AZOTNONATRIEVA4 SOL6 sodium nitrate
SOL6 ZAKISI/OKISI JELEZA ferrous/ferric salt
ZAKISNA4 OKISNA4 SOL6 JELEZA
GIDRAT ZAKISI/OKISI JELEZA ferrous/ferric salt

etc., where the stems underlined may be replaced by any of a number of other stems (up to 65 in some positions) in the particular type.

Translation of each type encounters problems common to almost all the types: (1) The Russian noun is translated as an English adjective, while the noun of the resulting English phrase is found among the modifiers of the Russian noun. (2) The Russian noun (English adjective) may be a metal with more than one valence state, the state indicated (if at all) by the modifiers. (3) The number of the resulting English noun-phrase is determined by some member of the Russian phrase other than the noun. (4) The phrase elements may occur compounded in the chemical phrase but free in other contexts, and dictionary storage must provide for this. The program permits translation of conjoined phrase elements as well.

The paper also includes an investigation into the deeper grammatical implications of this type of chemical nomenclature, and some excursions into the semantic correlations involved.

The Application of Table Processing Concepts to the Sakai Translation Technique

A. Opler, R. Silverstone, Y. Saleh, M. Hildebran, and I. Slutzky

*Computer Usage Company**

In 1961, I. Sakai described a new technique for the mechanical translation of languages. The method utilizes large tables which contain the syntactic rules of the source and target languages.

As part of a study of the AN/GSQ-16 Lexical Processing Machine, a modification of the Sakai method was developed. Five of six planned table scanning phases were implemented and tested. Our translation system (1) converts input text to syntactic and semantic codes with a dictionary scan, (2) clears syntactic ambiguities where resolution by adjacent words is effective, (3) resolves residual syntactic ambiguities by determining the longest meaningful semantic unit, (4) reorders word sequence according to the rules of the target language and (5) produces the final target language translation.

French to English was the source-target pair selected for the study. An Input Dictionary of 3,000 French stems was prepared and 17,000 entries comprised the Input Product Table (allowable syntactic combinations).

Since Sakai was working with highly dissimilar languages, he found it necessary to use an intermediate language. Because of the structural similarity between French and English, we found an intermediate language was unnecessary.

The method proved straightforward to implement using the table lookup logic of the Lexical Processor. The translation was actually performed on an IBM 1401 which we programmed to simulate the concept of the AN/GSQ-16 Lexical Processor. In our implementation magnetic tapes replaced the photoscopic storage disk.

* This work was performed while under contract to IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

Slavic Languages—Comparative Morphosyntactic Research

Milos Pacak

Machine Translation Research Project, Georgetown University

An appropriate goal for present-day linguistics is the development of a general theory of relations between languages. One necessary requirement in the development of such a theory is the identification and classification of inflected forms in terms of their morphosyntactic properties in a set of presumably related languages.

According to Sapir, "all languages differ from one another, but certain ones differ far more than others". As for the Slavic languages he might well have said that they are all alike, but some are more alike than others. The similarities stemming from their common origin and from subsequent parallel development enable us to group them into a number of more or less homogeneous types.

The experimental comparative research at The Georgetown University was focused on a group of four Slavic languages, namely, Russian, Czech, Polish and Serbocroatian.

The first step in the comparative procedure here described is the morphosyntactic analysis of each of the four languages individually. The analysis should be based on the complementary distribution of inflectional morphemes. The properties whose distribution must be determined are:

1) the graphemic shape of the inflectional morphemes,
2) the establishment of distributional classes and subclasses of stem morphemes and (on the basis of 1 and 2),
3) the morphosyntactic function of inflectional morphemes which is determined by the distributional subclass of the stem morpheme.

f(x,y)-l, where x is the distributional subclass of the stem morpheme (which is a constant) and y is the given inflectional morpheme (which is a free variable). On the basis of this preliminary analysis the patterns of absolute equivalence, partial equivalence, and absolute difference can be established for each class of inflected forms in each language under study.

