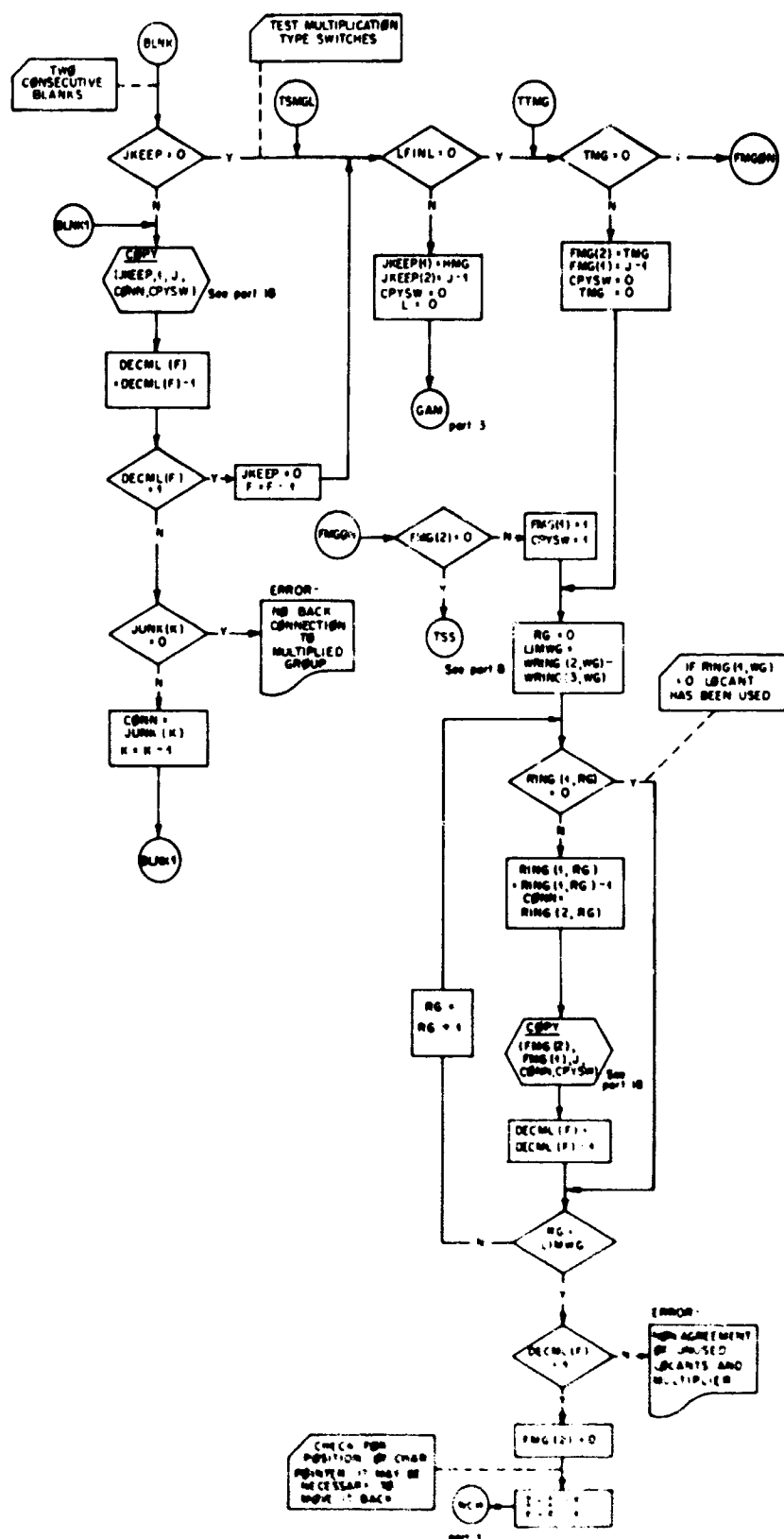


Figure 7. Flow-Chart for Transforming Wiswesser Notations - Part 7

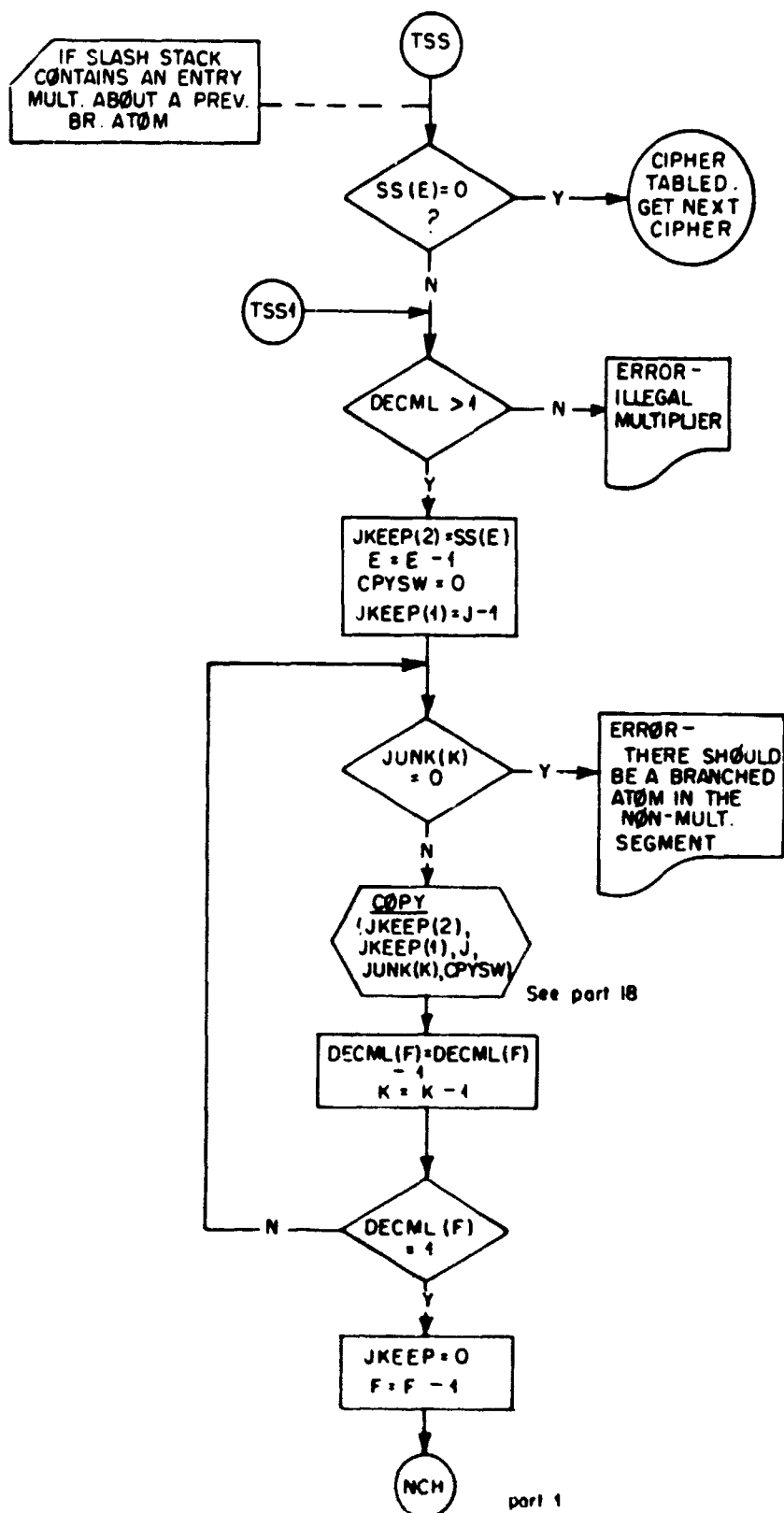


Figure 8. Flow-Chart for Transforming Wiswesser Notations - Part 8

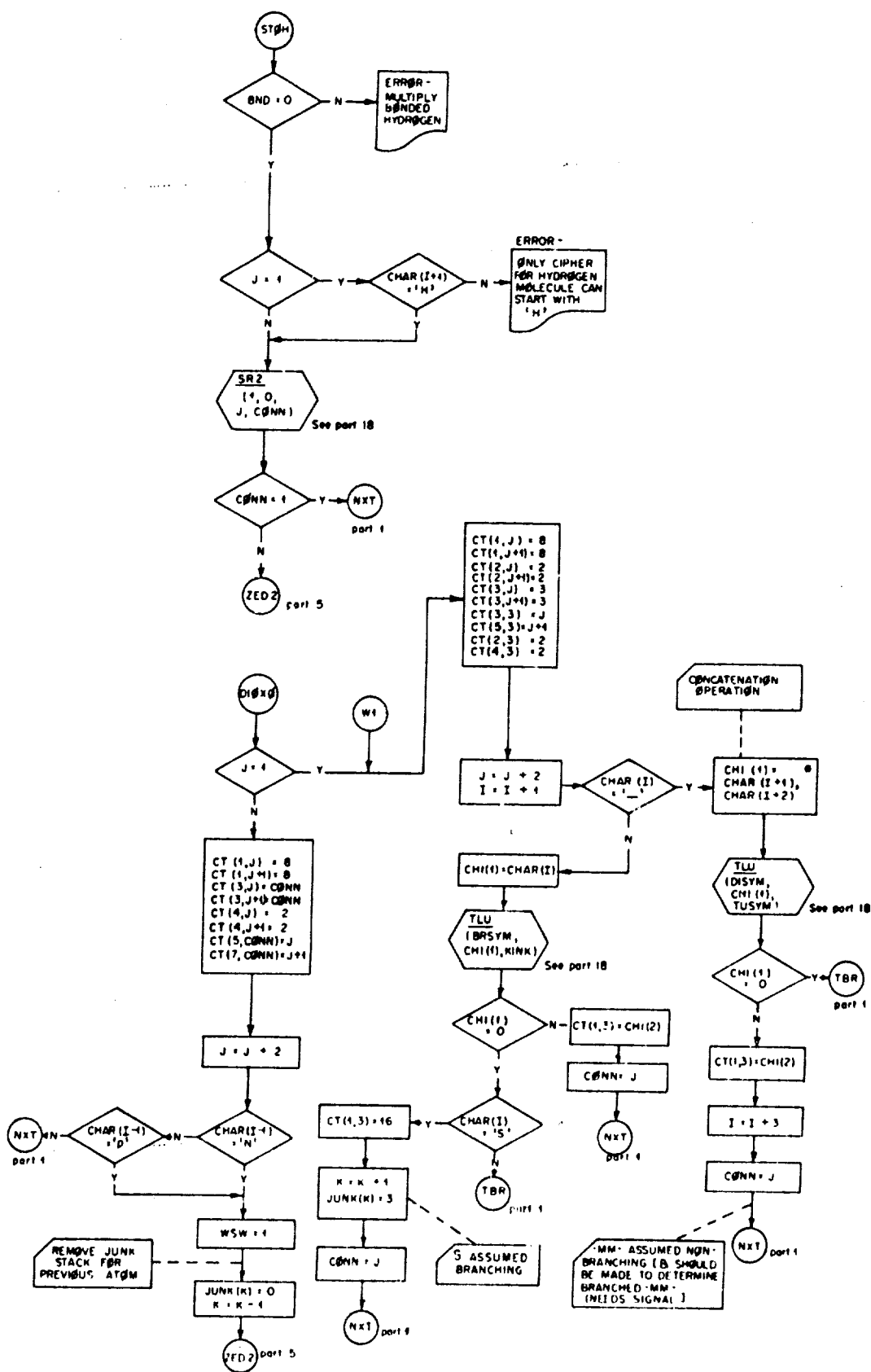


