Computer Interpretation of English Text and Picture Patterns

RUSSELL A. KIRSCH, MEMBER, IEEE

Summary—This paper considers a class of information sources consisting of text and pictures. The text is English language text appearing in scientific and technical documents. The picture sources are the largely schematic pictures that occur in the same class of documents. However, the discussion is broadened slightly to include other picture sources. For a tiny fragment of English, the paper shows how the syntactic structure of text may be described, and then goes on to suggest that a similar analysis may be performed on the class of pictures under study. The description of these two kinds of information sources with a single class of descriptive techniques is suggested as an alternative to the synthetic approach in which artificial languages are specified and then learned and used. The major reason for doing syntactical analysis of such sources discussed here is that several information processing operations, amounting to the interpretation of the information sources looked upon as languages, can be done by the technique of syntax direction which uses the results of syntactic analysis to mediate subsequent processes for manipulating the information tokens. The paper concludes with an illustration of an algorithm for matching the sentences given by a simple grammar against the class of simple pictures which these sentences purport to describe.

I. A Class of Information Sources

IN THIS PAPER we shall consider as a single class of information sources that which is usually considered to be two or perhaps more disjoint classes of information for information processing purposes. To exemplify this class of information sources, the information content of formal scientific and technical documents may be considered, even though in the present presystematic state of affairs, we have no means of quantitatively and precisely speaking of the information content of documents (except in the communication theoretic sense which shall not be of concern here). It is, nevertheless, intuitively plausible that two kinds of information sources be recognized, *viz.*, the textual matter of the documents (this paper shall be restricted to a consideration only of English text) and the pictorial matter. The text includes, but is not restricted to, sequences of English sentences. Text is also understood here to include the quasi-textual matter contained in the format of documents and also contained in captions, labels, and the like.

The pictorial information includes, but again is not restricted to, the stylized schematic information that occurs in diagrams and drawings. With the pictorial matter also it is desirable to include information of a format nature where the information bearing function is served by the juxtaposition of information symbols in two dimensions rather than by linear concatenation as in text.

A degree of overlap is seen here between text and pictures that suggests a first reason for not disjoining these two sources but instead attempting to treat them by uniform methods. It is certainly clear that most pictorial matter in formal scientific documents contains text within it, but if the converse (that text is also pictorial in nature) is not clear to the reader, he need only consider the fairly elaborate instructions necessary to enable several people key-punching nominally clear text to observe consistent conventions [1].

One reason such key-punching instructions must necessarily be fairly elaborate is that the key puncher must translate from a basically pictorial two dimensional source into the sort of linear string representation which characterizes text input for most computer processing purposes. One is thus led to the view that text as it occurs in technical documents is a special case of a pictorial source, admittedly with some very special properties derived from its parent, spoken language.

The inseparability of text and pictures is indicated still further by the fact that they perform much the same functions within documents Take the example of a document containing an electrical circuit diagram and some text which describes it. Questions may be asked and answered either on the basis of the diagrams or the text or both. One may be a paraphrase for the other. Indeed, in such documents as U. S. patents, the paraphrase is a most thoroughgoing one with the circuit diagrams that occur largely reproduced in the form of textual descriptions.

Now, although it might be possible to marshal many arguments to demonstrate the inseparability of natural language text and pictures (at least in technical documents), no important consequences would follow from this observation were it not for the significant fact that these two kinds of information sources can probably be handled with very similar techniques. From the standpoint of computer information processing, the important fact about natural language text and pictures is that they both have a syntactic structure which is capable of being described to a machine and of being used for purposes of interpreting the information within a data processing system. The problem of how to describe the syntactic structure of text and pictures and

Manuscript received March 13, 1964. The work in this paper was supported in part by grant GN-107 from the National Science Foundation.

The author is with the National Bureau of Standards, Washington, D. C.

how to use this syntactic description in interpreting the text and pictures will be the main concern of this paper.

In studying the syntax of these languages, we shall be concerned with identifying in the language the primitive symbols whose internal structure need not be considered. We shall then be mostly interested in the rules of arrangement of these primitive symbols that allow certain arrangements and disallow others, and that allow new language objects to be constructed from previously constructed ones.

We will be concerned with the semantics of these languages to the extent that we will consider how the syntactic descriptions of language objects may be interpreted. A syntactical system is interpreted when it is used to denote or refer to objects outside of itself [2]. We shall, however, speak of interpretation in a somewhat wider sense whenever we invoke the syntactic description of arrays of words and other symbols to perform manipulations on the arrays.

II. The Approach of Syntactical Description

In order to see how the syntax of a language may be exhibited, let us consider as examples a traditional grammar book which describes the purported grammar of English, the syntactical specification of a programming language like ALGOL, and the description given by a logician of the rules of formation for a logical language. The grammar books attempts to record the results of an empirical investigation into an existing spoken language. That the grammar book fails to describe the language as it is spoken (or even formally written), results from the normative approach usually taken in grammar books to the description of "correct" usage. The traditional grammar book fails to be descriptive and substitutes normative, prescribed usage in its place. In the case of ALGOL, prescription and usage must agree. If one ignores the possibility of minor errors in the construction of ALGOL algorithms on the part of users of the language, it is found that the formal definition of the language coincides very closely with the language as it is used. We say closely rather than completely because the ALGOL specifications do contain some informal comments regarding statements in the language. It is by no means difficult to observe the informal restrictions thus specified notwithstanding the fact that the restrictions are to an extent outside of the formal specification of the language [3]. In the third example, the language that the logician constructs is usually a comparatively simple language whose syntax is completely and precisely specified.

In making the above comparison between traditional grammar-books on the one hand and descriptions of formal languages on the other, our purpose has been to prepare for the suggestion that it would not be out of place to demand of descriptions of natural languages such as English, those very qualities of exhaustiveness and precision that we demand of descriptions of artificial languages such as ALGOL [4].

To describe and formalize a previously unformalized natural language may seem like an unnecessarily difficult task, particularly when there is the possibility of creating a special purpose artificial language and formalizing its description instead. If our task is, for example, to provide a computer with a programming language in which algebraic manipulations may be described, then an admittedly unattractive candidate for this language is a formalization of natural English. The demonstrated success of such artificial languages as ALGOL argues against the use of natural languages in such specialized contexts.

There are other contexts, however, in which the use of an artificial language is precluded for other reasons and where, if communication with a machine is at all to be possible by automatic techniques, the descriptive approach to an already existing language must necessarily be taken. Wherever there are archives of records and documents primarily intended for human use and where the cost of manual interpretation and preprocessing for machine purposes is prohibitive, there we find a language source that requires a descriptive approach to the archives. The world's technical literature is, and very likely will continue to be, a case in point insofar as the possibility of using anything other than natural language for communication between scientists seems remote. If such literature is also to be manipulated by machine, one is thus led to investigate the possibility of providing a precise description of a natural language to a machine. If that should prove possible, one would then wish to explore the possibility of providing such a machine with procedures for interpreting the language in terms of the functions which the language is capable of performing.

Let us first try to decide what it is about English language text that can be described to a machine. Whatever description we give to a machine, it must agree with the observed fact that English text consists of certain units which for the moment shall be undefined, and a set of allowable arrangements of these units. Not all units can occur in all places in text. From a formal analysis of English text, we would expect to get a formally-defined unit in terms of arrangements of which sentences may best be described. However, in this informal discussion, no harm will be done by resorting to an informally-defined unit, and one, moreover, which has little to do with whatever formal units a complete analysis might define. Our informally-defined unit will be called the "word," and it shall correspond to the native speaker's (reader's) informal notion. Using this intuitive unit, then, one may consider the "word" to be the basic unit of English text. Having done so, we will take the syntactical investigation of the properties of English text to be an investigation into the allowable arrangements in text of English "words." The weakest acceptable goal of such an investigation would be to produce an algorithm which, when given any sequence of English "words," would determine whether or not that sequence was an

English "sentence," where, again, we accept the judgments of English-speakers as specifying what "sentences" are. This algorithm would then provide a decision procedure for sentencehood applied to sequences of English words.

