FEIGENBAUM, E. A., AND FELDMAN, J. *Computers and Thought.* McGraw-Hill, New York, 1963.
KUNO, S. The predictive analyzer and a path elimination technique. *Comm. ACM 8*, 7 (July, 1965), 453–462.
LONGYEAR, C. The semantic rule. Rep. No. 67TMP-55, General Electric TEMPO Project, Santa Barbara, Calif., 1967.
MINSKY, M. *Semantic Information Processing.* MIT Press, Cambridge, Mass., 1968.
NEWELL, A., SHAW, J. C., AND SIMON, H. A. The processes of creative thinking. In *Contemporary Approaches to Creative Thinking*, H. E. Gruber, G. Terrell, and M. Wertheimer (Eds.), Atherton Press, New York, 1962, pp. 63–119.
PIAGET, J. *The Psychology of Intelligence.* Routledge and Kegan Paul, London, 1950.
QUILLIAN, M. R. Semantic Memory. In *Semantic Information Processing*, M. Minsky, The MIT Press, Cambridge, Mass., 1968.
——. Word concepts: a theory and simulation of some basic semantic capabilities. *Behav. Sci. 12* (1967), 410–430.
——, WORTMAN, P. AND BAYLOR, G. W. The programable Piaget: Behavior from the standpoint of a radical computerist. Unpublished dittoed paper, Carnegie-Melon U., Pittsburgh, Pa., 1965.
RAPHAEL, B., BOBROW, D. G., FEIN, L., AND YOUNG, J. W. *A Brief Survey of Computer Languages for Symbolic and Algebraic Manipulation.* North-Holland Pub. Co., Amsterdam, 1968.
REITMAN, W. R. *Cognition and Thought: An Information Processing Approach.* Wiley, New York, 1965.
SIKLOSSY, L., AND SIMON, H. A. Some semantic methods for language processing. Complex information processing Paper #129, Carnegie-Mellon U., Pittsburgh, Penn., 1968.
SIMMONS, R. F. Answering English questions by computer: a survey. *Comm. ACM 8*, 1 (Jan. 1965), 53–70.
—— AND BURGER, J. F. A semantic analyzer for English sentences. Rep. No. SP 2987, Syst. Develop. Corp., Santa Monica, Calif., 1968.
TESLER, L., ENEA, H., AND COLBY, K. M. A directed graph representation for computer simulation of belief systems. Dep. Comput. Sci., Stanford U., Stanford, Calif., 1967.
THOMPSON, F. B. The deacon project. Rep. No. 65TMP-69, General Electric TEMPO Project, Santa Barbara, Calif., 1965.
THORNE, J. P., BRATLEY, P., AND DEWAR, H. The syntactic analysis of English by machine. In *Machine Intelligence 3*, Donald Michie (Ed.), American Elsevier, New York, 1968, pp. 281–309.
WEIZENBAUM, J. Contextual understanding by computer. *Comm. ACM 10*, 8 (Aug. 1967), 474–480.

A Program for the Syntactic Analysis of English Sentences

HAMISH DEWAR, PAUL BRATLEY,*
AND JAMES PETER THORNE
Edinburgh University,† Scotland

A program is described which produces syntactic analyses of English sentences with respect to a transformational grammar. The main features of the analyzer are that it uses only a limited dictionary of English words and that it pursues all analysis paths simultaneously while processing the sentence from left to right.

The form of representation used for the dictionary and the grammar is indicated and an outline account is given of the analysis procedure. Techniques for keeping the size of the analysis record within reasonable limits and for avoiding the need for dynamic application of certain transformational rules are described.

A number of examples of output produced by the program are given. The output includes timing information.

KEY WORDS AND PHRASES: syntactic analysis, language processing, language analysis, parsing, analysis procedure, recognition procedure, English sentences, linguistics, psycholinguistics, transformational grammar, limited dictionary, predictive analysis
CR CATEGORIES: 3.42, 3.36

This work was supported by the UK Office for Scientific and Technical Information (grant number ID/102/2/06).
* Present address: Department d'Informatique, University of Montreal, Montreal, 3, Quebec, Canada.
† H. Dewar and P. Bratley are with the Computer Science Department; J. P. Thorne is with the English Language Department.