Figure 9. Flow-Chart for Transforming Wiswesser Notations - Part 9

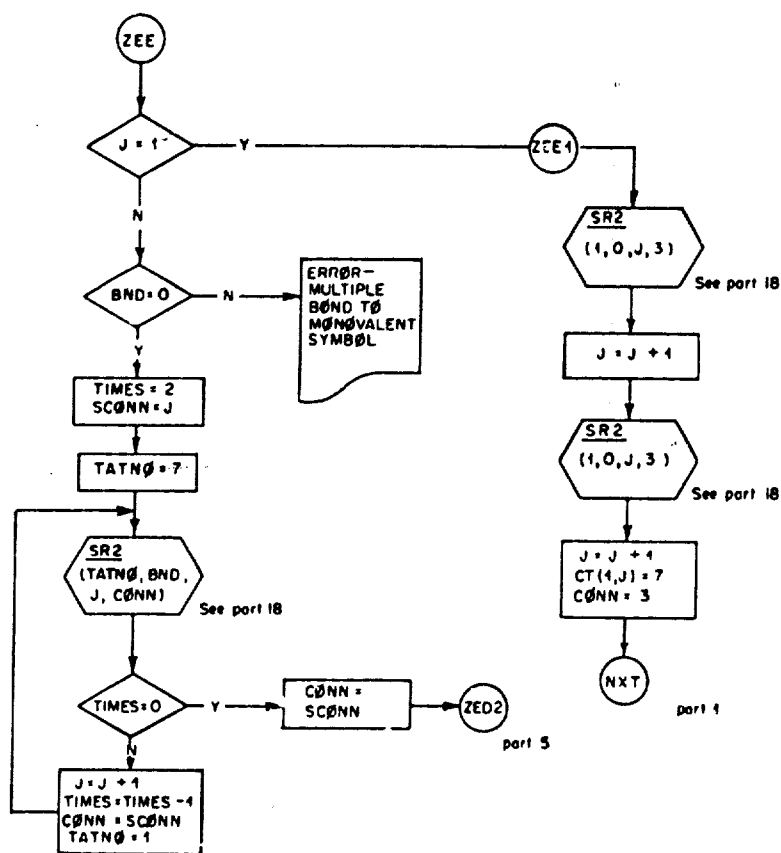
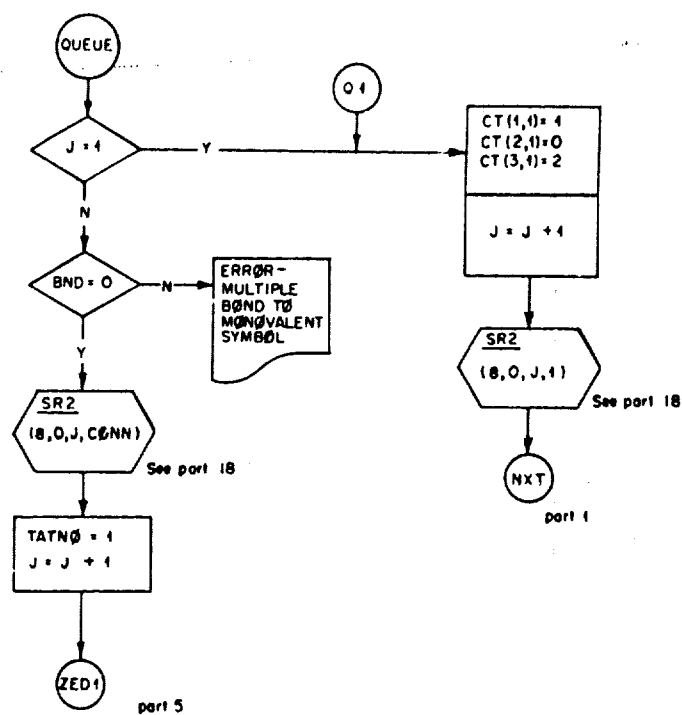


Figure 10. Flow-Chart for Transforming Wiswesser Notations - Part 10

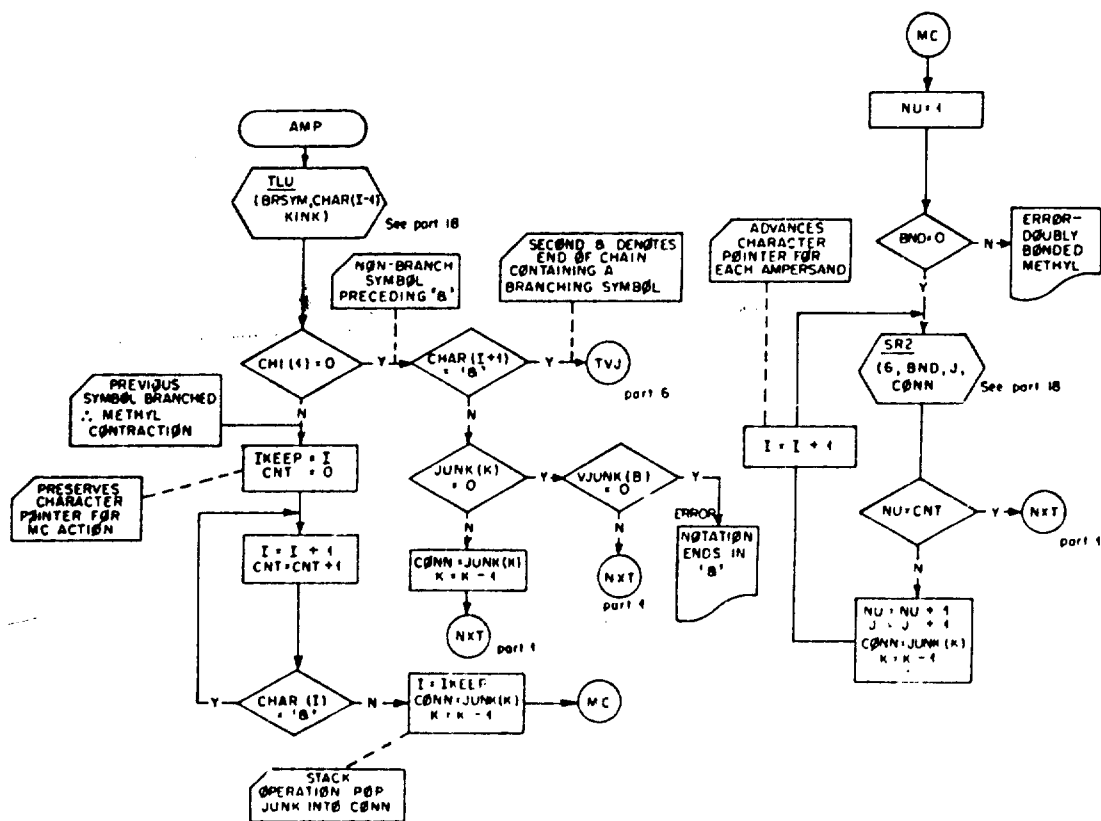
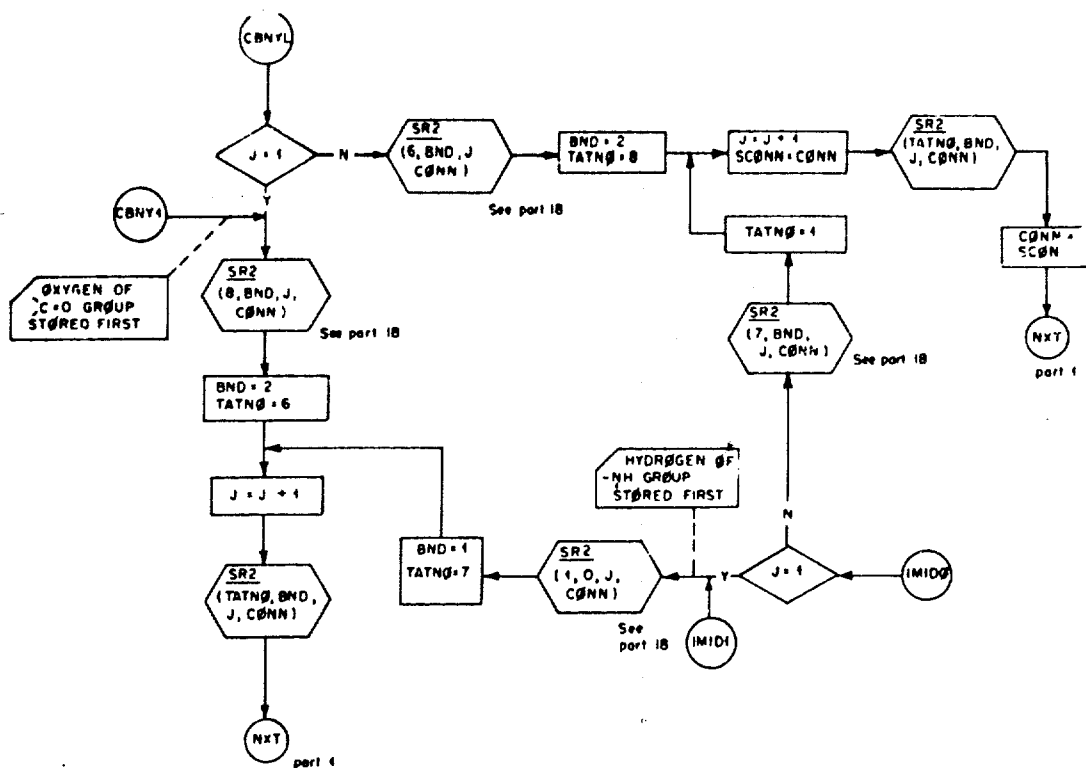