There is evidence, however, to give hope that a more difficult and stronger goal is both achievable and desirable. This goal of syntactic analysis would be not only to provide a decision procedure for sentencehood, but, in addition, to so classify words and sequences of English words that those which are intersubstitutable are similarly classified. Those sequences of English words which are intersubstitutable would be said to belong to the same syntactic category. Syntactic analysis would then assign words to syntactic categories and sequences of syntactic categories to other, not necessarily different, syntactic categories in turn, in such a way that for any sequence, if it is a sentence of the language, there will be a syntactic analysis assigned to certain subsequences in it and an analysis to the whole sequence which assigns it to the category of sentence. What we have outlined here is the procedure of immediate constituent analysis [5].

English text has syntactic structure notwithstanding several important problems which have been largely glossed over here. To mention them briefly, there is the question of the choice of a suitable analysis technique, immediate constituent analysis being only one. The identification of the primitive units at any one level of analysis is itself a difficult problem that we have alluded to. So is the problem of delimiting language. Many sentences are questionable grammatically, and assigning them degrees of grammaticalness is a difficult job. The difficulty of these and related problems may be taken as a partial explanation for the fact that there does not exist any complete syntactic analysis of English text. For certain very narrowly delimited segments of the language, complete syntactical descriptions or grammars exist. So do broad sketches of what a grammar for the whole of the language might look like [6].

But it is suggested above that the same arguments with suitable changes can also be applied to the analysis of pictorial images such as those which occur in technical documents. The case here is somewhat more difficult to make because there does not exist any tradition in analyzing pictorial images. By comparison, the tradition of linguistic analysis is highly developed with its modern descriptive version for the analysis of spoken utterances, at least, and also to some extent, of printed text.

In order to establish the thesis that pictures of certain kinds have a syntactic structure capable of being described to a machine, let us subclassify pictorial sources into three kinds. A prototype of the first kind will be the diagrams and schematics that occur in documents; examples are electrical circuit diagrams, chemical structure diagrams, and flow charts. The second class of pictorial sources will consist of synthesized images which purport to be representational and thus include sketches and drawings, especially mechanical drawings. The last class of pictorial sources will be photographs of natural objects, but this class goes beyond the scope of this paper.

In the first class of pictorial sources, the diagrams and schematics, it is fairly clear that the set of primitive symbols should be chosen from among that set for which we have character-recognition techniques currently in existence [7]. Thus in an electrical circuit diagram, it is quite reasonable to locate and identify capacitor, resistor, and inductance symbols and the many other symbols that occur, as well as the English letters which label them, by the use of character-recognition techniques of the kind that have already been developed. Let us call these symbols so recognizable the "characters" in the schematic sources. We immediately recognize, however, that the identification of the specific characters in a circuit diagram falls far short of the identification of all the significant information in the diagram. Of considerable importance, even more so than the recognition of the characters themselves, is the recognition of the way in which they are connected by lines and related to each other by juxtaposition. This interconnection of the characters is the syntactical structure of the diagram. and one would expect that the information content of a diagrammatic source would be conveyed as much by the syntactic structure as by the characters within the diagram. It is fairly evident, furtheremore, that the syntactic structure does exist and has the usual properties of a syntactically structured language, *viz.*, that there are allowable as well as disallowed juxtapositions of the primitive symbols.

For the second class of picture sources, the drawings and representations, the existence of a syntactic structuring is not quite so evident as it is for schematics. The question has recently been studied, perhaps not surprisingly, by an art historian, Gombrich [8]. He is concerned with graphic art in its many forms, in particular, drawing and sketching. What is important for our purposes is the conclusion that he draws: "Everything points to the conclusion that the phrase 'the language of art' is more than a loose metaphor, that even to describe the visible world in images we need a developed system of schemata." Gombrich's schemata are what we refer to as rules of syntax, and the point that he develops through his whole study is that the syntactic structure of graphic art, in particular representational art (thereby including drawings and sketches), is determined largely by a set of syntactic rules.

There is a common argument against the possibility of producing a syntactical description of pictorial sources or of English text, for that matter. The argument is to the effect that such languages are essentially complex. They have no simple syntactical description except one that only very loosely approximates the languages as they occur in practice. If these languages are exceedingly complex (and there is every reason to believe that they are) the argument concludes that it is unlikely that people will be able to understand the structure of such languages or be able to understand their syntactical description except as users of the languages. However, enough is known about how to write extremely large computer programs that the mere fact of complexity of a language being described should not in itself preclude any attempts to describe it to a machine because it may have a simple syntactical description when the correct tools are used in describing it. Even if its syntactical characterization is complex in the sense of being very large, that still does not preclude the possibility that the language can be so segmented that each of the segments has a sufficiently simple syntactic description to prove tractable. Some light is shed on syntactical description in this fashion by the study of the so-called sequential languages [9]. These languages have the useful property that their syntactic description can be built up in an hierarchical fashion from the primitives which compose the language. From these considerations we are led to the conclusion that complexity in a language and in its syntactical description need not rule out the possibility of providing such a description to a computer for subsequent interpretation, at least in limited areas of application.

III. Techniques for Syntactical Description

It has been argued thus far that information sources occurring in technical documents can profitably be looked upon as languages whose syntaxes are amenable to precise and exhaustive description of a computer. Of course it is one thing to recognize the existence of a language and to infer the existence of its grammar, and quite another matter to exhibit the grammar. It is, therefore, important that we investigate the techniques that are available for doing syntactic description of the textual and pictorial languages existing in documents in order to get an indication of how likely it is that these techniques will prove sufficient for describing the languages with which we are concerned.

Bobrow has surveyed the existing computer programs for syntactic analysis of English printed text [10]. Our purpose here will be to consider one or two such techniques in somewhat greater detail. In the next section, it will be shown how a particular technique, that of the use of context-free grammars, permits an effective interpretation of the languages so described.

If we compare different grammars for the same language as, for example grammars of English text, we find a substantial disagreement as to the kind of analysis assigned by the grammars to the text. Even where the agreement exists on what the corpus shall be, the analyses assigned to that corpus very often differ. This difference in analyses may be attributed to the difference in the linguistic intuitions of the linguists who write the grammars. It may also be attributed to differences in the dialects being accounted for. But the most important reason for differences is that the purposes of performing a syntactical analysis might differ from one case to the next. We can mention several different purposes served by a syntactic analysis of a language for any pair of which the syntactic analyses necessary to accomplish the purposes are likely to differ. For example, in performing mechanical translation in attempting to do auto extracting [11], in attempting automatically to paraphrase [12], in attempting to do automatic indexing and finally, in attempting to do automatic question answering [13], all from natural language text, it is likely that the syntactic analysis necessary to achieve the purpose will be quite specialized.

There are a few ways in which a syntactic description (of an underlying language which is to be interpreted for machine processing purposes) may be embedded in a larger system. One kind of syntactic description is implicit in the part of the system that does syntactic analysis. Thus, in the Kuno and Oettinger [14] syntactic analyzer, an underlying descriptive theory of the language is implicit in the analyzer and can indeed be resurrected and put in explicit form, although this is unnecessary for the intended purposes of syntactic analysis. Or in the discourse generator of Simmons, Klein, and McConlogue [15], the dependency analysis and their so-called transitivity rule have implicit in them a syntactic description of the underlying language.