Introduction

The input to the analysis program consists of English sentences in more or less normal orthographic form. The sentences are read and processed a word at a time, and at the end of each sentence the analysis or analyses produced are displayed on the printer. In analyzing the sentences the program makes use of information about the syntactic functions of individual words derived from a limited (closed-class) dictionary and information about sentence structure derived from a representation of a grammar. The main characteristics of the program can be summarized as follows:

(i) it does not require access to a complete dictionary of English;
(ii) it utilizes a predictive technique;
(iii) each sentence is processed word by word from left to right;
(iv) in the case of structurally ambiguous sentences or parts of sentences, all the analyses are pursued simultaneously (i.e. without backtrack);
(v) the analysis procedure is a one-stage operation.

These features were incorporated in the analyzer chiefly because one of the objectives in designing the program was to explore the possibility of constructing a device which would not simply analyze English sentences but which would also to some extent model the way in which we ourselves recognize the syntactic structure of sentences which we hear or read. That is, the construction of the analyzer was in the nature of a psycholinguistic experiment [1]. On this basis, each of the features listed above seemed to have reasonable a priori justification.

The Closed-Class Dictionary

The program has access to a dictionary which contains grammatical information about a limited number of words. These are words like prepositions, conjunctions, articles, auxiliaries, and pronouns which have fixed syntactic roles and play an essential part in the determination of sentence structure. In addition it contains a number of inflections (e.g. *-s*, *-ed*, *-ing*) which are syntactically significant. It does not include words like simple verbs (as opposed to double object verbs, sentential complement verbs, etc.), nouns and adjectives which belong to open classes—that is, classes which are indefinitely extendible.

The internal representation of the dictionary is in the form of a tree structure, the nodes of the tree storing the individual letters of the words and inflections. The terminal nodes of the tree (representing end-of-word) contain an operation code and a pointer to an element in an array of dictionary entries. Each entry comprises one or more codings, a coding being a category name and a set of values for the *features* (e.g. number, case, tense) associated with the category. Feature values are stored in the form of binary patterns segmented into fields which correspond to individual features. Any number of words may be linked to the same dictionary entry, possibly with different operation codes. The (nonzero) operation codes serve to express systematic relationships between different forms; for example, the word *my* is linked to the same entry as the word *I*, but it carries the operation code for possessive form. The terminal nodes for inflections specify only an operation code and not an entry. The following fragment of the dictionary in the form in which it is submitted to the set-up routine illustrates the way in which information is presented:

```
although, though : CONJ 0001
            have : HAVE 3680
                 + VERB 3682
             has = have (8)
             had = have (2)
           built = -ed
             -ss = -
```

To provide flexibility in the choice of exactly which words should be listed in the dictionary [2], the entry format is quite general and the dictionary look-up procedure in the program assigns codings to all input words whether or not they are listed. Standard entries are included for open-class words, proper names and numerals. For a word not explicitly listed in the dictionary the tree search leads (via default links) to a terminal node carrying the appropriate operation code if the word is inflected and to one of the standard entries if it is not. The codings which are transmitted to the analysis procedure are produced by applying the operation denoted by the operation code to the feature values in the located entry for the word or, in the case of inflections, the entry for the stem of the word.

The Grammar

The syntactic information available to the analyzer is based on a version of a transformational grammar. This version differs in some respects from the formulations of such grammars which are current in linguistic theory [3].

The base component is a *regular* (or finite-state) grammar. That is, it generates a set of linear strings without assigning hierarchical structure to them. Each element in the strings is a triple, consisting of a syntactic relation marker (SRM), a category label, and a set of feature values. The SRMs specify syntactic function (e.g. Subject, Object, Attribute); the category labels specify syntactic form (e.g. Pronoun, Adverb, Relative Clause); and the feature values (like those which appear in the dictionary) subcategorize with respect to the category. Generalized feature-matching rules govern the substitution for these elements of lexical units or transforms; a sentence produced from a base string using only lexical substitution rules is a kernel sentence (see Figure 1, e.g. examples 1, 5, 6). The realization of a syntactic relation by a transform is assimilated to the case of lexical realization by distinguishing the generalized substitution process from the transformational process. Thus the effect of many transformational rules is to produce as resultants, forms which may realize relations which are also realizable by individual words; loosely speaking, they map sentence forms to lexical forms.

The postulation of a regular grammar, rather than the customary context-free grammar, for the base component may seem surprising. Formally a context-free grammar differs from a regular grammar in its ability to cover recursion. Thus the adoption of a regular grammar involves attributing all (genuinely) recursive constructions in the language to the operation of the transformational component. The principle that recursion implies transformation seems to be generally valid, although it has possibly controversial consequences in some cases (for example, the noun phrase in English). Its advantage for a syntactic analysis procedure is that transformational rules defined (initially) on a regular grammar are formally more tractable than transformational rules defined on a context-free grammar. Many of them can be regarded as metarules, in the sense that the "structure indexes" on which they operate and their resultants are in the form of rule-readings, rather than phrase-markers.