Figure 11. Flow-Chart for Transforming Wiswesser Notations - Part 11

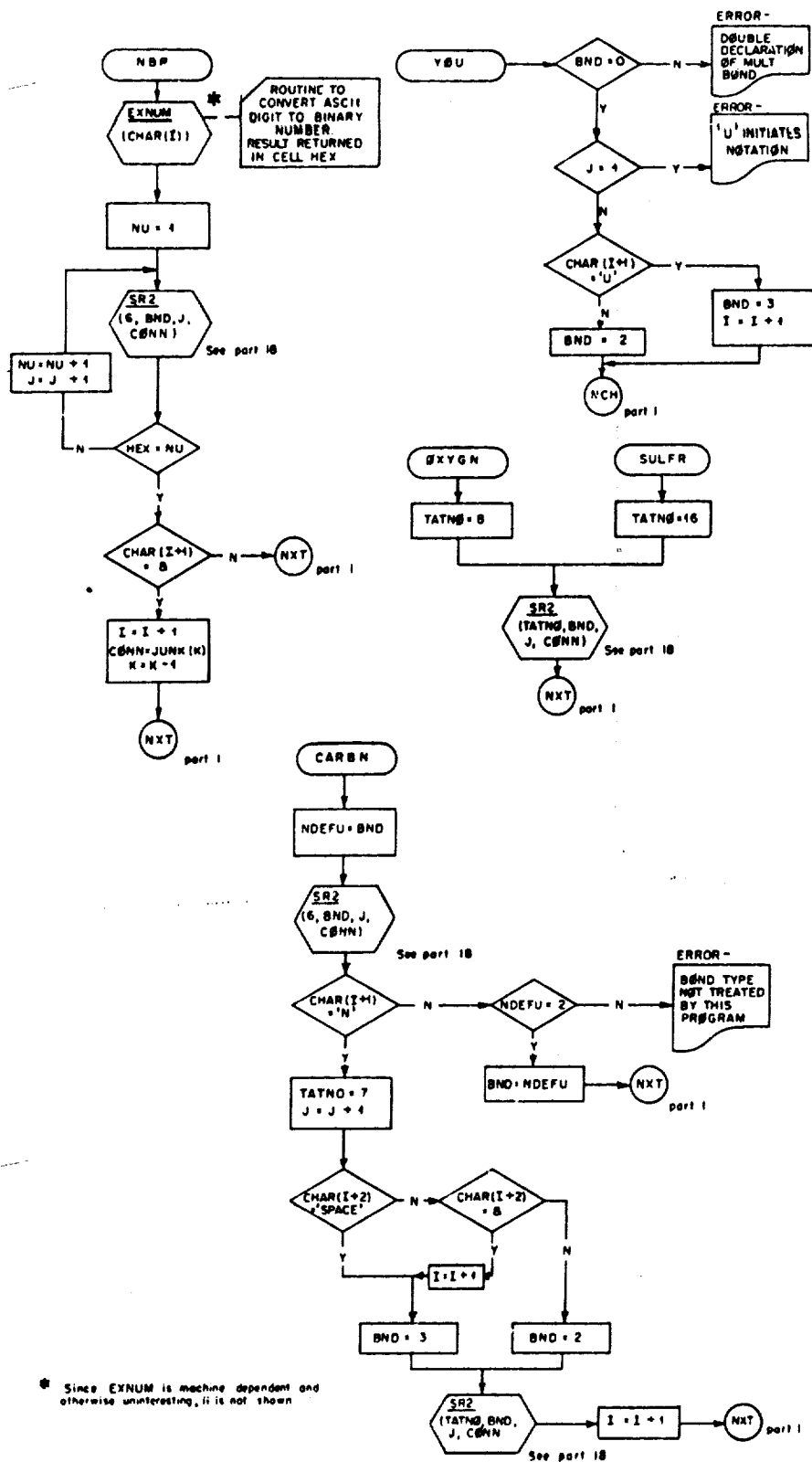


Figure 12. Flow-Chart for Transforming Wiswesser Notations - Part 12

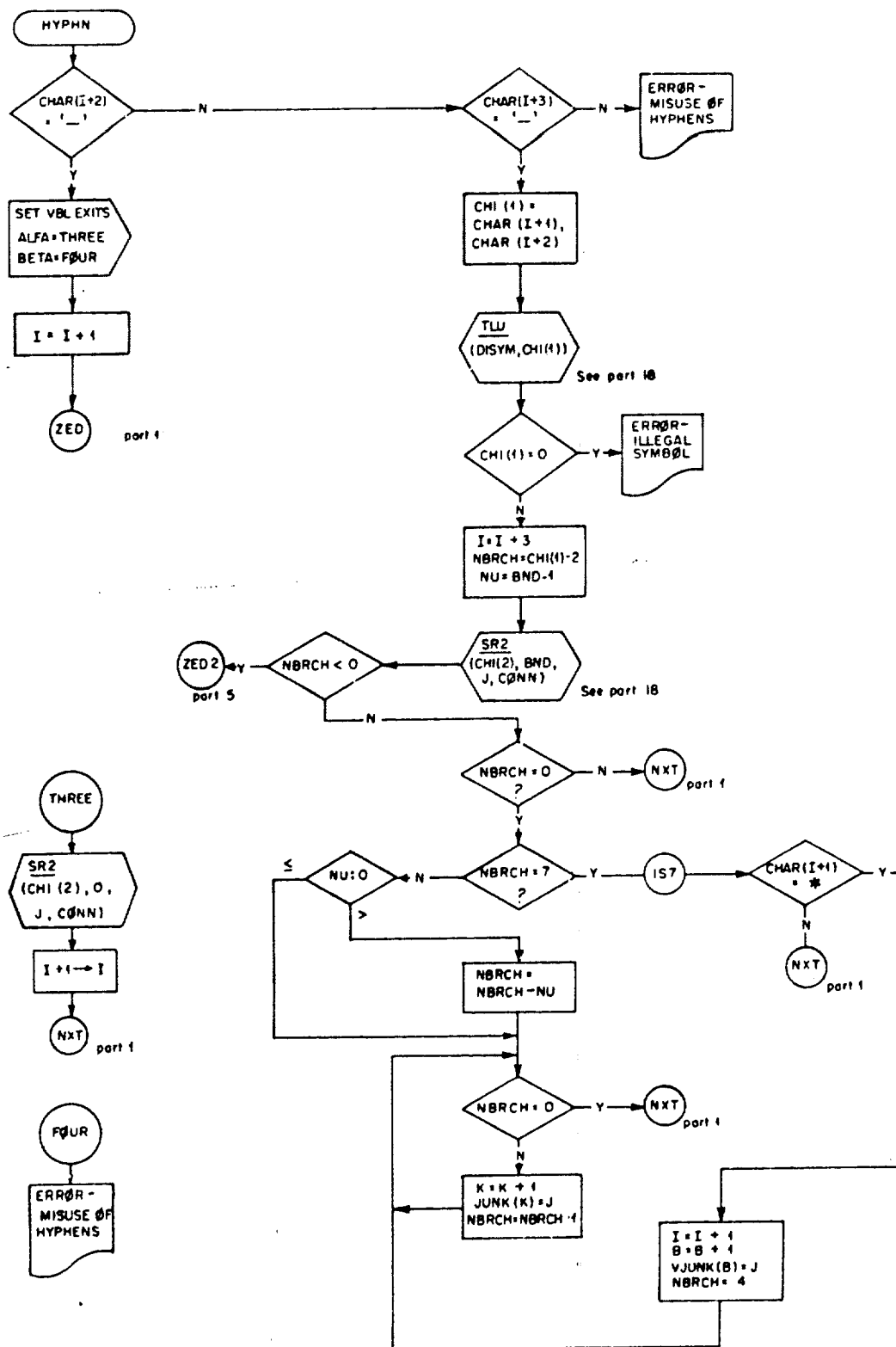


Figure 13. Flow-Chart for Transforming Wiswesser Notations - Part 13

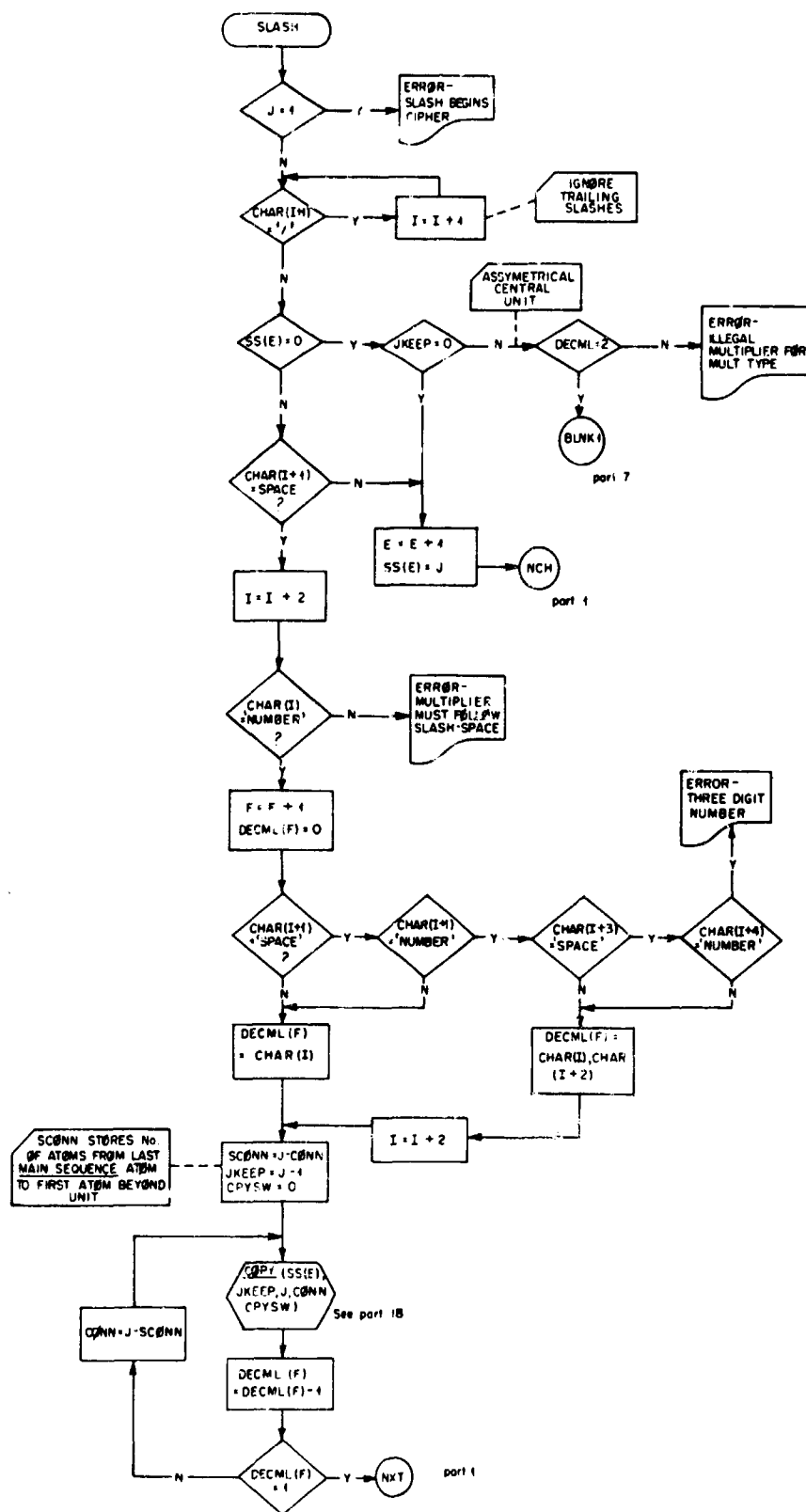


Figure 14. Flow-Chart for Transforming Wiswesser Notations - Part 14

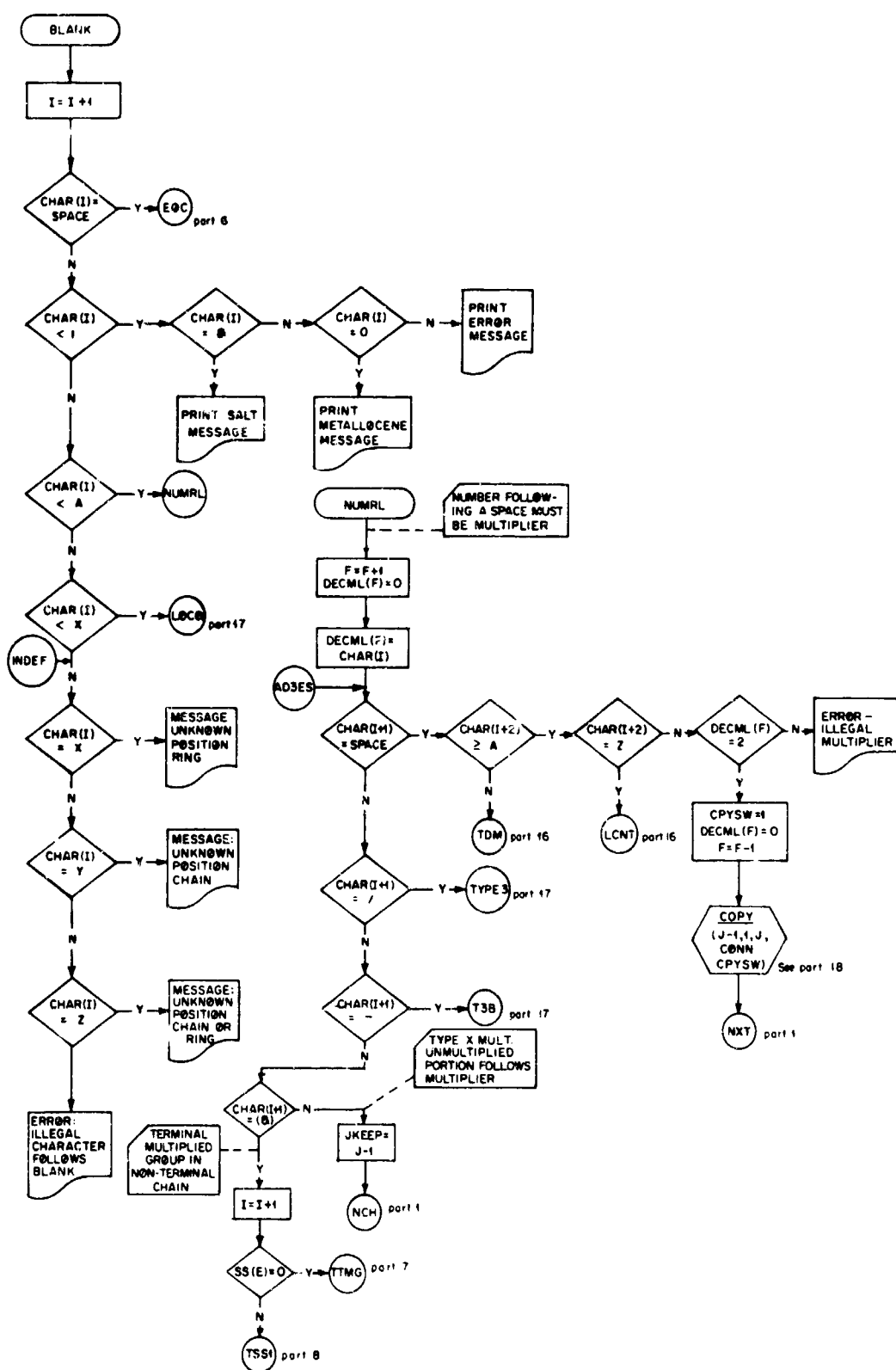


Figure 15. Flow-Chart for Transforming Wiswesser Notations - Part 15

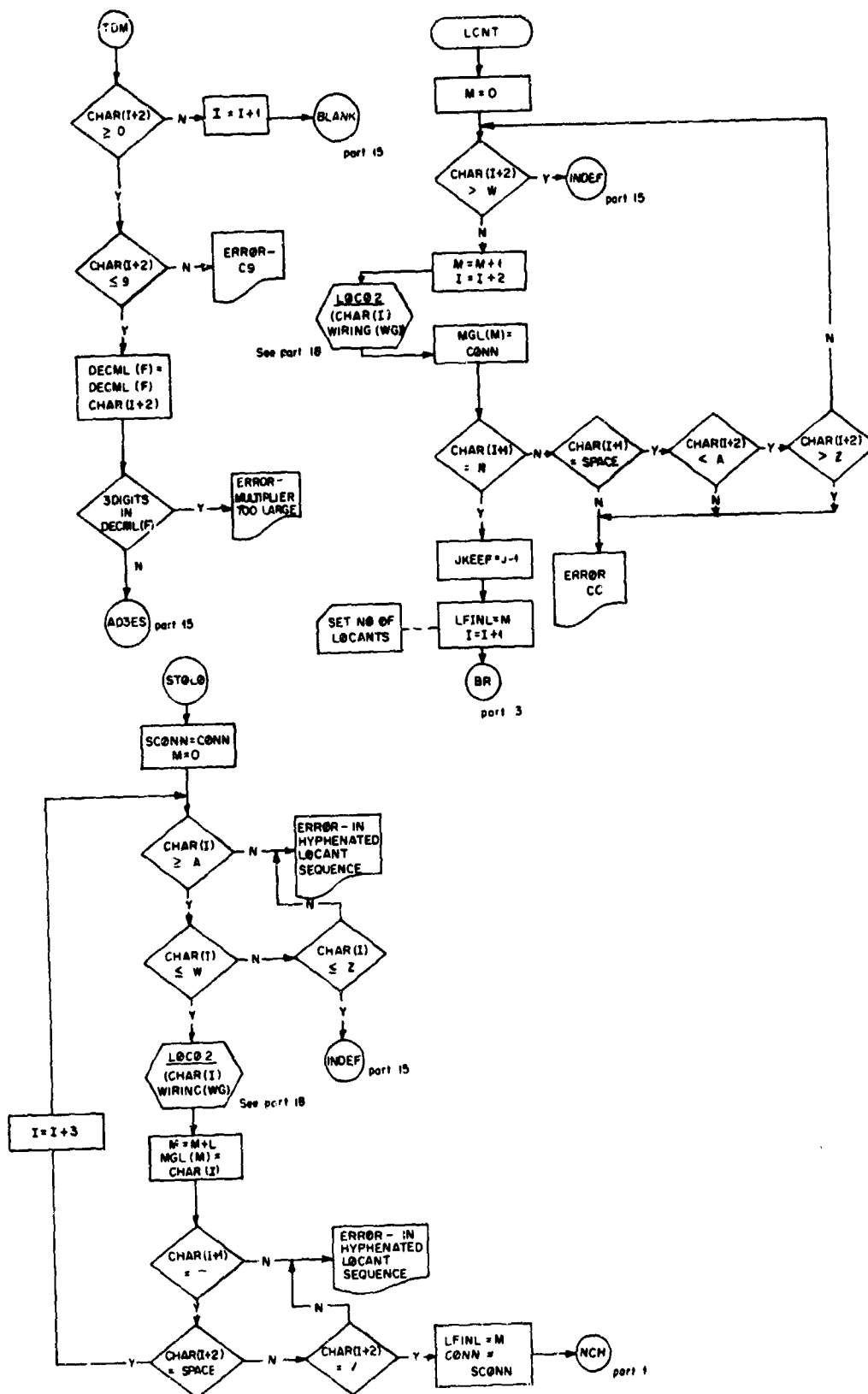