In the applications that are of interest here, what is being described in a descriptive language is itself a very large natural language whose description necessarily is an open-ended task. The empirical problem then is to determine what descriptive languages can be used in describing natural languages like English text or schematic pictures. There are several different types of descriptive languages which serve as candidates, and we wish briefly to compare some properties of these different kinds of descriptive languages within which syntactic theories of, *e.g.*, English text can be formulated.

The first possible descriptive language is a natural language, perhaps the natural language that is itself being described. We must reject this out of hand because of the unformalized state of the natural language. Thus, to describe English syntax in English is, in the present state of the art, useless to a machine. Paradoxically though, one result of achieving a syntactic description of English might be to permit a suitable formalization of a sufficiently large fragment of English that the product might be used for describing English itself. This initial bootstrapping step has yet to be taken.

When we come to the formalized programming languages as candidates for use in describing the structure of English text, we run into a paradoxical difficulty that the most powerful languages are the least useful. To explore this difficulty, let us consider the restricted task of describing the structure of English text in some suitable programming language by writing a program which is capable of systematically exhibiting all and only the English text that has occurred in some corpus plus a linguist's extrapolation thereof. If our task is thus restricted, the ideal candidate would, of course, be such a programming language as is convenient to use, easy to learn, and for which efficient object language can be compiled, etc. But, as a matter of fact, the task of producing a description of a natural language is not so restricted as we have assumed. To see this, let us assume that we had written a program P_1 which, when provided with a list of all English words [16], could in some systematic sense list a set of all and only the English sentences $\{S\}$ occurring in some corpus and in a reasonable infinite extrapolation therefrom [34]. To actually run P_1 and allow it to produce $\{S\}$ serves no useful purpose. The real purpose of writing P_1 is to use it as one of the inputs to another program P_2 which is capable of analyzing the sentences in $\{S\}$. The way the P_2 would operate is that it would accept as input S_i, a sequence of words, and also P_1. Then P_2 would furnish as output $P_1(S_i)$, the trace of P_1 in one possible process of its producing S_i, or an indication if no such process exists. Thus, P_1, in conjunction with the analyzing program P_2, produces the set of analyses $\{P_1(S)\}$ for the language $\{S\}$. But what happens if P_1 is written in some powerful programming language of a general purpose nature? The peculiar fact is that *in general* no such P_2 can be written unless we know *a priori* that P_1 has some limiting restrictions built into it which have the effect of making it a program in a much less powerful programming language than the one in which it would otherwise have been written.

It is important to realize that our argument (which is of a recursive function theoretic nature) [17] about the impossibility of constructing P_2 depends essentially upon the stipulation that P_2 be a *uniform* analyzing program, *i.e.*, one which works over a wide class of P_1's which are themselves programs in a general purpose programming language. To see why this stipulation is made, notice that if the S_i in $\{S\}$ are sentences of a natural language like English; then the task of writing P_1 is substantially an open ended one, perhaps never capable of definitive completion. P_2, on the other hand, need by no means necessarily be an elaborate program. Examples of some kinds of P_2's which one might want to construct are syntactic analyzers which calculate $P_1(S_i)$, or ambiguity testers which determine if there is a unique such $P_1(S_i)$ or grammar equivalence testers which determine of a P_1 and a P_1' whether for all S_i, $P_1(S_i)$ corresponds to $P_1'(S_i)$. In all these cases, if no restrictions are placed upon P_1 other than that it be a program in say FAP or IPL-V or machine language for just about any computer, then none of these different kinds of P_2's are constructable.

The way out of this impasse is, as was suggested above, to restrict P_1 to be a descriptive syntactic theory within a language of much less computing power than the conventional computing languages we are accustomed to using. As an example of a P_1 so restricted, we have the syntactic description of ALGOL. Large parts (but not all) of ALGOL are described by a P_1 which is in the Backus normal form (or a context-free phrase structure language). That the context-free phrase structure languages are incapable of being used as metalanguages to describe object languages of some very simple types is well known [18]. However, this very restriction of the ALGOL syntax to be largely context-free phrase structure in form is what enables syntactic analyzers for ALGOL to be written, even if we allow the possibility that ALGOL will be expanded, so long as the indicated restriction remains satisfied.

Of the many possible models available for descriptive analysis of a language, the one that we elect to use here is the context-free (or so-called "simple") phrase structure language, sometimes called immediate constituent analysis.

The choice of this model for syntactic description of a tiny fragment of English in this paper is dictated largely by expository considerations. Much probably unnecessary controversy exists over the question of what metalanguage is appropriate for the description of English. Many models have been suggested [19] [20] and the suitability of the various models for describing different corpora has been heatedly discussed in the literature.

Two of the candidate models, the phrase structure language and the finite state language, are both represented interestingly enough by the same example that is offered in Fig. 1. Here a grammar is presented for a tiny fragment of English consisting of simple declarative sentences about circles, triangles squares, and polygons, their color, size, and relative positions. The notation used for this context-free or phrase-structure grammar is similar to the Backus normal form [21] but for subsequent application purposes, the notation is modified sightly. To interpret the notation of this example, consider rule 1. This rule indicates that a syntactic category called SENT, which of course denotes "sentence," can be replaced by the concatenation, denoted by the plus symbol, of SUBJ with PRED. In other words, every sentence has a subject followed by a predicate in this simple grammar. Mnemonic symbols are chosen for purposes of clarity. Then, the expansion of the category SUBJ is obtained by choosing either of two alternatives, given in rules 2 and 3, *viz.*, the word "a" followed by the syntactic category NHEADSG or alternatively the word "each" followed by the syntactic category NHEADSG. By continuing to exercise choices among those available, ultimately a sequence of English words will be produced which purportedly constitutes an English sentence. For those familiar with Backus normal form, we indicate below how rules 39

1.	SENT	=	SUBJ + PRED
2.	SUBJ	=	a + NHEADSG
3.		=	each + NHEADSG
4.	NHEADSG	=	NOUN
5.		=	PREMOD + NOUN
6.		=	NOUN + POSTMOD
7.		=	PREMOD + NOUN + POSTMOD
8.	PREMOD	=	ADJS
9.		=	ADJ1 + ADJ2
10.	ADJS	=	ADJ1
11.		=	ADJ2
12.-15.	ADJ1	=	big, large, little, small
16.-17.	ADJ2	=	black, white
18.-20a.	NOUN	=	circle, triangle, square, polygon
21.	POSTMOD	=	UNLIM + COM
22.		=	COM
23.		=	LIM + ADJS
24.	UNLIM	=	REL + is + not
25.		=	REL + is
26.		=	not
27.	LIM	=	REL + is + not
28.		=	REL + is
29.-30.	REL	=	that, which
31.	COM	=	PREP1 + the + LOC
32.		=	RELPHR
33.	PREP1	=	on, at
34.-37.	LOC	=	right, left, top, bottom
38.	RELPHR	=	INTROD + NPHRZ
39.	INTROD	=	PREP1 + the + LOC + of
40.		=	PREP2
41.		=	COMPADJ + than
42.-44.	PREP2	=	in, below, above
45.-48.	COMPADJ	=	bigger, littler, larger, smaller
49.	NPHRZ	=	a + NHEADSG
50.		=	each + NHEADSG
51.	PRED	=	is + COMNA
52.		=	is + not + COMNA
53.	COMNA	=	COM
54.		=	a + NHEADSG

Fig. 1—A grammar for a small fragment of English.

through 41 would be written in that format.