A base component which is a regular grammar can conveniently be represented by a finite-state network (directed graph). The resultants of many transformational rules can be represented—as derived rules—in the same form. For this reason the bulk of the syntactic information available to the analyzer is contained in a finite-state network which represents not only the base rules but also a large number of transforms. This is why it is important that provision is made in the format of the base grammar for the specification of SRMs and feature values; in transformed strings, syntactic relations remain marked and selectional con-

(*Text continues on page 479*)

SYNTACTIC RELATION MARKERS

				CATEGORY MARKERS	
OO	inverted item	CO	complement	STAT	statement
SM	sentence modifier	IN	indirect object	QUES	question
PM	predicate modifier	PD	predeterminer	IMP	imperative
MO	other modifier	DE	determiner	INDS	indirect statement
SU	subject	QU	quantifier	INFC	infinitive clause
AU	auxiliary	AT	attribute	REL	relative clause
AV	active verb	HE	head	GER	gerund
PV	passive verb	SA	subject adjunct	CNP	complex noun phrase
CV	copulative verb	OB	object	PARC	participial clause

```
1  John helped Mary.
     (time taken: 0.511 sec    nodes used: 18)

 1                    STAT  1
 2  SU:John AV:helped OB:Mary  2
```

```
2  John helped the girl.
     (time taken: 0.557 sec    nodes used: 25)

 1                        STAT  1
 2  SU:John AV:helped OB:   CNP  2
 3                  DE:the HE:girl  3
```

```
3  The boy helped the girl.
     (time taken: 0.480 sec    nodes used: 26)

 1                              STAT  1
 2  SU:      CNP AV:helped OB:    CNP  2
 3  DE:The HE:boy          DE:the HE:girl  3
```

```
4  Why did the chicken cross the road?
     (time taken: 0.717 sec    nodes used: 30)

 1                                                  QUES  1
 2  OO:Why OO:did SU:      CNP AU:* AV:cross OB:    CNP PM:*  2
 3              DE:the HE:chicken          DE:the HE:road     3
```

```
5  Chew gum.
     (time taken: 0.438 sec    nodes used: 18)

 1              IMP  1
 2  SU:* AV:Chew OB:gum  2
```

```
6  We are going to London.
     (time taken: 0.462 sec    nodes used: 14)

 1                          STAT  1
 2  SU:We AU:are AV:going PM:to + London  2
```

```
7  Last week we visited John.
     (time taken: 0.945 sec    nodes used: 37)

 1                              STAT  1
 2  PM:      CNP SU:we AV:visited OB:John  2
 3  QU:Last HE:week                       3
```

```
8  It is easy to make a mistake.
     (time taken: 0.616 sec    nodes used: 24)

 1                                  STAT  1
 2  SU:It CV:is CO:easy SA:         INFC  2
 3                  to AV:make OB:   CNP  3
 4                        DE:a HE:mistake  4
```

```
9  Anyone can make a mistake.
     (time taken: 0.520 sec    nodes used: 23)

 1                              STAT  1
 2  SU:Anyone AU:can AV:make OB:   CNP  2
                         DE:a HE:mistake  3
```

```
10  How difficult was it to find digs?
      (time taken: 0.705 sec    nodes used: 24)

 1                                              QUES  1
 2  OO:How + difficult OO:was SU:it CV:* CO:* SA:  INFC  2
 3                                  to AV:find OB:CNP  3
 4                                          HE:digs    4
```

```
11  He observed the girl with the telescope.
      (time taken: 0.698 sec    nodes used: 32)

 1                                          STAT  1
 2  SU:He AV:observed OB:                    CNP  2
 3                  DE:the HE:girl MO:with +  CNP  3
 4                                  DE:the HE:telescope  4
```

```
 1                                          STAT  1
 2  SU:He AV:observed OB:    CNP PM:with +    CNP  2
 3                DE:the HE:girl  DE:the HE:telescope  3
```

```
12  He observed her with the telescope.
      (time taken: 0.594 seconds    nodes used: 25)

 1                                  STAT  1
 2  SU:He AV:observed OB:her PM:with +  CNP  2
 3                        DE:the HE:telescope  3
```

```
13  Which magazines do you prefer?
      (time taken: 0.705 sec    nodes used: 26)

