3. A PARSING PROCEDURE

M. KAY (UK)

A large family of strategies can be devised for a parsing procedure, in such a way that all immediate constituent structures allowed by a given grammar are developed quickly and easily. The one presented here has been chosen, because it enables the notion of presupposition to be fully exploited as a means of referring to the grammar. This works as follows: For each immediate-constituent rule of the form $A \rightarrow BC$, one of the elements on the right-hand side is chosen as presupposing the other. For example, we may say "C presupposes a preceding B to form A" and write

$$C | - B | A.$$  

Alternatively, we may write

$$B | + C | A.$$  

The plus and minus signs indicate whether the presupposed item comes to the left or the right. The choice of the presupposed item in a rule is made with a view to minimizing the greatest number of items presupposed by any one item.

The parsing procedure has the following salient characteristics:

1. A constitute consists of two and only two constituents.

2. All partial structures ending at a given word in the sentence are developed before the following word is examined.

In describing the procedure, we shall allow ourselves to say that a constitute $A$ precedes another constitute $B$ only if the first word of $B$ immediately follows the last word of $A$.

The first word of the sentence is read into the machine, and its $n$ grammar codes entered in the first $n$ spaces on the main data list. The entries are annotated to show that they are constitutes beginning with the first word of the sentence. A counter is associated with each word read into the machine, indicating the point in the list where the first grammar code for that word is stored. These counters, together with the word numbers in the data list, enable the items which precede a given word to be located very readily.

Also associated with each word is a prediction list. The prediction list for the first word of a sentence is always empty, and the list for the second word is completed as soon as the first word has been read into the machine. If the first word, or, in general any new item, presupposes a following item, a note of the relevant grammatical details is made on the prediction list of the following word. We are now ready to read the second word. Its $m$ grammar codes become the next $m$ entries in the data list. A new pointer is set and predictions are made as before. Parsing proper now begins. Each grammar code for the new word is considered in turn to see if it fills any of the predictions in the list for that word, and if it presupposes any of the items which precede it in the sentence! If either of these conditions is met, a new constitute is formed and an appropriate entry made in the next available space in the data list. The new entry is marked with the word number of the first word which it includes, and also with references to the two entries in the list which represent its constituents, all having the new word as last member.
These are now taken in turn as potential right-hand constituents of new constitutes. The process continues in this manner until all the items currently on the data list have been considered together with all the items which precede them, as candidates for a new constituent. Only then is a new word brought into the store.

Consider the sentence: “We are parsing sentences”. A very simplified grammar will enable us to develop two structures.

\[
\begin{align*}
\text{VERB} & \rightarrow \text{NOUN} \quad \text{PREDICATE} \\
\text{VERB} & \rightarrow \text{NOUN} \quad \text{SENTENCE} \\
\text{PRES. PARTICIPLE} & \rightarrow \text{AUXILIARY VERB} \\
\text{PREDICATE} & \rightarrow \text{NOUN} \quad \text{SENTENCE} \\
\text{ADJECTIVE} & \rightarrow \text{NOUN} \quad \text{NOUN}
\end{align*}
\]

At the end of the procedure, the data list might appear as follows as in Table 1.

<table>
<thead>
<tr>
<th>No</th>
<th>WORD</th>
<th>GRAMMAR CODE</th>
<th>WORD NUMBER</th>
<th>FIRST CONSTIT.</th>
<th>SECOND CONSTIT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WE</td>
<td>NOUN</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>ARE</td>
<td>AUXILIARY</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>SENTENCE</td>
<td>VERB</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>PARSING</td>
<td>PRES. PART.</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>ADJECTIVE</td>
<td>VERB</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>NOUN</td>
<td>VERB</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>VERB</td>
<td>NOUN</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>PREDICATE</td>
<td>SENTENCE</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>SENTENCE</td>
<td>PREDICATE</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>SENTENCES</td>
<td>VERB</td>
<td>2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>NOUN</td>
<td>PREDICATE</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>SENTENCE</td>
<td>NOUN</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>16</td>
<td>SENTENCE</td>
<td>NOUN</td>
<td>1</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>SENTENCE</td>
<td>NOUN</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

The results happen to be the last two entries on the list. The complete trees can be traced out using the last two columns.