⟨INTROD⟩ = : : ⟨PREP1⟩ / the ⟨LOC⟩ of / ⟨PREP2⟩ / ⟨COMPADJ⟩ than.

Of the infinite number of distinct sentences producible by this grammar, we list below a few typical examples.

1) Each polygon smaller than a black triangle is a square.
2) A big black polygon which is on the left is a triangle.
3) Each triangle that is not in a circle below each square is larger than a polygon which is at the top.

In this grammar, although we have used the context-free phrase structure mechanism, and notwithstanding the fact that this mechanism is capable of syntactically describing languages which are not describable by the so-called finite state grammars, it turns out, nevertheless, to be an accidental fact that the particular grammar that we present is weakly equivalent to a finite state grammar. The reader may, if he wishes, construct a state graph [22] for this grammar by treating words as output and the left to right sequence of unexpanded syntactic categories at any one time as the name of the state in the finite state process for producing the sentence. Such a diagram will turn out to have a finite number of states in it; in fact, it has 79 states. This is, of course, an accident of the peculiarly small language we have chosen to describe with this grammar. In fact, with the addition of one particular rule to the grammar, it becomes impossible to produce a finite state diagram for the modified language. The single rule is shown below.

INTROD = between + a + NPHRZ + and.

The self-embedding introduced by this additional rule makes the language no longer finite state.

As a final illustration of how our simple grammar works, we show in Fig. 2 a tree diagram for the first of the sentences mentioned above as it is produced by or analyzed with respect to the grammar shown in Fig. 1.

In discussing the techniques that are available for syntactic description of a natural language, our illustrations and discussion thus far have been confined to the one dimensional languages. The reason for this is that all of the study which has been devoted thus far to properties of metalanguages for syntactic description has been directed toward the linear, that is one dimensional, languages. For the two dimensional languages, what we want to be able to do is to articulate the two dimensional structure of images in such a way that the well-formed sub-parts of a picture will be assigned to syntactic categories in very much the way that well-formed sub-strings in a sentence grammar are assigned to suitable syntactic categories. Before that can be done, however (and it shall not be done in this paper), it is necessary to solve several problems. It shall be satisfactory here merely to mention some problems which must be solved for the two dimensional case before any satisfactory grammar or syntactic description of an important pictorial source can be written.

The first problem is how to generalize the notion of juxtaposition, which in one dimension reduces to concatenation. The arrangement of words within a sentence in text is apparently entirely describable in terms of the one dimensional notion of concatenation. However, for pictorial sources, quite evidently concatenation must be replaced by at least one two-dimensional notion and perhaps several such operating simultaneously.

As a class of pictorial sources, consider electrical circuit diagrams. Several kinds of juxtaposition can be

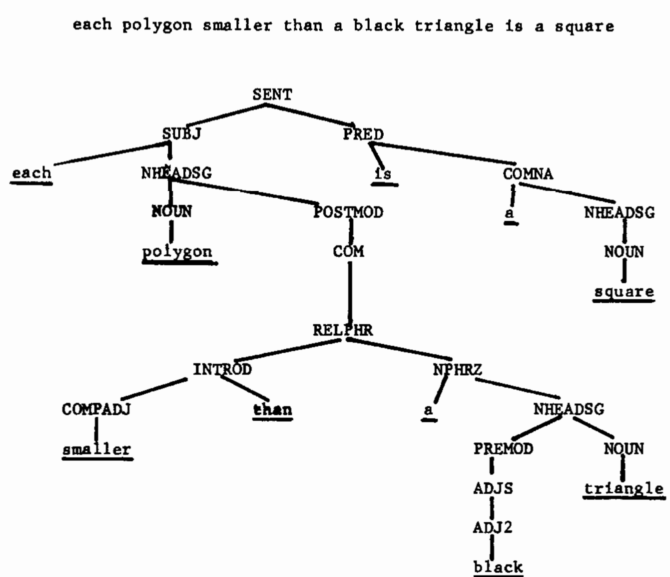

Fig. 2—Tree for a sentence derived from the grammar of Fig. 1.

suggested that will probably be useful for circuit diagrams. First there is a juxtaposition of proximity independent of direction. This is the kind of juxtaposition that relates a resistor symbol to the letter that acts as its label and that is generally used for labelling purposes. A second quite different kind of juxtapostion in two dimensions for circuit diagrams is that which denotes electrical connection between symbols. A third is superimposition. In circuit diagrams, for example, superimposition holds between the symbols for, say, a capacitor and the arrow which denotes variability. A still different kind of juxtaposition in circuit diagrams is left-to-right and sometimes top-to-bottom juxtaposition which usually indicates input-output relations in circuit diagrams. There are other (some quite unusual) kinds of juxtaposition that occur in circuit diagrams.

A possible solution to the choice of a metalanguage in which to represent the syntactic structure of such schematic sources is implicit in a suggestion made by Minsky [23]. His functional notation would be used for denoting the syntactic structure of circuit diagrams by invoking a set of specialized predicates each of which is used to denote a special kind of juxtaposition that exists in a schematic drawing between the symbols that it relates. Unfortunately, it is not yet possible with such a solution to the two dimensional representation problem to produce a syntactic theory having the clarity of a generative grammar such as was exhibited above for English text. The reason for this difficulty is that with several different kinds of functions denoting position in circuit diagrams, there occur conflicts between them, and the rules for resolving these conflicts to produce a diagram whose syntactic description is given are not evident. Thus, the main problem in developing two dimensional grammars is to get a good generalization of the notion of one dimensional concatenation.

Among the other problems that must be solved, there is the problem of how to do syntactic analysis. The scanning of a one dimensional sentence from left to right is a good heuristic and there are others that make a parser or syntactic analyzer efficient. No such simple correspondence suggests itself for circuit diagrams or some of the other pictorial sources.

Another problem is the identification of the primitive symbols or what we call characters. Probably the simplest way to solve that problem is to define those symbols as primitive which are recognizable by suitable character recognition equipment. A final problem in developing a two dimensional grammar is the problem of displaying the results of a syntactic analysis. The tree representation is a convenient two dimensional form for displaying the results of one dimensional syntactic analysis. Perhaps some suitable three dimensional representaion is necessary to display the results of syntactically analyzing two dimensional pictorial sources.

Although there has been no study of two dimensional languages for syntactic description of two dimensional sources, it may be suggestive to see what one might look like. In Fig. 3 is shown a simple two dimensional grammar for a class of pictures consisting of 45° right triangles drawn in the plane which has been divided into unit squares. The alphabet consists of the symbols, V, H, W, L, I, B, R, and the blank symbol. The grammar consists of the ten production rules shown in Fig. 3. If these rules are applied starting with the single symbol, V, as in the left side of production 1, and if the rules are applied an arbitrary number of times in an arbitrary order until no rule applied, then a terminal array of symbols will be obtained which is in the form of a triangle as shown in Fig. 3. There are an infinite number of such triangles obtainable but what is important about our example is that in each such triangle, the symbols which denote it constitute a syntactic analysis of the triangle, the right angle being marked with an R, the two remaining vertices with letters V and W, the hypotenuse letters H, the base with letters B, the other leg with letters L, and the interior points of the triangle with letters I. For such a simple pictorial source as these triangles, the syntactic analysis is self-evident and the mechanism that we exhibit is sufficient to do the analysis of the triangle. It is not clear, however, how to extend the implicit underlying model used here to other pictorial sources of greater interest and importance. This example may suggest some, however.