 1                                  QUES  1
 2  OO:      CNP OO:do SU:you AU:* AV:prefer OB:*  2
 3  DE:Which HE:magazines                        3
```

```
14  The cat and dog play.
      (time taken: 0.618 sec    nodes used: 26)

 1                        STAT  1
 2  SU:          CNP AV:play  2
 3  DE:The [HE:cat and HE:dog]  3
```

```
15  Candy is dandy but liquor is quicker.
      (time taken: 0.889 sec    nodes used: 31)

 1                                          STAT  1
 2  [SU:Candy CV:is CO:dandy but SU:liquor CV:is CO:quicker]  2
```

```
16  I like bathing beauties.
      (time taken: 0.598 sec    nodes used: 21)

 1                        STAT  1
 2  SU:I AV:like OB:       CNP  2
 3          AT:bathing HE:beauties  3
 1                      STAT  1
 2  SU:I AV:like OB:     GER  2
 3          SU:* AV:bathing OB:  CNP  3
 4                      HE:beauties  4
```

```
17  Will you tell John to bring back the book?
      (time taken: 0.630 sec    nodes used: 26)

 1                                                  QUES  1
 2  OO:Will SU:you AU:* AV:tell IN:John OB:         INFC  2
 3                          to AV:bring back OB:     CNP  3
 4                                  DE:the HE:book   4
```

```
18  I can give you a rough estimate.
      (time taken: 0.661 sec    nodes used: 20)

 1                              STAT  1
 2  SU:I AU:can AV:give IN:you OB:  CNP  2
 3                DE:a AT:rough HE:estimate  3
```

```
19  Did you see the house he built?
      (time taken: 0.774 sec    nodes used: 29)

 1                                          QUES  1
 2  OO:Did SU:you AU:* AV:see OB:           CNP  2
 3                  DE:the HE:house AT:     REL  3
 4                          OO:* SU:he AV:built OB:*  4
```

```
20  The film which Punch recommended was banned.
      (time taken: 0.726 seconds    nodes used: 24)

 1                                  STAT  1
 2  SU:                  CNP AU:was PV:banned  2
 3  DE:The HE:film AT:       REL              3
 4          OO:which SU:Punch AV:recommended OB:*  4
```

```
21  The policeman stopped and questioned him.
      (time taken: 0.879 sec    nodes used: 39)

 1                                  STAT  1
 2  SU:      CNP [AV:stopped and AV:questioned OB:him]  2
 3  DE:The HE:policeman                               3
```

```
 1                                  STAT  1
 2  SU:      CNP [AV:stopped and AV:questioned] OB:him  2
 3  DE:The HE:policeman                               3
```

```
22  He likes reading Shakespeare's plays and performing them.
      (time taken: 1.250 sec    nodes used: 52)

 1                                          STAT  1
 2  SU:He AV:likes OB:                       GER  2
 3          SU:* [AV:reading OB:  CNP and AV:performing OB:them]  3
 4                  DE:Shakespeare's HE:plays                    4
```

```
23  Take an egg and beat it.
      (time taken: 0.781 sec    nodes used: 41)

 1                          IMP  1
 2  SU:* [AV:Take OB:    CNP and AV:beat OB:it]  2
 3          DE:an HE:egg                        3
```

```
24  The butler did what we wanted him to do.
      (time taken: 0.821 sec    nodes used: 35)

 1                                          STAT  1
 2  SU:      CNP AV:did OB:                  NOMC  2
 3  DE:The HE:butler    OO:what SU:we AV:wanted IN:him OB:  INFC  3
 4                                  to AV:do OB:*  4
```

```
25  Where have you and your father been hiding?
      (time taken: 0.778 sec    nodes used: 27)

 1                                          QUES  1
 2  OO:Where OO:have [SU:you and SU:  CNP] AU:* AU:been AV:hiding PM:*  2
 3                        DE:your HE:father                            3
```

FIG. 1. Examples of output. An asterisk marks the position from which an *inverted item* has been displaced. In constructions with no inverted items, it marks the position of a deleted *proform*.

straints are preserved. It is interesting to note that, even with the addition of transforms, the grammar remains quite compact. This is largely because of the substantial degree of right-equivalence among the rule strings.