Once this has been accomplished, the results can be used in order to determine the extent of distributional equivalences among the individual languages. The applicability of this procedure was tested on the class of adjectivals. Within the frame of adjectivals the follow-

ing morphosyntactic properties were analyzed within each language first and compared among the four languages:

1) the category of gender,
2) the category of animateness,
3) the category of case and number.

The product of this comparative analysis is a set of formation rules which embody a system for the identification of the inflected forms. The detailed result will be presented in an additional report.

Types of Language Hierarchy

E. D. Pendergraft

Linguistics Research Center, The University of Texas

Various relations lead to hierarchical systems of linguistic description. This paper considers briefly a typology of descriptive metalanguages based on such relations and sketches possible consequences for computational linguistics.

Its scope is accordingly limited to metalanguages having operational interpretations which specify individual linguistic processes and structural interpretations which specify language data of individual languages. Immediate-constituent, context-free metalanguages are used to illustrate hierarchical types.

Path Economization in Exhaustive Left-to-right Syntactic Analysis

Warren J. Plath

Computation Laboratory, Harvard University

In exhaustive left-to-right syntactic analysis using the predictive approach, each path of syntactic connection which originates at the beginning of a sentence must be followed until it is clear whether or not it will lead to the production of a well-formed analysis. The original scheme of following each path until it terminates either in an analysis or in a grammatical inconsistency has been considerably improved through the. incorporation of two path-testing techniques. Using the first technique, the program abandons a path as unproductive whenever a situation is detected where the prediction pool contains more predictions of a given type than can possibly be fulfilled by the remaining words in the sentence. Employment of the second technique, which is based on periodic comparison of the current prediction pool with pools formed on earlier productive paths, eliminates repeated analysis of identical right-hand segments which belong to distinct paths.

Taken together, the two path-testing procedures frequently enable the program to terminate the processing of a path well before its end has been reached. For most sentences, this means a considerable reduction in the total path length traversed, accompanied by a corresponding increase in the speed of analysis. Comparison of runs performed using both versions of the program indicates that employment of the new techniques reduces the average running time per sentence to less than one-fifth of its former value.

A Computer Representation for Semantic Information

Bertram Raphael

Computation Center, Massachusetts Institute of Technology

This paper deals with the problem of representing in a useful form, within a digital computer, the information content of statements in natural language. The model proposed consists of words and list-structure associations between words. Statements in simple English are thought of as describing relations between objects in the real world. Sentences are analyzed by matching them against members of a list of formats, each of which determines a unique relation. These relations are stored on description-lists associated with those words which denote objects (or sets of objects). A LISP computer program uses this model in the context of a simple question-answering system. Functions are provided which may grow, search, and modify this model. Formats and functions dealing with set-relations, part-whole and numeric relations, and left-to-right spatial relations have been included in the system, which is being expanded to handle other types of relations. All functions which operate on the model report information concerning their actions to the programmer, so that the applicability and limitations of this kind of model may more easily be evaluated.

Specifications for Generative Grammars Used in Language Data Processing

Robert Tabory

IBM Thomas]. Watson Research Center

It becomes more and more evident that successful pragmatics (i.e. automatic recognition and production procedures for sentences) cannot be performed without previously written generative grammars for the languages involved, using an underlying meta-theoretical framework proposed by the present school of mathematical linguistics. Two aspects of grammar writing are examined:

1. A taxonomy over the non-terminal vocabulary, using a subscripting system for signs and fitting into the more general string taxonomy of phrase structure components. The resulting more complex lexical organization is studied.

2. A command syntax for phrase structure components limiting the full, not necessarily needed generative power of these grammars. The proposed restrictions correspond to a priori linguistic intuition. Applicational order and location of the rules is studied.

Finally, the recognitional power and generative capacity of a computer are examined, the machine being structured according to a Newell-Shaw-Simon list system. It is well known that pushdown stores are particular cases of list structures, that context-free grammars