Although no studies have been reported in the literature of the underlying languages to be used for syntactic description of pictorial sources, there have nevertheless been several successful attempts to describe certain pictorial sources without an explicit underlying model. Eden [24] has shown successfully how cursive handwriting may be described as a language in which certain primitive strokes are juxtaposed according to rules which he makes explicit to form cursive handwriting. The pictorial source here may be considered to be an example of the class that we called "synthetic representations." Grimsdale, *et al.* [25] studied the syntactic structure of hand printed block letters for character recognition purposes. Narasimhan [26] has applied the technique of syntactic description to photographs taken in hydrogen bubble chambers of particle tracks. Finally, in the Sketchpad system of Sutherland [27], there is implicit the syntactic description of the class of drawings consisting of straight lines and circular arcs. In none of these examples, however, has any systematic attempt been made to construct a language for syntactic description and in that language to describe the pictorial sources being processed. As has been suggested above, further study of the underlying languages is needed before such a syntactic description may conveniently be constructed.

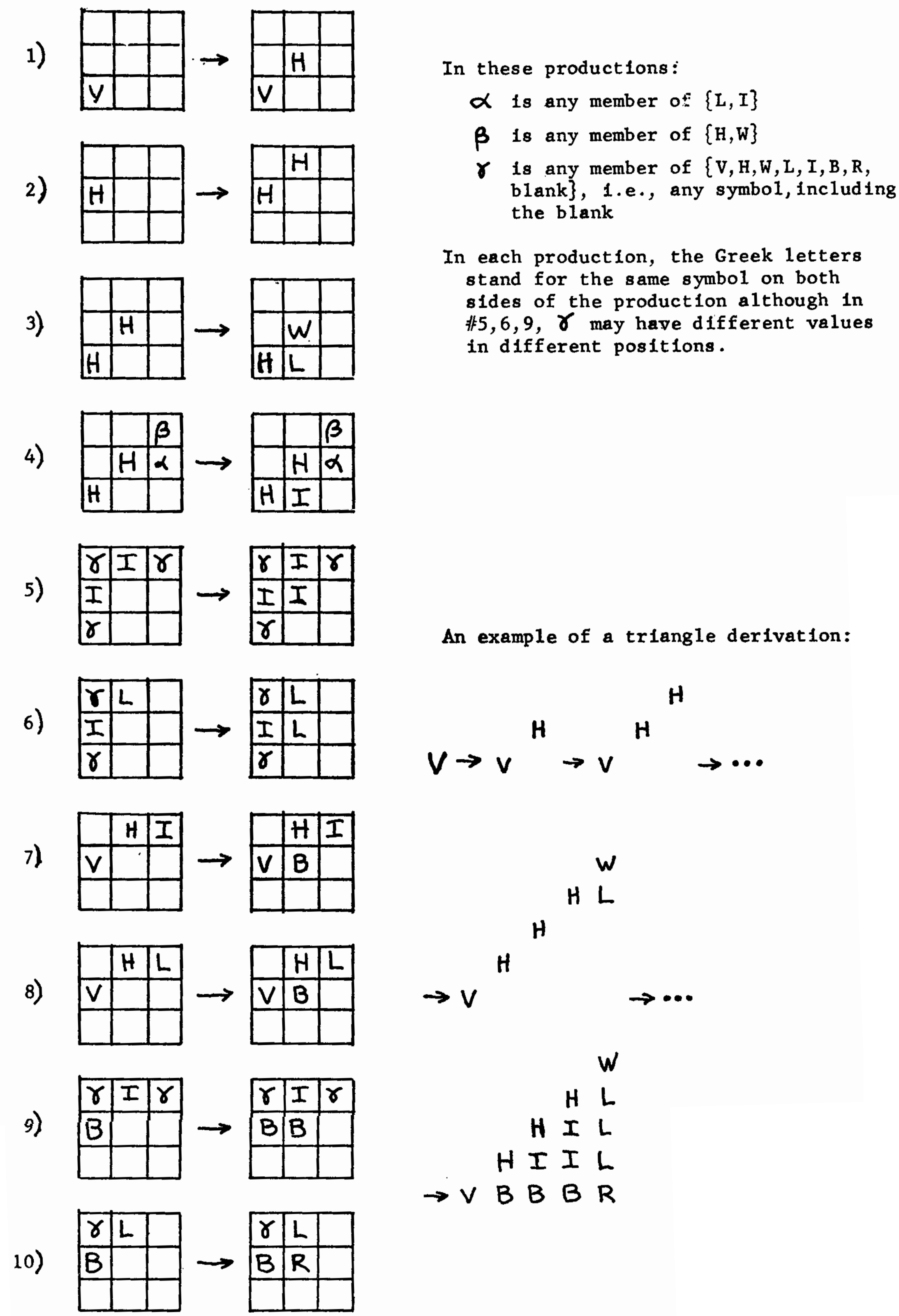

Fig. 3—A two-dimensional grammar for triangles.

IV. Syntax-Directed Interpretation of English Sentences

The two preceding sections have attempted to show that text and pictures may be syntactically described to make explicit some of the implicit properties of such information, and have discussed some techniques that may profitably be used in exhibiting such a syntactic description and in analyzing information sources with respect to such syntactic descriptions. Syntactic analysis, however, from the point of view of this paper, is not an end in itself but an important and probably essential intermediate step in other information processing operations. This section will consider one class of such information processing operations in which the results of a syntactic analysis of English text are used to provide to a machine a certain kind of "understanding" of the English sentences in the text.

The operations that will be performed here upon the syntactically analyzed sentences amount to an interpretation of these sentences. The notion interpretation is used here in a manner roughly analogous to the way it is used in logic. Objects are given in a syntactic metalanguage which, in this case, consists of the syntactically analyzed English sentences. Then, with these objects, or certain ones of them, are associated objects of a different system which need not even be a language. The objects associated with the syntactic objects are their interpretation. Operations upon the objects are also associated with corresponding operations on their interpretations. This process of associating the syntactic and operations on them with the objects and operations of another system will be referred to as interpreting them.

The kind of interpretation in mind here is fairly common in the so-called syntax directed compilers. In such compilers, first there is obtained the syntactic analysis of the input string and then this syntactic analysis is used to produce machine code. The process of producing the machine code from syntactically analyzed input strings is an interpretation process (on objects only in this case) in the sense that the term is used here.

It will appear below that several kinds of information processing operations may be looked upon as instances of interpretation of syntactically analyzed information sources where the syntactic analysis directs the nature of the interpretation operation. Thus, such processes may be called "syntax directed interpretation."

To see in detail how the process of syntax directed interpretation for English sentences, in particular, may be accomplished, the problem of translation into a logical language will be considered in some detail. It is desired to see how a set of English sentences, in particular those given by the grammar in Fig. 1, may be so translated into a logical language representation that it becomes possible by mechanical procedures to determine of the given sentences whether or not they are correct descriptions of simple pictures which they purport to describe. This shall be accomplished by the syntax directed interpretation of the syntactically analyzed English sentences interpreted in a language which is very much like a first order predicate or functional calculus. The well-formed formulae (WFF) of the logical language will consist of predicates or functions denoted by mnemonic strings of letters and arguments shown inside of parentheses. There will be allowed the logical connectives of conjunction shown by the symbol & and implication shown by the symbol $\supset$. Two quantifiers will be allowed, the existential quantifier denoted by $\exists$, and the universal quantifier denoted by $\forall$. The tilde (~) will denote negation.