**DISCUSSION**

R. M. NEEDHAM (UK). Since it is easy to devise many procedures, and individual ones present little novelty, how does one choose a good one?

M. KAY (UK). I wish I could give a good general answer to this question. Whatever reasons one may find are unlikely to be more than superficial. However the following points may be of interest.
1. Presupposition is only useful in a few procedures.
2. It is often difficult to identify the end of a sentence, since the period is not used only for this purpose. A procedure, such as the one described, which identifies everything up to a given point in the sentence may be useful, in that it can be discontinued only when a suitable result occurs together with appropriate punctuation. In other words, the length of the sentence need not be known initially.
3. It might be useful to discover alternative sentence structures in order of their increasing depth, as defined by Yngve. However, this can readily be computed in the process of working down the tree.

All these procedures involve nested cycles of instructions and it is worth considering if any one is particularly suited for use on a given machine.

4. **THE PROGRAMMING OF GRAMMAR**

H. SCHNELLE (Germany)

I shall describe a dynamic interpretation of a given grammar quite different from the interpretations currently in use. In the latter, the rules which constitute the grammar are interpreted as rewriting rules for given sequences of symbols. In contrast to this, it will be shown how syntactic structures, formulated in a context-free constituent-structure grammar, can be activated directly and automatically. Among the several advantages this programming system offers, I want to stress the fact that this procedure activates the grammar both for the generation of sentences and for recognition of their structure.

From the programming point of view our system can be characterized as a generalized bidirectional list processor (which may contain recursion cycles). The list can be obtained from a constituent-structure grammar, e.g. in the Chomsky regular form), by an automatic procedure. Since such a grammar can be viewed as a unidirectional list, each rule indicating the two “subsequent” list members (i.e. rules), in addition to the (operator-)relation between them (e.g. \( P = O + Q \) or \( N = P + Q \) or \( N = P, Q \)); and since it is essential for our program to find the way back in the list, we make the list bidirectional by indication of the “previous” list members and the alternatives in this direction as well. Moreover, places or programs which specify the choices at alternative rules should be made explicit. The following is an example of such a transformation.

The grammar for the (English) object clause is the following list of rules:

\[
\begin{align*}
\text{OBJ} & = \text{POBJ, DOBJ} \\
\text{POBJ} & = \text{Prep + DOBJ} \\
\text{DOBJ} & = \text{Art + Subst}
\end{align*}
\]

The corresponding bidirectional list is shown in Table 1.

<table>
<thead>
<tr>
<th>Number of rule</th>
<th>rule name</th>
<th>Number of position for activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OBJ</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>POBJ</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>DOBJ</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>DOBJ</td>
<td>4</td>
</tr>
</tbody>
</table>

Obviously the 3rd rule has been added to indicate that DOBJ can be reached both from OBJ and from
The separability of parsing procedures from grammars is now well known. Equally well known is the fact that parsing procedures can be written with a variety of purposes: to find a single structure (if any) for each sentence, to find a few structures, or to find all structures allowed by the grammar. This note merely calls attention to the separability of parsing procedures from grammars.

5. EXHAUSTIVE PARSING PROCEDURES *

D. G. HAYS (USA)

Zero, one, or more alternative branches are open, and exactly one must be chosen. When a point with zero paths is reached, either the sentence has been parsed or the sequence of decisions already made cannot lead to a complete parsing. In the first case, the completed structure is output. In either case, the next branch (in arbitrary but definite order) at the latest decision point still containing an open branch is followed, leading to another dead end or another complete structure. The parsing is complete when the decision tree has been completely traced.

In a parallel plan, all branches of a decision tree are followed simultaneously. With present equipment, this can only mean that all branches at a given decision point are considered before passing to the next. For each branch, a separate result is stored.

In a morsel plan, fragments are put together in such an order that the n-th fragment to be attempted is necessarily constructable as some combination of a selection of the n-1 fragments already constructed.