Figure 16. Flow-Chart for Transforming Wiswesser Notations - Part 16

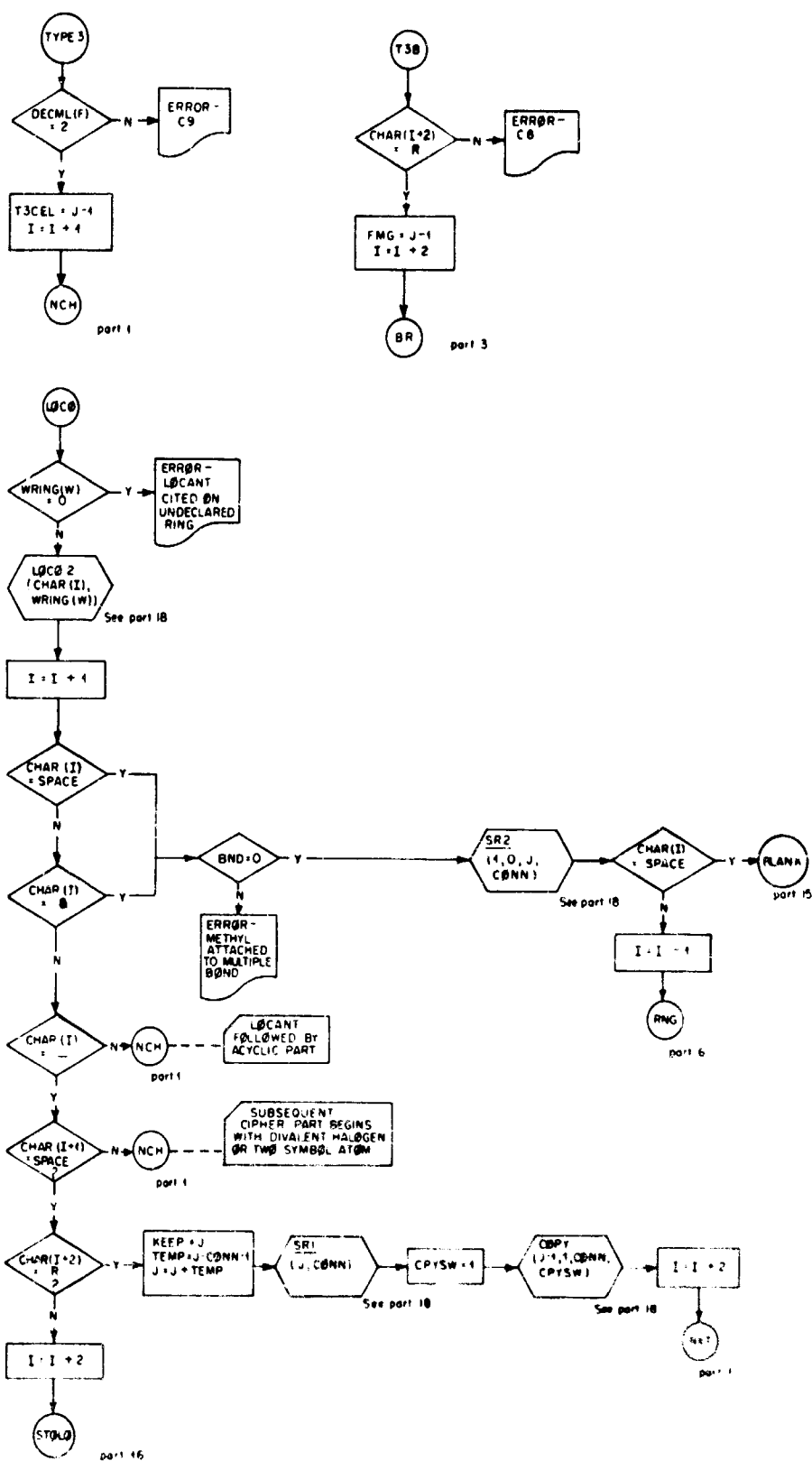
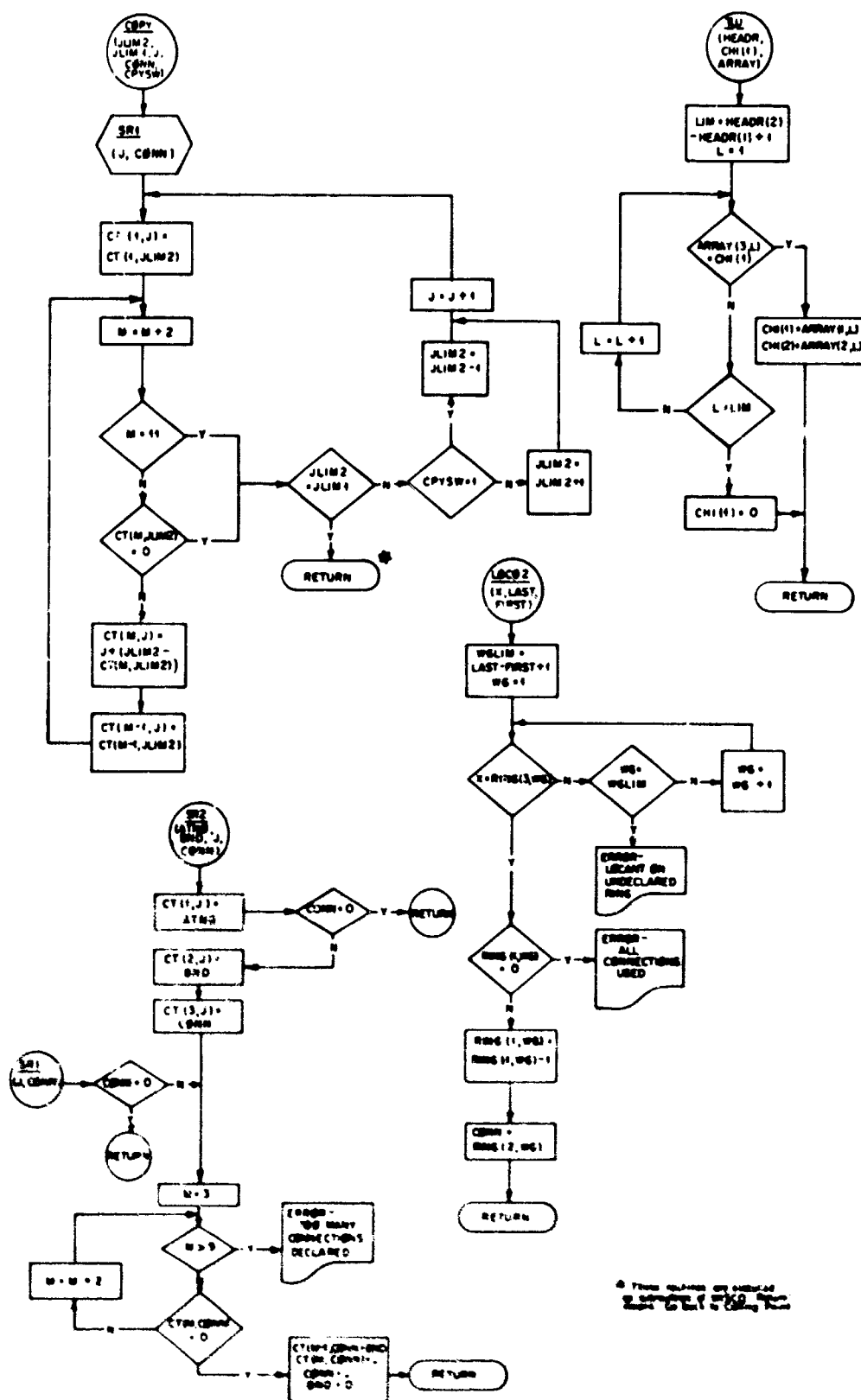


Figure 17. Flow-Chart for Transforming Wiswesser Notations - Part 17



© 1980 BY THE AMERICAN
CHEMICAL SOCIETY
Reprints: See Part 1 of Catalogue

Figure 18. Flow-Chart for Transforming Wiswesser Notations - Part 18

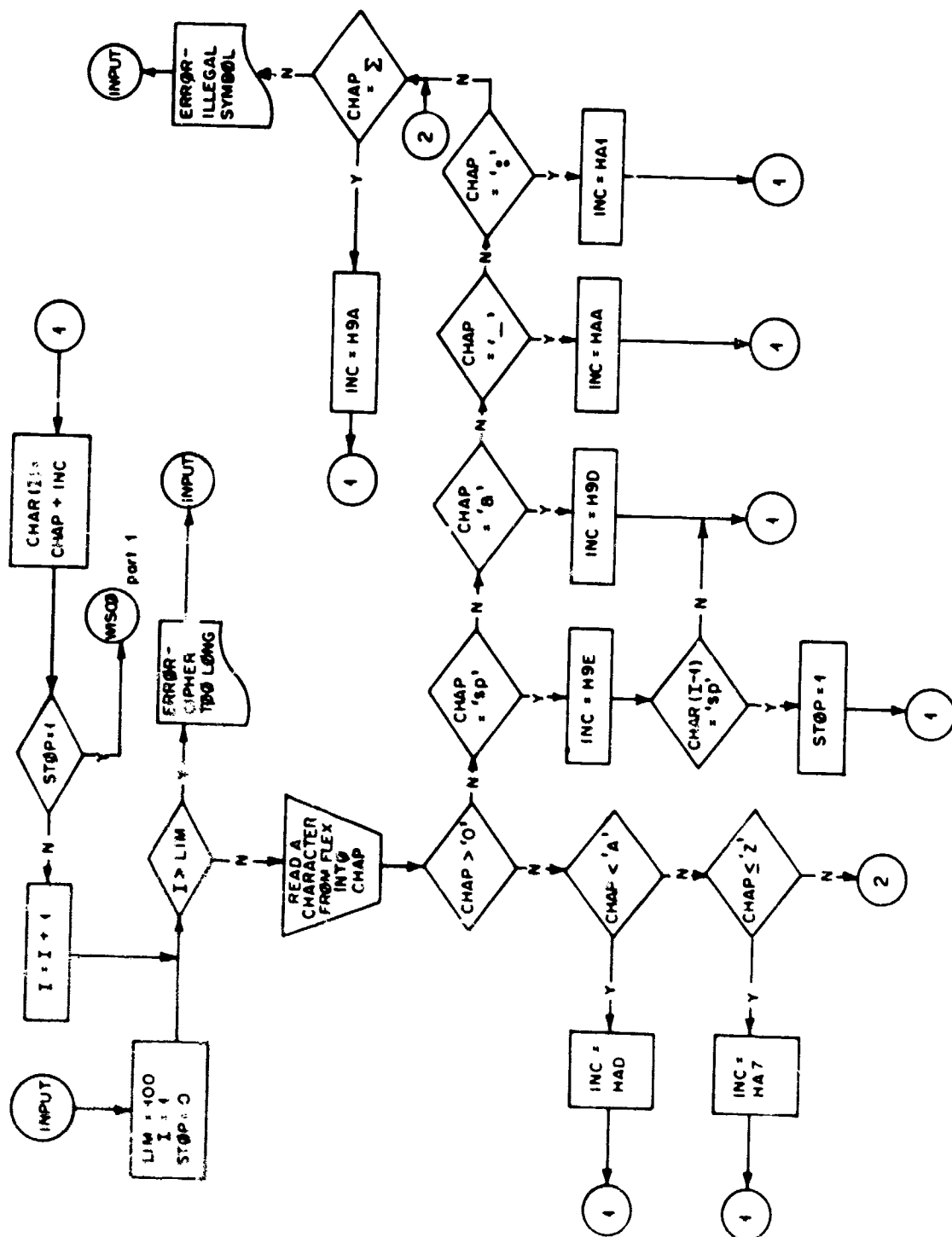


Figure 19. Flow-Chart for Transforming Wiswesser Notations - Part 19

Integer-procedure DECODE(SIZE); comment Initial locant: (ILOC).
 Number of rings: (N);