Fig. 4 shows the rules for interpreting the syntactically analyzed sentences generated by the grammar of Fig. 1. Corresponding to each syntactic replacement rule in the grammar for English sentences of Fig. 1, there is an interpretation rule with an identical number given in Fig. 4. The interpretation rule tells how to transform WFF's of the logical language corresponding to each operation performed upon the syntactic categories of the English language sentences. This process can be illustrated by reference to both Figs. 2 and 5.

In Fig. 2 a particular English sentence producible by the grammar is shown, "each polygon smaller than a black triangle is a square." In the process of producing this sentence according to the syntactic analysis given in the tree diagram of Fig. 2, there are 21 replacement steps to produce the final sentence. In the first step, the syntactic category SENT is replaced by SUBJ followed by PRED. In the next step, SUBJ is expanded into the word "each" followed by NHEADSG, etc. For each of these steps in the syntactic construction of the sentence, there is a corresponding step illustrated in Fig. 5 for the production of the logical translation of this sentence. In step number 1, corresponding to the syntactic category SENT, there is written first the well-formed formula symbol WFF of the logical language. Then when syntax rule number 1 is used to expand SENT correspondingly, interpretation rule number 1 is used to expand WFF to the well-formed formula PRED(SUBJ) which is shown in line 2 of Fig. 5.

The next syntax expansion rule used to produce the given sentence is rule number 3 of the syntax, and corresponding to it, rule number 3 of the interpretation requires that the well-formed formula in line 2 be written in the form shown in line 3 of Fig. 5. If this process is continued, at each step performing an interpretation operation corresponding to the identically numbered syntactic operation, there will be produced, at the end, two objects; the first object is a sentence with its syntactic analysis, and the second is the logical translation of that sentence. The logical translation of the given sentence is shown in line 21 of Fig. 5. It is thus found that the translation of the sentence, "each polygon smaller than a black triangle is a square," is the well-formed formula

$$\forall x((\text{pgn}(x) \;\&\; \exists y(\text{smr}(x, y) \;\&\; \text{bk}(y) \;\&\; \text{tri}(y))) \supset (\text{sq}(x))).$$

1.	WFF	→	PRED (SUBJ)
2.	PRED(SUBJ)	→	∃x (PRED(x) & NHEADSG(x)) where x is a previously unused variable
3.		→	∀x ((NHEADSG(x)) ⊃ (PRED(x))) where x is a previously unused variable
4.	NHEADSG(x)	→	NOUN(x)
5.		→	PREMOD(x) & NOUN(x)
6.		→	NOUN(x) & POSTMOD(x)
7.		→	PREMOD(x) & NOUN(x) & POSTMOD(x)
8.	PREMOD(x)	→	ADJS(x)
9.		→	ADJ1(x) & ADJ2(x)
10.	ADJS(x)	→	ADJ1(x)
11.		→	ADJ2(x)
12.-15.	ADJ1(x)	→	bg(x), lg(x), lt(x), sm(x) respectively
16.-17.	ADJ2(x)	→	bk(x), wt(x)
18.-20a.	NOUN(x)	→	cir(x), tri(x), sq(x), pgn(x) respectively
21.	POSTMOD(x)	→	UNLIMCOM(x)
22.		→	COM(x)
23.		→	LIMADJS(x)
24.	UNLIMCOM(x)	→	~(COM(x))
25.		→	COM(x)
26.		→	~(COM(x))
27.	LIMADJS(x)	→	~(ADJS(x))
28.		→	ADJS(x)
29.-30.			No operation
31.	COM(x)	→	LOC(x) LOC is a unary predicate here
32.		→	RELPHR(x)
33.			No operation
34.	LOC(x)	→	rt (x) if LOC is a unary predicate
	LOC(x,y)	→	rt(x,y) if LOC is a binary predicate
35.	LOC(x)	→	lf(x) if LOC is a unary predicate
	LOC(x,y)	→	lf(xy) if LOC is a binary predicate
36.-37.			Same as 34 and 35 for top(x), top(x,y), bot(x), and bot(x,y) respectively
38.	RELPHR(x)	→	INTROD(x,NPHRZ) i.e., the unary predicate RELPHR becomes a binary predicate INTROD
39.	INTROD(x,NPHRZ)	→	LOC(x,NPHRZ) LOC is a binary predicate here
40.		→	PREP2(x,NPHRZ)
41.		→	COMPADJ(x,NPHRZ)
42.-44.	PREP2(x,NPHRZ)	→	in(x,NPHRZ), bel(x,NPHRZ), abv(x,NPHRZ) respectively
45.-48.	COMPADJ(x,NPHRZ)	→	bgr(x,NPHRZ), ltr(x,NPHRZ), lgr(x,NPHRZ), smr(x,NPHRZ) respectively
49.	Φ(x,NPHRZ)	→	∃y(Φ(x,y) & NHEADSG(y)) where Φ is any binary predicate having NPHRZ as an argument and Φ(x,y) is the result of substituting any previously unused variable, y, for the occurence of NPHRZ.
50.		→	∀y((NHEADSG(y)) ⊃ (Φ(x,y))) Same as for 49
51.	PRED(x)	→	COMNA(x)
52.		→	~(COMNA(x))
53.	COMNA(x)	→	COM(x)
54.		→	NHEADSG(x)

Fig. 4—Interpretation rules for Fig. 1 grammar.

A similar process may be carried out for any sentence of the infinite number of sentences that can be generated from the grammar of Fig. 1. For each such sentence, there is a logical translation and it is obtained by carrying out the process given in Fig. 4.

The reason for obtaining the logical translation of such sentences is that from the logical translation it is very easy to determine whether the given sentences are true or false descriptions of patterns that they purport to describe. Let us consider how one might determine for the sentence of our example whether it is a correct description of a picture. For the given picture, one would first have to determine what the syntactically well-formed objects are in the simple pictures which the sentences of our grammar can describe. These syntactically well-formed objects would be what we have once called "blobs," that is, disjoint objects in the picture [28]. The variables denoted by x, y, z, etc., in our logical language will range over the blobs in the picture. Then a character-recognition process would determine for any arbitrary blob whether it had the property of being a polygon or of being black or of being a triangle or of being a square, each of which is denoted, respectively, in our logical language by the symbols pgn, bk, tri, sq. A simple subroutine could determine of two such blobs, x and y, whether x is smaller than y, *e.g.*, in area, in which case the logical predicate smr(x, y) would be true for x and y. Finally, to determine of the whole logical expression whether it is true for a given picture, it is necessary to find out by a search program whether for every object x, if the character recognizer says that it is a polygon and there is an object y such that x is smaller than y, and the character recognizer says that y is black and that y is a triangle, then, in such case, the character recognizer says that x is a square. If this is the case, then indeed the logical expression given in line 21 of Fig. 5 is satisfied by the given picture and one may say that the original sentence is a correct (partial) description of the given picture.

The reader may wish to test his own implicit character-recognition procedure on the three pictures of Fig. 6 and verify that the original sentence, the logical translation of which is given in Fig. 5, is a true description of the first two pictures in Fig. 6 and not a true description of the third.

This example has illustrated how the truth-functional evaluation of simple sentences as descriptions of simple pictures may be mechanically accomplished. The truth-functional properties that are assigned to the English sentences are actually properties of the logical language WFF's into which the sentences have been translated. If the sentences in the logic language are intuitively acceptable interpretations of the original English sentences, then to that extent the truth-functional evaluation of the logical sentences is a corresponding truth-functional evaluation of the original English sentences. For other purposes, different logical languages or perhaps no logical language at all is needed. If one's purpose were to do question answering, the different types of questions might require processing operations of logically distinct types. Thus, for answering yes-no types of questions, a mechanism virtually identical to the one illustrated here would be sufficient. For answering the wh-type questions (those introduced by "which," "what," "when," etc.), one would use a logical language in which properties other than truth-functional ones could be evaluated, that is, in which the values of certain functions would be objects or the names of objects within the picture.