Consider as examples, a backtrack plan with predictive grammar, a parallel plan with immediate constituents, and a morsel plan with dependencies.
M. PIVOVONSKY (USA). I agree that high quality machine translation is not feasible, but I believe that translation of better quality than that achieved now, is feasible. The problems are: What is the upper limit of the quality attainable, and can this question be asked without any further experimentation?

A. KENT (USA). I, too, agree that high quality translation is not feasible. However, one of the chief purposes of translation is the reasonable communication of the contents of foreign language records to specialists, and this may be accomplished by providing the specialist with a low-quality mechanical translation. This
will allow identification of an interesting document from a collection of documents, and identification of interesting portions of the document for "precise" translation by humans.

Y. BAR-HILLEL (Israel). Whether low-quality machine translation is of any use, is for others to decide. I tend to believe, that in this particular game, humans will be able to beat a translation machine in all foreseeable time. Though I agree that the exact place of the upper limit of the quality attainable has not yet been determined, I think that we know enough to state unqualifiedly that this limit is too low for almost all practical uses.

H. P. GLOCKMANN (Germany) and J. VAN HORN (USA). What do you propose as a means of solving the tremendous translation problem now existing for the transmission of information, particularly of results of scientific and technical research, generated in various languages?

Y. BAR-HILLEL (Israel). I do not think that the translation of scientific material today poses a worse problem than in previous times. I dealt with this question on another occasion and do not want to repeat my arguments here.

A. CARACCIOLO (Italy). I should ask Pr. Bar-Hillel whether he thinks that human translation is possible, at least in general. Natural languages make it quite easy to formulate contradictory sentences. How should such a sentence be translated?

Y. BAR-HILLEL (Israel). The translation of a contradictory sentence should, of course, be a contradictory sentence in the target language. Under certain conditions the translator might wish to add a *sic*.

Human translation is possible, and many people make a living out of it. What you probably intended to ask was rather whether humans can provide "perfect" translation. I would however prefer not to draw this red herring of "perfect" translation into our discussion. The standard of comparison is, of course, just what one would ordinarily call high-quality translation.

HELEN BROWNSON (USA). The references to a time period of thirty years and to the expenditure of tens of millions of dollars are misleading. Although a conference was held ten years ago, the oldest research project in the field is only about six years old, and only a few million dollars have been spent on research. My point is that the field is still in its infancy, and the contention that we should admit defeat seems just as premature as claims that the problem is virtually solved. Do you have any evidence that we will not learn how to program machines to handle the complexities, as we learn more and more about these complexities?

Y. BAR-HILLEL (Israel). Though the references might mislead some people, they are still correct. Around 1933 the Russian, Smirnov-Troyanski, and the Frenchman, Artsouni, made some quite detailed proposals for mechanical devices for translation.

I had no intention of implying that research time and money were evenly distributed during the periods mentioned. Though it is surely a matter of judgment when to admit defeat in the pursuit of a certain aim, I believe that the time has arrived when the responsibility for making such a decision in the field of mechanical translation should no longer be postponed.

P. STONE (USA). The work of such psychologists as Piaget and Bruner has shown that the detailed strategies of human cognitive functioning can be revealed by quite simple experimental techniques. Have such techniques been employed to study the strategies used by humans in translation? It would seem that such a study of human functioning would be a basic initial step towards building later mechanical models.

Y. BAR-HILLEL (Israel). What Piaget and Bruner have shown is only that some detailed strategies of human cognitive functioning can be revealed by such techniques. To my knowledge, no such techniques have been employed to study the strategies used by human translators, though it was the first idea I had when I started working on mechanical translation eleven years ago. I would agree that a study of human translation might well be a necessary preliminary step towards the construction of mechanical models.

A. SESTIER (France). I suggest that nobody knows how to measure objectively and accurately the quality of any translation chosen at random. Failing which, everybody tries to have his own scale of evaluation, and our discussion has reflected primarily the disagreement of these scales. I suspect that only evaluations based on the satisfaction of users can be valid, but human and mechanical translation have almost disjoint sets of advantages and drawbacks, and this situation will undergo no substantial change in the foreseeable future. Therefore, as long as this situation lasts, their purposes must be different and the same scale of evaluation cannot apply to both.