Comment This procedure is for decoding linear strings of fused rings coded in WLN. All ring sizes and the initial locant of each ring are assumed to have been stored, by earlier parts of the general program in SIZE(R) and ILOC(R) respectively, where $0 < R \leq N$ (N = the number of rings in the fused ring aggregate). This procedure fills a two-dimensional array RING with the enumeration of each atom in each ring. RING is so ordered that within each row every atom is connected to its neighbor, with the last atom in the row being connected to the first atom in the row, i.e., RING(R,J) is connected to RING(R,J $\oplus$ 1) and RING(R,J $\ominus$ 1), where $\oplus$ and $\ominus$ define arithmetic operations modulo SIZE(R);

value N; integer USED,FUSED,N,K,R,J;

integer array SIZE,ILOC[1:N],RING[1:N,1:R];

begin

USED:=0;

R:=1;

GAMMA:K:= ILOC [R];

J:=1;

BETA:RING[R,J]:= K;

if K = USED then K:=FUSED else begin

if J = SIZE[R] then begin

if R=N then go to STOP else begin

FUSED:=K;

USED:=ILOC [R];

R:= R+1; go to GAMMA end end else

K:= K+1 end

DELTA: J:= J+1; go to BETA;

STOP: end DECODE

Figure 20. ALGOL-60 Program for Converting Wiswesser Line Notations of Linearly Fused Ring Systems to Connection Tables.

Notes for Figures 21 through 25:
Flow Chart Subroutine To Examine
Storage After Processing a Wiswesser Notation

DEBUG - A Subroutine of WISCO

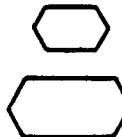
SYMTAB - A table generated at assembly time by the assembler. The length of the table is stored in a location called SYMEND. Each word of the table (SYMTAB(I)) has three fields: SYMTAB(I)₁ contains the name of a location or of a storage area; SYMTAB(I)₂ contains the address corresponding to the name; SYMTAB(I)₃ contains the number of words accessed by the name if a storage area or a flag (all ones) if a named location (i.e., the address of an instruction).



encloses a message to be printed out on-line.



encloses instructions for on-line input.



encloses operations which are machine-dependent and are indicated only in a general statement.



contains comments intended to explain a set of operations and create perspective.

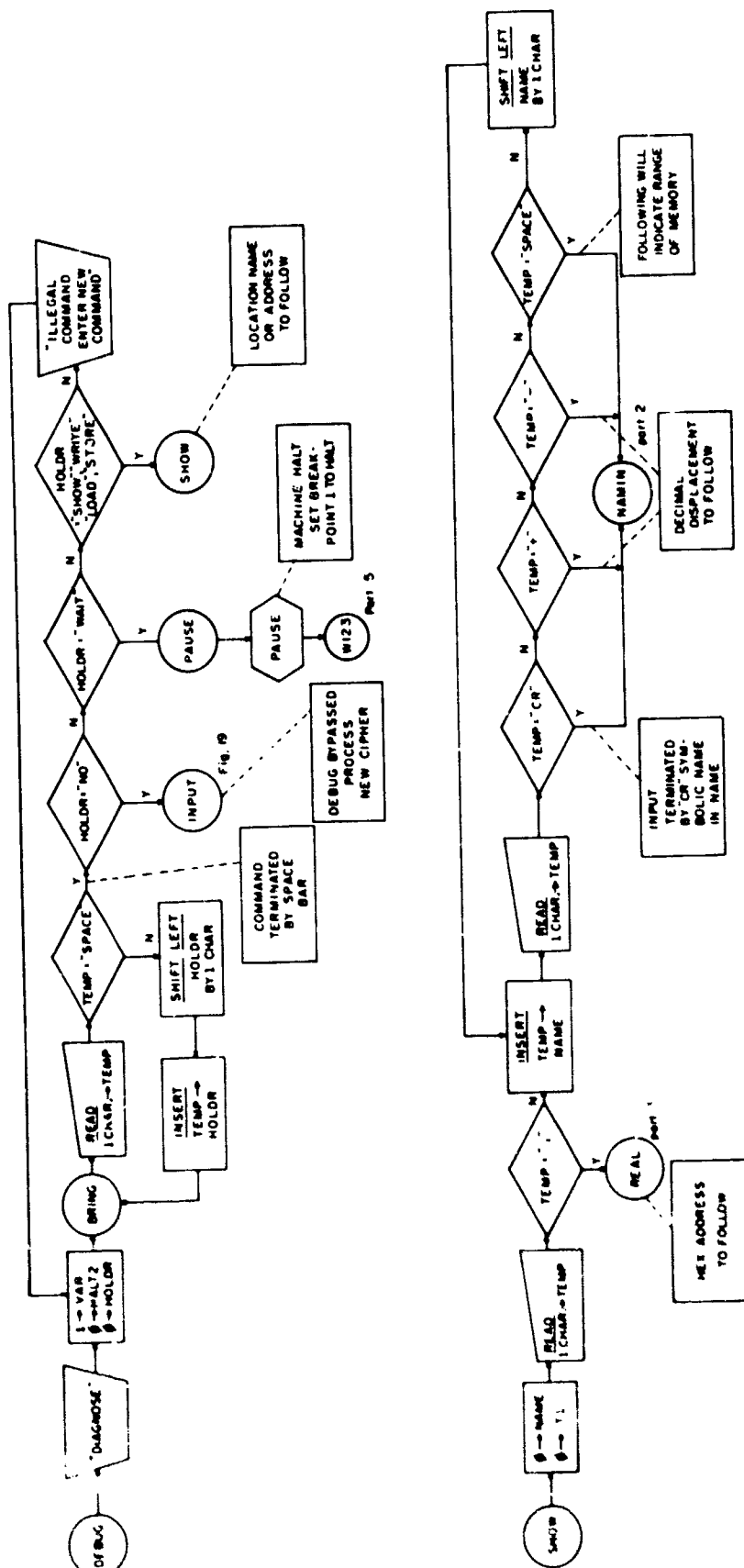
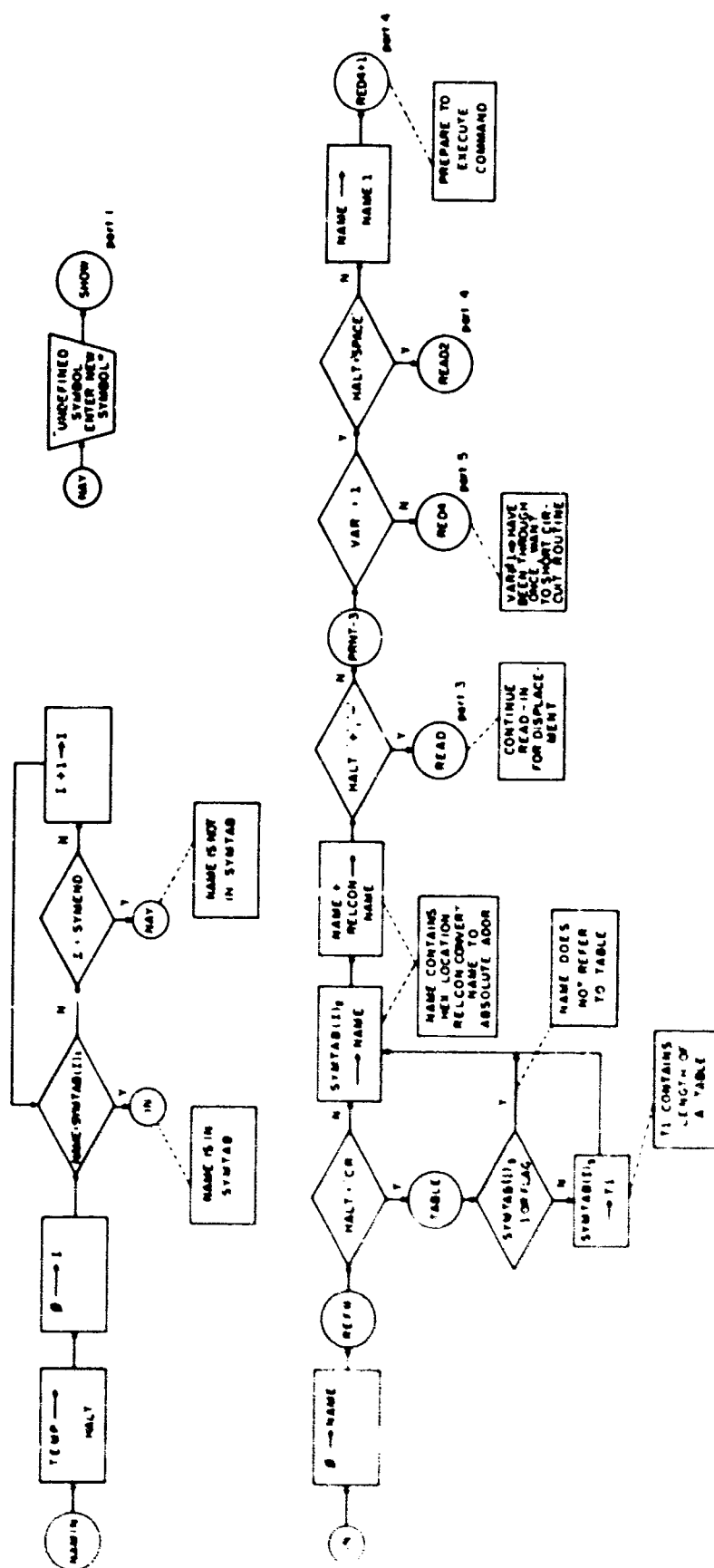


Figure 21. Flow-Chart for Subroutine to Examine Storage After Processing a Wisesser Notation - Part 1