An attractive alternative to the use of logical languages suggests itself in the possibility of using the language of the syntactically analyzed English sentence as the logical language itself. The reason for using a logical

```
1.   WFF
2.   PRED(SUBJ)
3.   ∀x((NHEADSG(x)) ⊃ (PRED(x)))
4.   ∀x((NOUN(x) & POSTMOD(x)) ⊃ (PRED(x)))
 :
7.   ∀x((pgn(x) & RELPHR(x)) ⊃ (PRED(x)))
8.   ∀x((pgn(x) & INTROD(x,NPHRZ)) ⊃ (PRED(x)))
 .
10.  ∀x((pgn(x) & smr(x,NPHRZ)) ⊃ (PRED(x)))
11.  ∀x((pgn(x) & ∃y(smr(x,y) & NHEADSG(y))) ⊃ (PRED(x)))
12.  ∀x((pgn(x) & ∃y(smr(x,y) & PREMOD(y) & NOUN(y))) ⊃ (PRED(x)))
 :
20.  ∀x((pgn(x) & ∃y(smr(x,y) & bk(y) & tri(y))) ⊃ (NOUN(x)))
21.  ∀x((pgn(x) & ∃y(smr(x,y) & bk(y) & tri(y))) ⊃ (sq(x)))
```

Fig. 5—Steps in developing a WFF translation for the sentence of Fig. 2.

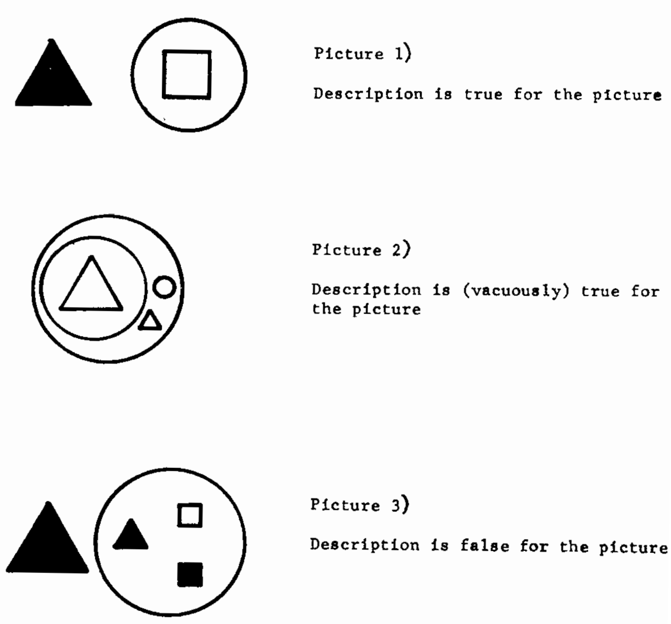

Fig. 6—Three pictures for testing the logical translation shown in Fig. 5.

language is that the logical properties of it are usually well understood. However, to use a logical language necessitates a translation into that language. The alternative of using the language of the syntactically analyzed source sentences requires that if one were interested in performing such processes as inference, question answering, or truth-functional evaluation, then these processing operations would have to be reformulated in terms of the language used for the syntactic analysis of the English sentences. This possibility, however, will not be further explored here.

V. Syntax-Directed Interpretation in Other Applications

In the example discussed in Section IV, a syntax-directed interpretation process was used to match syntactically analyzed sentences against syntactically analyzed pictures, although, admittedly, syntactic analysis of the pictures was relatively trivial. The process of syntax-directed interpretation, however, has many other applications which we wish briefly to mention in this section.

The first application is to the process of performing inference on natural language text which can clearly be done to the extent that text can be translated into a logical language as we illustrated in the simple example of the previous section. The inference in turn corresponds to a natural-language inference. This possibility of performing inference has been a main concern of symbolic logic for a long time. What is apparently new, however, is the possibility of making use of an exhaustive syntactic analysis as the input which can direct the process of translation into a logical language. The language which is used for representing the syntactically-analyzed sentences can itself serve as a logical language in terms of which rules of inference may be constructed. This possibility does not seem to have been explored previously.

Another application of syntax-directed interpretation is encoding of text for storage and transmission. The usual procedures for encoding text make use of letter frequencies, or letter pair, or in some cases, higher order n tuples of letters, or perhaps words and their corresponding frequencies. Here the probability distribution used for designing a code is based on observed statistics of various corpora. A different possibility, however, would be based upon a probability distribution that comes directly out of a grammar for the underlying language.

Suppose that in the grammar of Fig. 1, each of the alternative choices made in the process of generating a sentence were assumed to be equally likely. This assumption would induce a probability distribution on the sentences produced which would have the desirable property of usually associating low probabilities with longer sentences and higher probabilities with shorter sentences. Thus, the sentence used for an example in Fig. 2 would have a probability of $2^{-22} \cdot 3^{-2}$ or approximately 2^{-26}. A Huffman code based on this probability distribution (described in Fano [29]) would thus represent the sentence "each polygon smaller than a black triangle is a square," with a 26-bit sequence.

There are applications of the syntax-directed interpretation process to the problem of inductive inference. A technique such as the one described immediately above for assigning probabilities to sentences can similarly be used to assign probabilities to the events described by such sentences. Then when one wishes to assign probabilities to alternative hypotheses about a particular universe of discourse (pictures in our simple case above), the probabilities that are assigned to the hypotheses are the probabilities assigned to the sentences that denote these hypotheses.

For complex languages like English, this approach to inductive inference offers an interesting alternative to the one suggested by Carnap and Bar-Hillel [30] and the one suggested by Kochen [31].

Syntax directed interpretation for pictorial sources is an important possibility for investigation. The first instance of syntax directed interpretation is in the encoding of pictures for storage and transmission purposes. The usual approach to picture encoding makes use of picture statistics such as run-length statistics and the design of matching codes for these statistics. Code compressions up to about 10 to 1 have been achieved by such techniques. However, for certain pictorial sources like schematic diagrams, the explicit syntactic analysis of such diagrams and then the use of syntax directed coding techniques would very likely lead to a considerable further code compression for storage or transmission purposes. Thus, for an electrical circuit diagram, the list of the names of the circuit components (the primitive symbols) and the list of the syntactic relations that hold among them would be, for most purposes, a sufficient characterization of the diagrams to serve the purposes of reproducing them. We would estimate that if an exhaustive grammar of electrical circuit diagrams were to be written, then the encoding or transmission of such diagrams as information sources could be done with a code compression of perhaps 100 to 1 or 1 decimal order of magnitude greater than can be achieved with statistical coding techniques [32].

Another pictorial language which could lend itself to syntax directed interpretation is the language of chemical structure diagrams. Much attention has been devoted to techniques for going from such diagrams by manual procedures to linear ciphers, and some effort has been devoted to machine manipulation of some of these ciphers [33]. There seems further to be the possibility of providing a syntactical characterization of chemical structure diagrams as pictorial sources and the possibility of manipulating these diagrams as the codes which represent chemical compounds rather than their linear cipher representations.