In addition to the substitution rules, some general transformational rules are applied dynamically in the course of the process of analysis. These are rules concerned with constructions which exhibit inversion (as in Questions and Relatives) and coordination (involving words like *and* and *or*) and they are implemented by specific parts of the analysis procedure. It is obviously a weakness that they are implemented in this way, but we have been unable to find a satisfactory method for dealing with them within a more general framework.

Certain constructions (for example, those containing *as* and *than*) are not handled at all. This is largely attributable to the fact that no adequate account of the grammar of these constructions has as yet been provided by linguists.

The Analysis Procedure

The task of the analysis procedure is essentially the progressive construction of a data structure in which the predictions satisfied by successive words of the sentence to be analyzed are recorded and which is then used to determine what new predictions may be made for the following words. This structure, which combines the functions of an analysis record and a prediction store, is basically a diverging tree with nodes which may themselves represent subtrees. At the start of the sentence the structure consists of a single node which represents the root of the main tree.

For each word of the sentence the "open ends" of the structure (i.e. the latest active nodes so far added on each analysis path) are traversed, and their possible continuations are tested. In the simple case, the immediate predictions which may be made from a node are found by means of a reference to an element in the finite-state network; the possible continuations of the path are indicated by the successors of this element in the network—or rather, a subset of them determined by the feature values associated with the analysis path.

For a prediction which is satisfiable directly by a lexical unit, the grammar element is matched against the codings assigned to the input word by the dictionary look-up procedure and if the match is successful a new (simple) node is added to the analysis path. In addition to identity of category, a successful match requires that the intersection of corresponding fields in the feature values for the grammar element and the input word coding should be nonnull and that a condition of feature concord—chiefly determined by the SRM—should hold between the overall feature values for the analysis path and the word coding.

In the case of a prediction realizable by a transform phrase, reference is made to a table to determine if it is the first prediction of the transform encountered at the current point in the sentence. If it is, a new tree is established with a root node which specifies the initial predictions for the transform and these predictions are in turn investigated. If none of the initial predictions is successful a failure indicator is set in the table; otherwise a node representing the new subtree is added to the analysis path from which the original prediction was made. Where the prediction of a transform is not the first encountered, it is necessary only to test the failure indicator in order to determine whether a new node (linked to the existing subtree) is to be added to the analysis path. In this way multiple predictions of the same type of transform at the same point in the sentence give rise to the establishment of a single subtree.

If a node represents a possible termination of the analysis of a transform, the higher level paths on which the prediction of the transform was encountered can be reactivated. The feature values computed for complete transforms are matched in the same way as word codings.

When the end of the sentence has been reached, all complete analyses are traced and printed out. If at any point before the end of the sentence no prediction remains—indicating that the grammar provides no analysis for the sentence (either because the grammar is incomplete or because the sentence is in fact ungrammatical)—the comment *no complete analyses* is printed out.

Programming Details

The analysis program is written in Atlas Autocode, a high level language belonging to the Algol family. The size of the program, including take-on procedures for the grammar and dictionary, is about 1000 source statements, compiling to some 10,000 machine instructions. The program runs at present on the ICL KDF9 computer of the Edinburgh Regional Computing Centre, which has 16K of 48-bit word core store. The operating system and compiled program occupy about 6K, leaving 10K for data space (including the representation of the grammar and the closed-class dictionary). No backing store is required.

Sentences to be analyzed are submitted in free format on paper tape and results are normally output on the line printer; when readability is more important than speed, results may be output on paper tape for reproduction on an off-line device with a more comprehensive character set. For each sentence a line is printed indicating how long it took to analyze the sentence (cpu time only), and how many nodes were required on the analysis structure.

Work is in progress to modify the program to take advantage of additional language facilities provided in IMP (an extension of Atlas Autocode). The new version will run on the multiaccess ICL 4/75 and the IBM 360/50, which are to replace the KDF9 at Edinburgh.

RECEIVED APRIL, 1969

REFERENCES

1. THORNE, J. P. A computer model for the perception of syntactic structure. *Proc. Roy. Soc. Series B 171* (1968), 377–386.
2. BRATLEY, P. AND DAKIN, J. A limited dictionary for syntactic analysis. In *Machine Intelligence 2*, Oliver and Boyd, Edinburgh, 1968, pp. 173–81.
3. CHOMSKY, N. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Mass., 1965.
4. THORNE, J. P., BRATLEY, P., AND DEWAR, H. The syntactic analysis of English by machine. In *Machine Intelligence 3*, Michie, D. (Ed.), American Elsevier, New York, 1968.