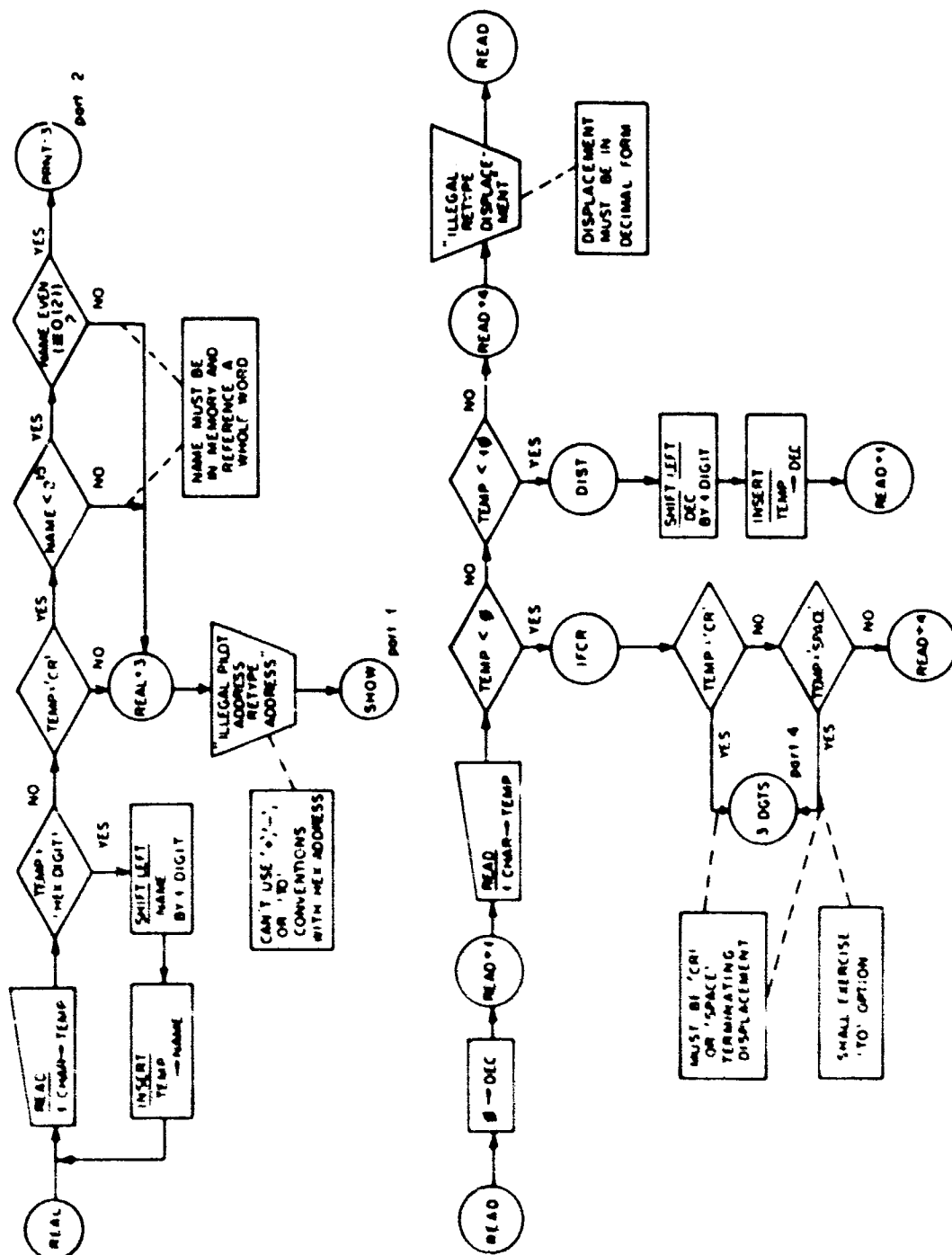


Figure 23. Flow-Chart for Subroutine to Examine Storage - Part 3

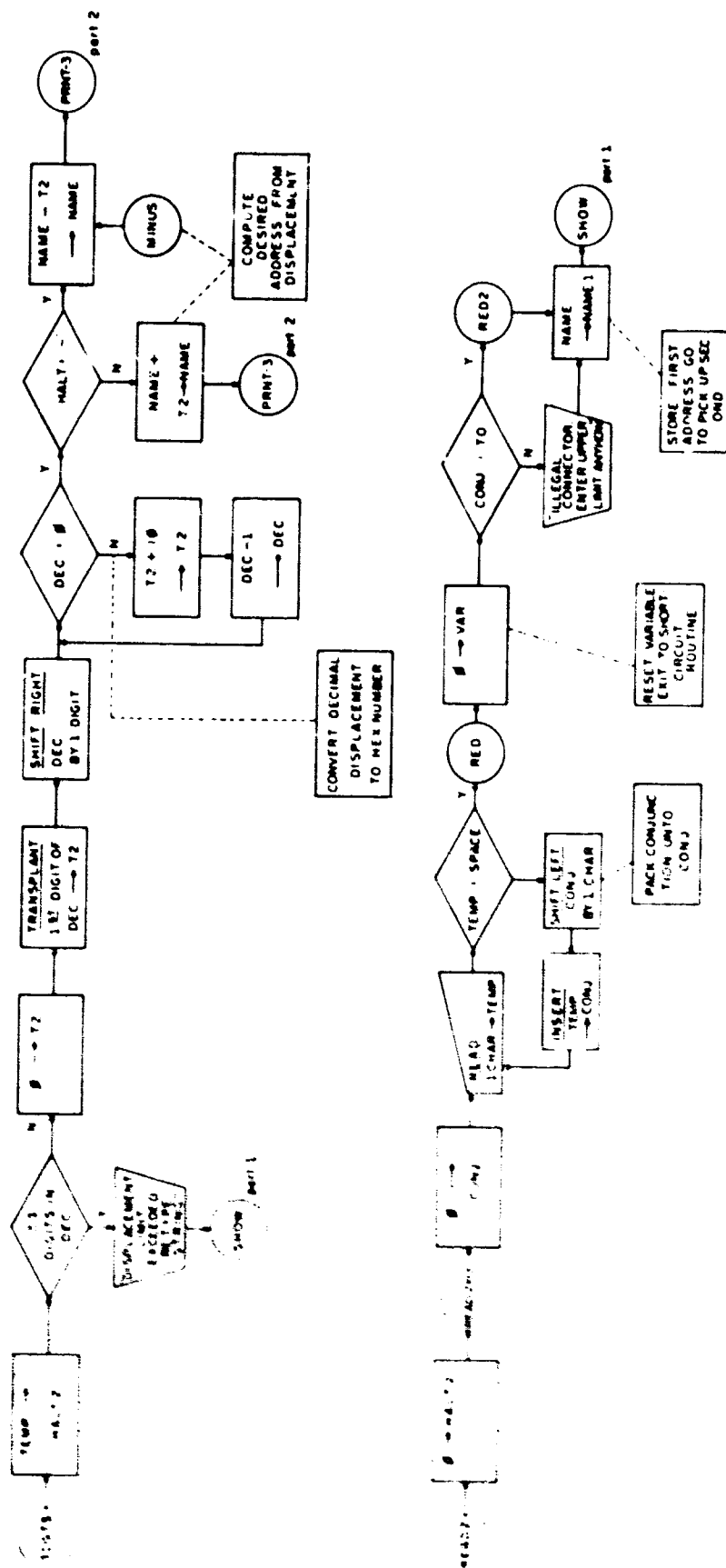


Figure 24. Flow-Chart for Subroutine to Examine Storage - Part 4

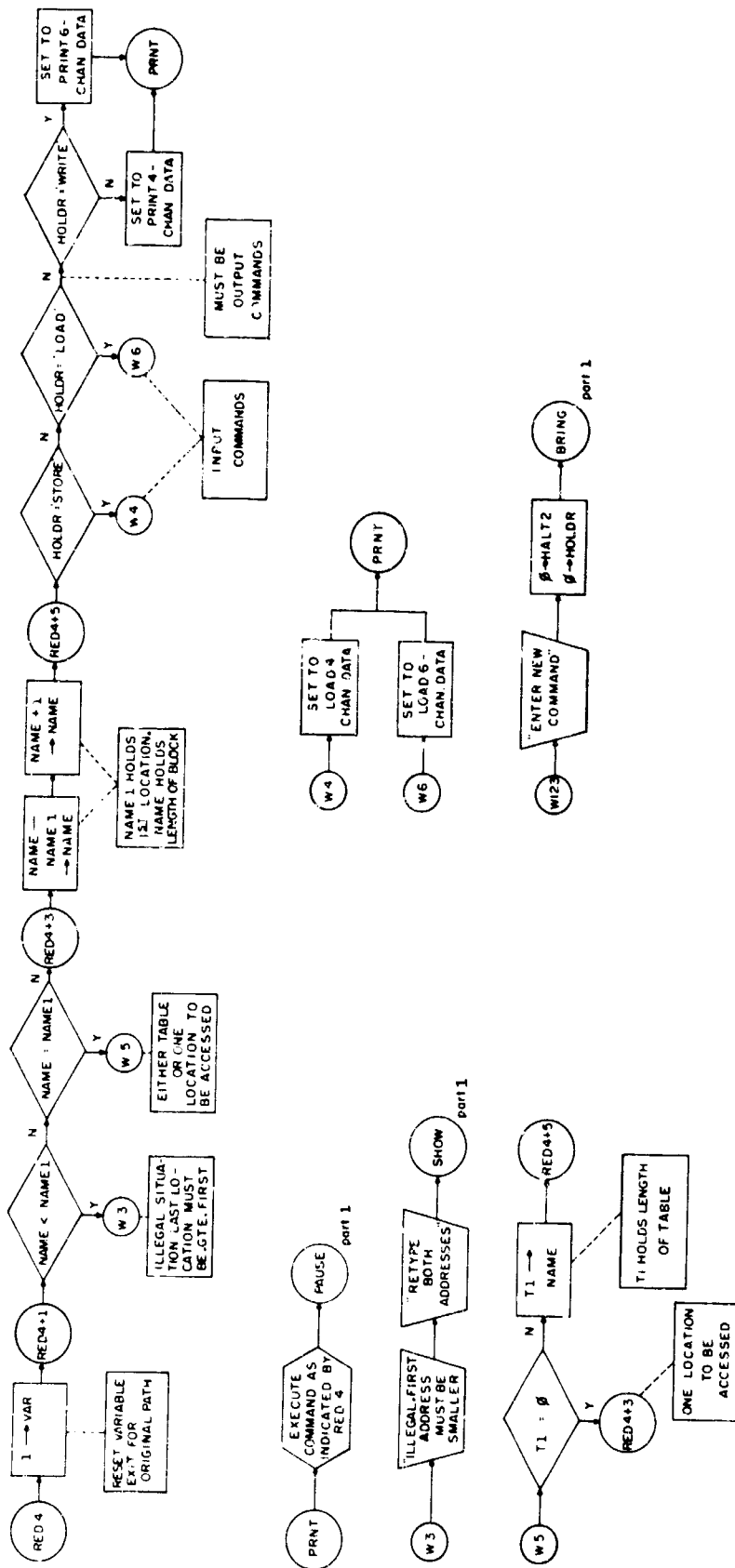


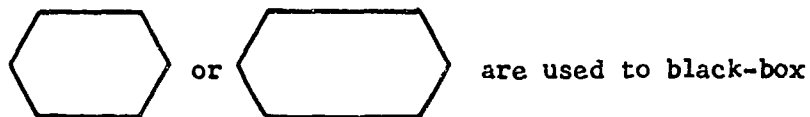
Figure 25. Flow-Chart for Subroutine to Examine Storage - Part 5

Notes for Figures 26 through 28:
Flow Chart for Pseudo-Dynamic Storage Allocation

GIMME - A Subroutine of WISCO (Adaptable to a general
Storage Allocator)

Common cell SIZE contains either number of words requested or first address of storage block being returned. In the first case, control is returned to calling point with table limits in SIZE. In the second, the existence of the table indicated is deleted from system records.

INUSE is a block of 20 locations used by GIMME to store pointers to the tables it has created. Copies of these pointers are returned via SIZE to the calling routine.



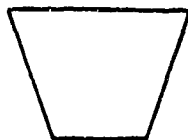
operations which are machine-dependent, knowledge of which being unessential to the understanding of GIMME's operation.



Boxes of the form

Which are intended to explain an operation in the context of the end result desired of the program.

Lower case subscripts u and e indicate the portion of a word in INUSE containing respectively the upper and lower limits of the corresponding tables. Upper case subscripts L and R designate the left and right half respectively of the words referred to. In general data will be stored right-justified (i.e., in the right half).



encloses a message to be printed out on-line.