VI. Conclusion

In this paper, an attempt has been made to show that the class of information sources occurring in technical and scientific documents which contain text and pictures can be uniformly treated as languages. They are languages in the strong sense that descriptive theories of these languages can be exhibited by means of computer processing techniques that are available or with some extensions of these techniques. Such descriptive theories for these languages can provide to machines an effective partial substitute for the cultural environment in which human beings ordinarily encounter the languages. With such descriptive theories of text and picture sources, a machine can analyze individual information items at a purely syntactic (formal) level. It appears that if the descriptive theory goes deep into the structure of the underlying language, then the corresponding syntactic analysis made by a machine can lead, through the process of syntax direction, to an interpretation of these languages by entirely automatic procedures. It is tempting to identify these interpretation operations with the informal notion of "understanding." The methods of syntax directed interpretation for natural languages described here will appear attractive to those who wish to provide machines with the very elaborate structural details that are known and can be discovered about natural languages.

Acknowledgment

Many of the ideas in this paper arose in discussions with the author's colleagues to whom credit is gratefully given: S. Cappell, D. Cohen, A. J. Goldman, R. W. Hsu, E. Jones, J. Randolf, B. K. Rankin III, W. A. Sillars and W. C. Watt. Rankin and Randolf wrote grammars similar to those of Figs. 1 and 3 and Sillars wrote a translation algorithm similar to that of Fig. 4. Finally, to Mrs. Arlene V. Ruhl who did all the output work, special thanks are extended.

Bibliography

[1] S. M. Newman, R. W. Swanson, and K. Knowlton, "A Notation System for Transliterating Technical and Scientific-Texts for Use in Data Processing Systems," U. S. Patent Office, Washington, D. C., Research and Dev. Rept., No. 15; May 22, 1959.

[2] R. Carnap, "Introduction to Symbolic Logic and Its Applications," Dover Publications, Inc., New York, N. Y., Ch. B, Sec. 20; 1958.

[3] R. W. Floyd, "On the Non-existence of a phrase structure grammar for ALGOL 60," *Commun. ACM*, vol. 5 pp. 483–484; September, 1962.

[4] Z. S. Harris, "String Analysis of Sentence Structure," N. V. Mouton and Company, The Hague, The Netherlands; 1962.

[5] R. S. Wells, "Immediate Constituents," *Language*, vol. 23, pp. 81–117; April–June, 1947.

[6] N. Chomsky, "Syntactic Structures," N. V. Mouton and Company, The Hague, The Netherlands; 1957.

[7] M. E. Stevens, "Automatic Character Recognition: A State-of-the-Art Report," National Bureau of Standards, Washington, D. C., Tech. Note 112, PB 161613; May, 1961.

[8] E. H. Gombrich, "Art and Illusion," Pantheon, New York, N. Y.; 1960.

[9] S. Ginsburg and H. G. Rice, "Two Families of Languages Related to ALGOL," System Development Corp., Santa Monica, Calif., Rept. No. TM-578; July, 1961, and *J. ACM*, p. 350, March, 1962.

[10] D. G. Bobrow, "Syntactic Analysis of English by Computer—A Survey," Proc. Fall Joint Computer Conf., vol. 24, pp. 365–387; 1963.

[11] H. P. Edmundson and R. E. Wyllys, "Automatic abstracting and indexing, survey and recommendations," *Commun. ACM*, vol. 4, pp. 226–234; May, 1961.

[12] S. Klein, "Some Experiments Performed with an Automatic Paraphraser," System Development Corp., Santa Monica, Calif.; 1963.

[13] E. A. Feigenbaum and J. Feldman, "Computers and Thought," McGraw-Hill Book Co., New York, N. Y.; 1963. (See the chapters by R. K. Lindsay and B. F. Green Jr., pp. 207–223.)

[14] S. Kuno and A. G. Oettinger, "Syntactic Structure and Ambiguity of English," *Proc. Fall Joint Computer Conf.*, pp. 397–418; 1963.

[15] R. F. Simmons, S. Klein and K. McConlogue, "Co-occurrence and Dependency Logic for Answering English Questions," Systems Development Corp., Santa Monica, Calif., SP-1155, pp. 1–29; April 3, 1963.

[16] A. F. Brown, "Normal and Reverse English Word List," University of Pennsylvania Press, Philadelphia, Pa.; January, 1963.

[17] M. Davis, "Computability and Unsolvability," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958.

[18] R. W. Floyd, "The Syntax of Programming Languages—A Survey," this issue, page 346.

[19] N. Chomsky, "Formal Properties of Grammars," in "Handbook of Mathematical Psychology," vol. 2, R. R. Bush, E. H. Galanter, and R. D. Luce, eds., John Wiley and Sons, Inc., New York, N. Y.; 1962.

[20] V. H. Yngve, "A Model and an Hypothesis for Language Structure," *Proc. American Philsophical Soc.*, vol. 104, pp. 444–466; October, 1960.

[21] J. W. Backus, "The Syntax and Semantics of the Proposed International Algebraic Language—Zurich ACM-GAMM Conference," *Proc. Internat'l Conf. on Information Processing*, pp. 125–132; June, 1959.

[22] R. F. McNaughton and H. Yamada, "Regular Expressions and State Graphs for Automata," IRE Trans. on Electronic Computers, vol. EC-9, pp. 39–47; March, 1960.

[23] M. L. Minsky, "Steps Toward Artificial Intelligence," Proc. IRE, vol. 49, pp. 8–30; January, 1961.

[24] M. Eden, "On the Formalization of Handwriting," in "Structure of Language and Its Mathematical Aspect," American Mathematical Society, Providence, R. I., pp. 83–88; 1961.

[25] R. L. Grinsdale, *et al.*, "A System for the Automatic Recognition of Patterns," *Proc. IEE*, vol. 106, Pt. B, pp. 210–221; March, 1959.

[26] R. Narasimhan, "Syntactic Description of Pictures and Gestalt Phenomena of Visual Perception," Digital Computer Laboratory, University of Illinois, Urbana; July 25, 1963.

[27] I. E. Sutherland, "Sketchpad, A Man-Machine Graphical Communication System," *Proc. SJCC*, vol. 23, pp. 329–346; 1963.

[28] R. A. Kirsch, L. Cahn, L. C. Ray, and G. H. Urban, "Experiments in Processing Pictorial Information with a Digital Computer," *Proc. EJCC*, pp. 221–229; December, 1957.

[29] R. M. Fano, "Transmission of Information," Mass. Inst. Tech. Press, Cambridge, Mass., 1961. (see pp. 75ff.)

[30] R. Carnap and Y. Bar-Hillel, "An Outline of a Theory of Semantic Information," Research Lab. of Electronics, Mass. Inst. of Tech., Cambridge, Mass., Tech. Rept. No. 247; October 27, 1952.

[31] M. Kochen, "An Experimental Program for the Selection of 'Disjunctive Hypotheses'," *Proc. WJCC*, vol. 19, pp. 571–578; May 1961.

[32] Cherry, C., M. H. Kubba, D. E. Pearson, and M. P. Barton, "An Experimental Study of the Possible Bandwidth Compression of Visual Image Signals," Proc. IRE, vol. 51, pp. 1507–1517; November, 1963.

[33] L. C. Ray and R. A. Kirsch, "Finding Chemical Records by Digital Computers," *Science*, vol. 126, pp. 814–189; October 25, 1957.

[34] P. Ziff, "Semantic Analysis," Cornell University Press, Ithaca, N. Y.; 1960. (See esp. Section 21 and discussion of Ziff's "E**.")

Reprinted from IEEE TRANSACTIONS
ON ELECTRONIC COMPUTERS
Volume EC-13, Number 4, August, 1964

PRINTED IN THE U.S.A.