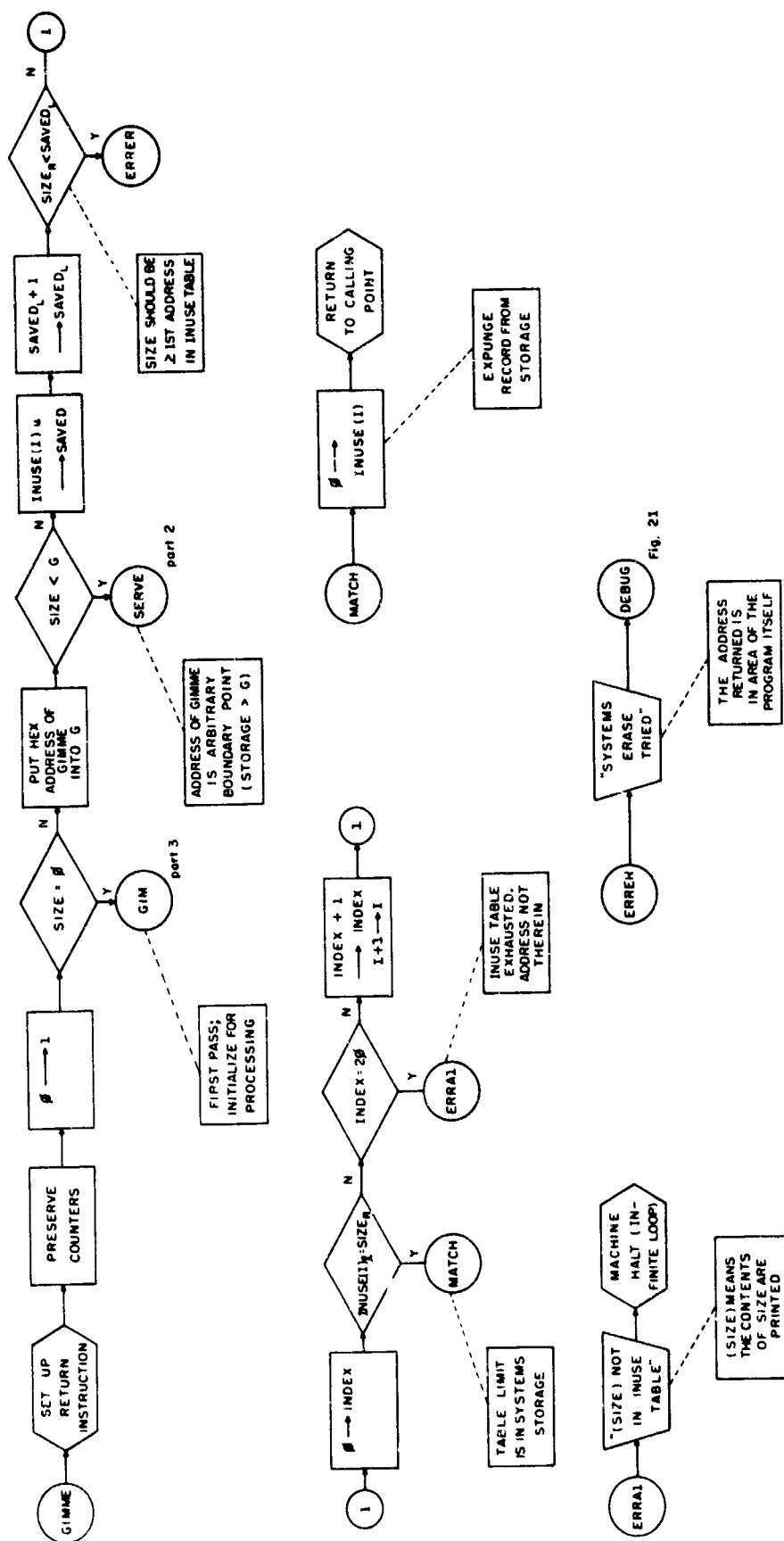


Figure 26. Flow-Chart for Pseudo - Dynamic Storage Allocation - Part 1

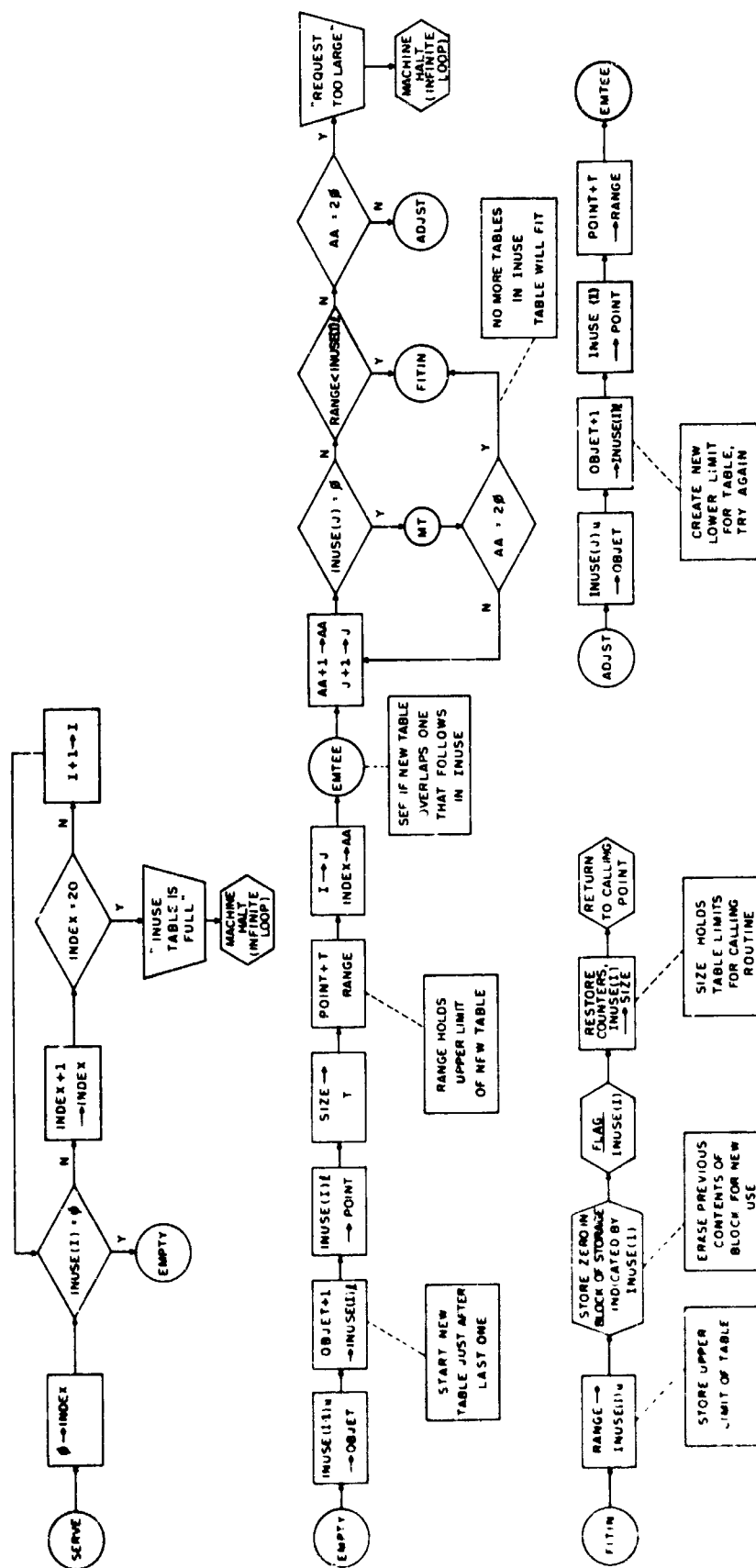


Figure 27. Flow-Chart for Pseudo - Dynamic Storage Allocation - Part 2

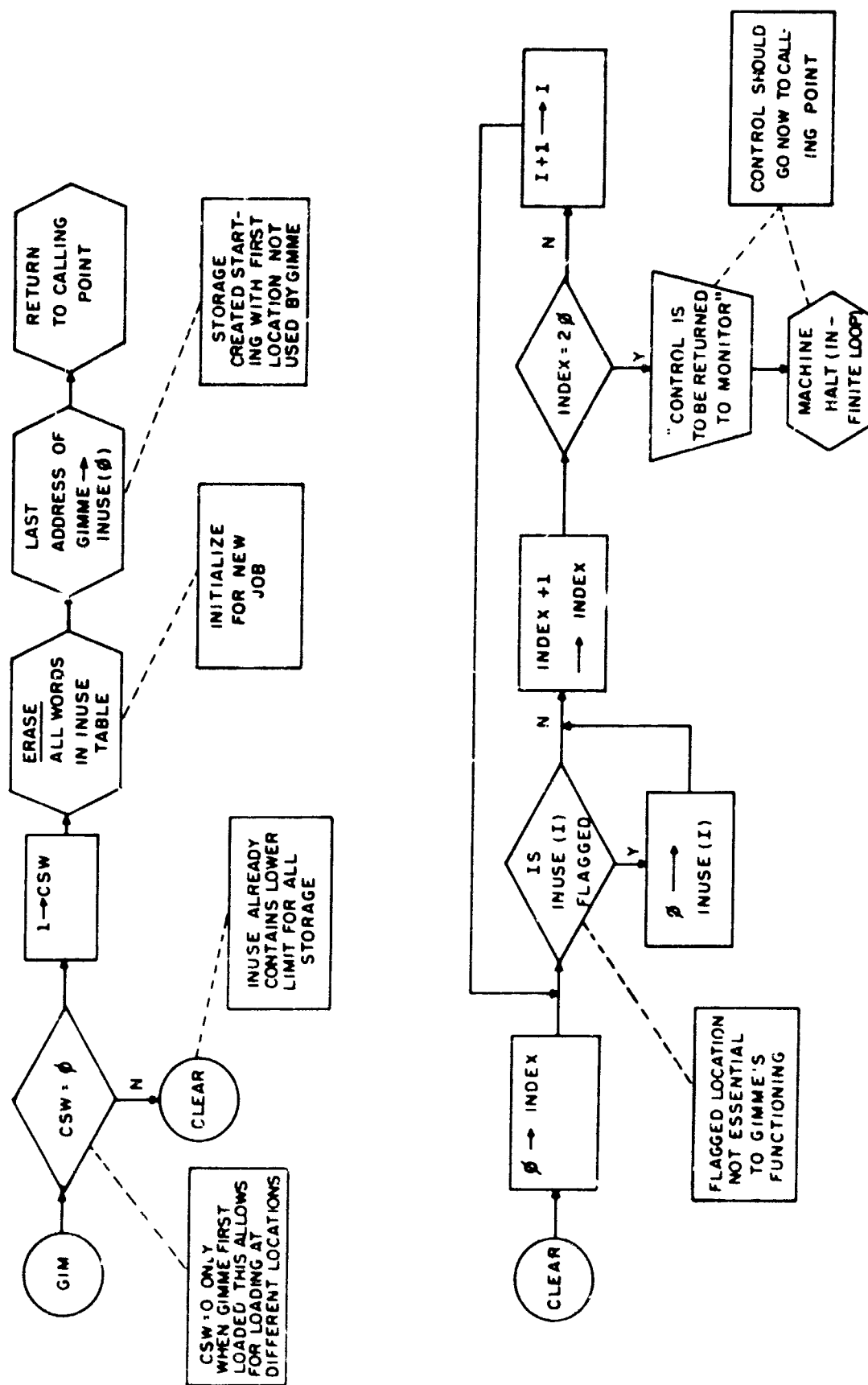


Figure 28. Flow-Chart for Pseudo - Dynamic Storage Allocation - Part 3

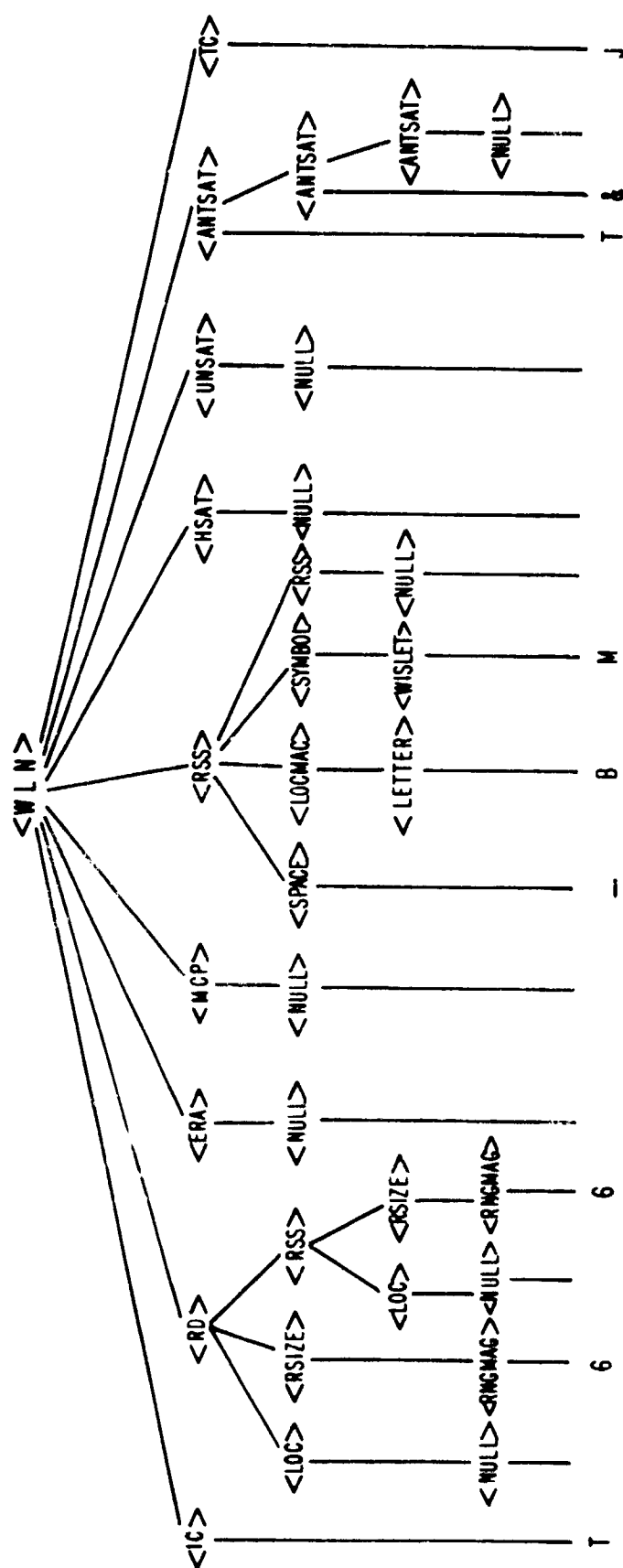


Figure 29. Parsing of the Wiswesser Notation T66-BMT&J

THE FORMAL ANALYSIS OF NOTATION SYSTEMS

James Munz
Department of Linguistics
University of Pennsylvania

Our orientation toward chemical notations* in general and "cipher systems" in particular is somewhat different than that of the principal users of notations. We are not directly concerned with economics of use of linear notations or with their ease of manipulation in indexing and searching. We view linear notations as languagelike structures exhibiting syntactic and semantic properties characterizable with the techniques of formal linguistics and mathematical logic. While there is no direct connection between the formal approach we take and the pragmatic approach that is evident among users, the two are strongly interconnected. It is only because there are regular semantic relations between the expressions of linear chemical notations and chemical structures that the notations are practical tools for storing and processing chemical information. The utility of linear notations as a mode of storing and processing large amounts of structural information depends on the existence of easily characterizable syntactic regularities reflecting semantic features in the codes. On the other hand, the practical demands of the chemist, who is the ultimate user and hence arbiter of notations even though his fluency in notations may not extend beyond a rudimentary knowledge of conventional nomenclature, forces notations to tolerate a reasonable amount of complexity in order to be able to provide chemically relevant structural information more easily and quickly.**

The structural formula is the basic means of representing structural information about chemical species. Even though the conventions under

* We will use this term following I. M. Hunsberger Survey of Chemical Notation Systems, NAS-NRC Publication 1150, pp 3-4, with the exception that while "notation" will be used indiscriminately for conventional and nonconventional systems, "code" will be reserved for nonconventional systems.

** This issue is rather ticklish. It has been pointed out in connection with information storage and retrieval systems (C. P. Bourne, G. D. Peterson, B. Lefkowitz, and D. Ford, Requirements, Criteria, and Measures of Performance of Information Storage and Retrieval Systems, Stanford Research Institute Project No. 3747, p8) that the user himself is frequently a poor source for direct comment on his needs, and usually cannot discriminate between his actual needs and his way of performing work. It seems reasonable to assume that this fact limits the chemist's judgement about notations.

which structural formulas are drawn are not codified or universally adhered to, structural formulas are reasonably uniform. They do not directly convey geometrical information about the chemical species they represent. In fact, the bulk of the information they directly convey is graph theoretic or can be represented in graph theory with a minimum of formal chicanery.* Since the most obvious and natural graph theoretic interpretation of structural formulas ignores the chemical identity of the nodes of the graph, it is convenient to augment this interpretation with information about the labelling of nodes; i.e., about the atomic species represented by a node. At least part of the information that can be conveyed in finite labelled multigraphs is what is described in all existing notations.**

A linear notation maps structural formulas onto linear expressions. It should be noted that nonlinear representations of structural information as long as they are discrete; e.g., various topological codes, matrix or connection table representations, can be trivially linearized. The labelled multigraphs depicted in structural formulas consist of sets of elements and sets of relations on those elements, and these can be trivially linearized by giving the set of elements or nodes in a graph with their labels and the pairs satisfying each relation. The rules that determine the mapping are the encoding rules for the notation. The encoding rules for a notation contain a subset of rules that specify certain substructures to be encoded. They also contain a subset of rules that specify how the distinguished substructures are to be encoded and finally a subset of rules that specify how fragmentary codes representing distinguished substructures are to be assembled. In some cases, the rules for distinguishing substructures do not exhaust the graph, e.g., fragmentation codes. In some cases, they partition the graph; e.g., the linear codes of Elsmann and Hinz. In some cases, they exhaust but do not partition the graph, e.g., IUPAC or Wiswesser codes. In some cases, the rules for encoding distinguished substructures are trivial. In some cases, the result of applying a rule depends on the environment of the substructure, e.g., rules 3.3 and 3.4 of the IUPAC manual† or rules 301

* As normally represented, information about stereoisomerism, optical activity, and absolute configuration is not strictly graph theoretic. But as long as the set of features distinguished is denumerable, it can be made graph theoretic by introducing new relations on the graph.

** Hunsberger, I. M. Survey of Chemical Notation Systems. NAS-NRC Publication 1150.

† Rules for I.U.P.A.C. Notation for Organic Compounds. Longmans, 1961.

and 10 of the WLN manual.* In some cases, the assembly rules depend on features of the graph; e.g., WLN manual rule 1 and rule 15 or rules 4.732 and 5.44 of the IUPAC manual. In some cases, they depend on features of the coded expressions; e.g., rule 2 of the WLN manual or rule 2.65 of the IUPAC manual. In some cases, they are completely arbitrary; e.g., a fragmentation code without rules for ordering fragments.

When coded expressions are intended to be generated manually the rules are often left implicit, stated imprecisely or stated in terms of fairly complex chemical concepts. Even in such cases an often surprising uniformity of encoding practice is observed. But such uniformity cannot be a product of the rules as stated, but rather of the uniformity of chemical practice and linguistic usage. This uniformity of encoding may also be a product of the user's restricted range of chemical interests. At any rate, implicit in an unformalized set of encoding rules is a family of sets of completely formal encoding rules some of whose member rule sets differ for as yet unencountered cases. Making a code completely formal consists initially in selecting one or more extensionally identical rule sets from this family.**

Now, if we say that any connected sequence of symbols in the vocabulary of a linear notation is a string in that notation, some of the strings thus defined will never appear in any string generated by the encoding rules. The class of strings in a code that is the image of the structures under the mapping should be decidable. This seems to be a minimal formal requirement of adequacy for a linear notation. If we restrict our attention to the class of these strings and their substrings, then, with respect to the encoding rules, we can define the notion of ambiguity. A string in the class is ambiguous if, and only if, more than one graphic structure maps onto this string. Further, if a string appears as a proper part of another string in the set and is ambiguous in this occurrence, it is locally ambiguous. If a string in this set is the image of an entire structure, such an occurrence is said to be autonomous and when the autonomous occurrence of the string is ambiguous, the occurrence is said to be properly ambiguous. An occurrence of a string that is not ambiguous is unambiguous. The autonomous occurrence of a string can be properly

* W. J. Wiswesser's Line-Formula Chemical Notation. E. G. Smith. 1966 Mimeographed.

** This involves first satisfying the intuition of the authors of the notation. Further selection is based on formal properties of the rule sets themselves and on convenience of a particular rule set for a predetermined application.

ambiguous by virtue of containing locally ambiguous occurrences of strings whose ambiguity is unresolved in the complete string. But, it can also be properly ambiguous without having locally ambiguous substrings. An unambiguous string can have local ambiguities so long, of course, as all but one of the ambiguous readings is eliminated in the total string.

In terms of the encoding rules, it is also possible to define the concept of uniqueness. A string is unique if, and only if, at least one structure maps onto it and every structure that is mapped onto the string by the encoding rules maps only onto this string. A notation is unambiguous if all its autonomous occurrences of strings are unambiguous and unique if all its autonomous occurrences of strings are unique.

Given that the encoding rules are effective, another minimal formal requirement for adequate codes; it may not be decidable whether or not a given string or an autonomous occurrence of a string is ambiguous, as we have defined it using the encoding rules. For it would be necessary to examine a denumerably infinite class of structures and at any point in the process the failure to have located two structures that mapped onto the same string would not constitute a proof that no two such structures existed. Uniqueness of a string with respect to a structure is decidable for given effective encoding rules, it is sufficient to encode the structure to determine conclusively whether or not the string is the only one representing the structure. A reasonable approach to avoiding the problem with ambiguity would be to "reverse" the encoding rules and find the image of a string in the code. However, if the coding rules map an indefinitely large class of structures onto a given string, such a procedure would not be effective. Effectiveness of decoding does not seem to be an absolute requirement for a minimally adequate code. In particular, fragmentation codes map infinite classes of structures onto a string (including the null string). If the ambiguity involved is always local and resolved to at least a finite number of structures for the autonomous occurrence of the string, then there is an alternative possible. (In practice, any significant amount of local ambiguity is sufficient to force the alternative approach.) Another set of rules can be constructed to recover the underlying structures.

These decoding rules consist of a subset of rules that locate particular substrings of autonomous strings, a subset of rules that specify what substructures are to correspond to the distinguished substrings and, finally, a subset of the rules that specify how the fragmentary substructures are connected to one another. In the case of fragmentation codes or empirical formulas, "decoding rules" could be constructed for each, but in the former case they would either not be effective or would not produce the inverse image of the encoding rules, and in the latter case, while the number

of structures satisfying any empirical formula are finite, it is not practically feasible to decode. Hence, with codes of this type, there is no interest in decoding. The same comments can be made with respect to informally stated decoding rules and their formalized counterparts as were made for informal and formal encoding rules.

Observed uniformity in decoding is not strictly a product of the informal rules themselves. And corresponding to the informal rules is a family of sets of formal rules. Selecting one or more extensionally identical rule sets from this family is another part of making a code completely formal. But another factor enters in the case of decoding rules. The motivation for distinct encoding and decoding rules is that either the encoding rules are not reversible or it is not practical to reverse them. Thus, the encoding rules have a certain priority, and the decoding rules must actually produce the inverse image of the encoding rules. This is an additional formal requirement for an adequate code with both encoding and decoding rules. For a large class of codes, it is required that the decoding rules be effective.

For a code with decoding rules, it is possible to give alternative definitions of ambiguity and uniqueness provided that the decoding rules meet the last requirement. A string is ambiguous if, and only if, it is mapped onto more than one structure by the decoding rules. And a string is unique if, and only if, it maps onto a set of structures and no other string maps onto a member of the set. Now, if the decoding rules are effective, then ambiguity is directly decidable under this definition. Uniqueness is not decidable when so defined. Uniqueness and ambiguity are duals with respect to decoding and encoding. It is also worth noting that if the encoding and decoding rules are inverses, then the alternate definitions of ambiguity and uniqueness are equivalent.

To be practically valuable, a notation need not be formally adequate in our sense. In fact, it is not definitely known whether any of the existing codes that are intended to be unique, unambiguous, and complete even meet the minimum formal requirements. Given a pressing practical demand for such a code, the formal properties of a candidate are relatively unimportant if it will breach the gap. In such a situation, empirical and statistical tests of adequacy are very valuable; however, the greater the investment in implementing such a code and the wider the class of structures to be dealt with and the less tolerable possible failure of a retrieval system, the more practically important a formalized code and the formal proofs of adequacy, which the formalized code makes possible, become.

Beyond such considerations, it is mandatory for the encoding and decoding rules to be formulated if mechanical encoding and decoding are envisioned for a particular application. Mechanical decoding has been achieved, or will shortly be achieved, for the Wiswesser line notation at least to the extent of producing connection tables or other formal equivalents of structural formulas for a large class of structures. The rapid progress of computer technology makes it seem very likely that in the near future it will be feasible to input structural formulas directly and economically. It would seem appropriate for the developers of the Wiswesser line notation to pave the way for mechanical encoding by giving serious attention to formalizing the encoding rules.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) CO, Edgewood Arsenal ATTN: SMUEA-TSTD Edgewood Arsenal, Maryland 21010		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
		2b. GROUP N/A
3. REPORT TITLE PROCEEDINGS OF THE WISWESSER LINE NOTATION MEETING OF THE ARMY CHEMICAL INFORMATION AND DATA SYSTEMS PROGRAM 6-7 OCTOBER 1966		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Meeting held on the 6th and 7th of October 1966.		
5. AUTHOR(S) (First name, middle initial, last name) Mitchell, James P., ed.		
6. REPORT DATE January 1968	7a. TOTAL NO. OF PAGES 212	7b. NO. OF REFS 43
8a. CONTRACT OR GRANT NO. A. PROJECT NO. 2P020401A727 C. d.		8b. ORIGINATOR'S REPORT NUMBER(S) EASP 400-8 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) N/A
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES Chemical information and data system	12. SPONSORING MILITARY ACTIVITY N/A	
13. ABSTRACT A significant application of Wiswesser Line Notation (WLN) is to enable computer discrimination among chemical compounds based on their characteristics as represented by symbolic code designations. These proceedings cover use and techniques of WLN by various industrial, military, and academic organizations. Areas of use discussed include registration of compounds, storage, and retrieval of structures in several collections, and generation of structural fragment codes for rapid structure and substructure searches. Other papers cover quick scan and symbols, a computer-generated open ended fragment code, permutations and classification numbers, the "Dot-Plot" computer program, and a partial algorithm for development of connection tables from WLN. A presentation on the formal analysis of notation systems is also made. Discussions in techniques include file maintenance and updating procedures, and machine management of small chemical data systems.		

DD FORM 1473
1 NOV 66

REPLACES DD FORM 1473, 1 JAN 66, WHICH IS
OBSOLETE FOR ARMY USE.

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Wiswesser Line Notation						
Low-cost storage						
Permuted line notation						
Registering compounds						
File maintenance procedure						
Updating procedure						
Quick scan						
Structural fragment codes						
Structure searches						
Permutations						
"Dot-Plot" computer program						
Notation systems						
Chemical structures						
Retrieval system						
Topological codes						
Automatic data processing						
Time-cost data						
Symbols						
Molecular formula codes						
Open ended fragment code						
Man-machine system						
QUICK						
CIDS						
Chemical information and data system						
SUBMAP						
QUEEK						
Quick-see search						
BATCH number						
Chemical structure notations						
Correction tables						
Contractions						
Ring system						
Syntax analysis						
Transformation algorithm						