GLP: A GENERAL LINGUISTIC PROCESSOR

G. Goerz

Univ. Erlangen-Nuemberg, RRZE, Martensstr. 1
D-8520 Erlangen, W. Germany

Abstract

GLP is a general linguistic processor for the analysis and generation of natural language. It is part of a speech understanding system currently under development at the Computer Science Department of our university [2].

1. The Structure of GLP

GLP is based on a second generation version of the General Syntactic Processor GSP of Kaplan and Kay [3]. Its architecture is shown in fig. 1.

GLP uses two central data structures, chart and agenda. The chart is a directed graph which represents the utterance being analysed (or generated) together with all its component structures for any point of time of the processor's operation. For the sake of simplicity our illustration of the chart's usage is limited to the simpler case of text processing. In this case the chart is initialized by a sequence of vertices which mark the start and the end of the sentence and the boundaries between words. The vertices are connected by edges which are labelled by the words themselves and lexical information (see fig. 2.). During processing GLP introduces more and more edges into the chart representing constituents, partial derivations, etc. Processing is finished when at least one spanning edge from the first to the last vertex is found which represents a completely specified interpretation of the sentence (see fig. 3). All edges along one path through the chart belong to the same decomposition of the sentence. Besides these inactive edges during processing there are also active ones which represent only a part of a phrase, together with an indication which kind of information would be necessary to complete it, i.e. to make an inactive edge out of it.

The agenda is a list of tasks to be carried out over the chart. Each task is the procedural incarnation of a rule of the grammer, or a part of it. As GLP realizes a multiprocessing scheme all tasks can be executed independently of one another as asynchronous parallel processes.

The underlying grammar is a procedural one similar to an Augmented Transition Network (ATN). Its rules contain linguistically defined operators, among which are operators for the formation of structures, selectors for accessing structures, predicates for testing the applicability of a rule or of parts of it, operators to cause side effects and to affect the flow of control. The formalism used is similar to that of Kay's[3] reversible grammar system in which the rules are treated as coroutines.

The whole processing is controlled by a monitor which is responsible for the initialization of the system and the generation and management of processes. It is the monitor which creates new tasks, splits complex tasks into potentially parallel executable subtasks, and maintains process and state information. The tasks themselves are executed by an interpreter (or processor) whose instruction set is the set of grammatical operators. Whenever a task sends an interrupt the monitor has to update the chart with the information sent along with it and to look at inactive edges whether there are suspended processes which would need exactly this information to be resumed. The monitor causes the selection of tasks from the agenda by means of a selector in an order determined by strategical reasons which are based on linguistic theory. The parsing strategy is realized by a scheduler which gives priorities to tasks and so fixes their order of execution. Thus the strategy may be flexible over the whole parsing process; top-down or bottom-up processing are not characteristic for the processing as a whole but only for parts of it.

Clearly the structure of GLP does not limit its ability to perform syntactic analysis; with a suitable linguistic data base (lexicon and grammar) it can be applied from phonological/morphological to semantic processing. The desire to break down the traditional borders between morphological, syntactic and semantic processing was not the least reason to choose this kind of systems architecture. Particularly in systems for processing continuous speech, because of uncertain data, progress in one level of analysis can only be achieved by confirmations from different levels so that a common data structure for all levels so that a common data structure for all levels of processing and a unique control structure allowing a flexible strategy become essential

2. Special Features of GLP

Besides an improved set of grammatical operators, GLP is able

- to perform direction-independent island parsing,
- to deal with gaps in the input utterance and to handle quality scores for word and phrase hypotheses, and
- to tie syntactic and semantic analysis closely together.

In order to process input data containing gaps and errors as is the case in speech understanding, GLP is able to start its operation at any point within the chart at a word hypothesis with a high quality score and continues by trying to expand this island on both sides by making predictions on words and word categories according to the grammar. Predictions are based on the entire context of the island as well as on its internal syntactic structure. If there is more than one island GLP tries to find appropriate syntactic hypotheses in order to merge the islands. This is achieved automatically by processing the tasks which seek to complete inactive edges in the chart. To perform these steps efficiently, the grammar has been preprocessed in a simple way to provide information about the word categories of potential predecessors or successors of the already recognized islands. Parsing from right to left is performed by a new class of tasks, the so-called scantasks, which look for a grammatically fitting expansion of the island determined by the information about it represented in the chart. To be precise, a scan-task tries to identify a starting node for a fitting category left of the island which is then verified by the corresponding syntax task.

The chart is initialized in such a way that it contains about ten (eventually overlapping) robust word hypotheses. The gaps between them are closed by specially labelled edges. If a GAP-edge is found while attempting to expand an island, GLP starts processes which try a match with a set of permissible sequences of word categories of the word edges at its left and right end which are accessible through the preprocessed grammar. If the match is successful, a task is generated automatically which requests word hypotheses with the desired features.

Word hypotheses carry quality scores from which priority scores are derived which in turn influence the scoring of the constituents containing them. The scoring methods we are currently experimenting with are essentially Woods' [7] shortfall and density scoring. In this fashion the scores determine the priorities of the tasks which shall work on these edges which in turn are subject to the parsing strategy obeying a "best-first" discipline.

To achieve an overall adaptive behavior leading to an acceptable recognition rate it is necessary to tie syntactic and semantic analysis closely together. Similar to the approaches by Miller [5], R. Bobrow [1] and Woods [7] in our system we provide the following processing stages: First the utterance is analyzed bottom-up in order to expand islands to local constituents. At this stage, the phase of semantic clustering is started by generating semantics-tasks which use semantic relationships in the form of case-/valence-frames from the lexicon, in which the semantic and pragmatic knowledge resides, mainly associated with the head word of a phrase. These tasks apply semantic interpretation rules from the grammar to generate semantic hypotheses at the phrase level. These are represented by edges to which instantiated case-frames are attached. In a third stage these hypotheses are evaluated syntactically in a top-down fashion which may cause the generation of new syntactic hypotheses by means of corresponding tasks or may give new scores to syntax-tasks which have been suspended in the appropriate portion of the chart. From the strategic point of view this means assigning new priorities to the newly generated or still present but not yet finished tasks. In this way constituents are interpreted as soon as they are parsed, and the structure of the semantic interpretations thus produced is checked when filling the case-/valence frames. This organization permits the semantic and pragmatic knowledge to control the syntactic analysis tightly, while at the same time syntactic and semantic processing are cleanly separated.

GLP tries to clarify remaining areas of uncertainty, which can only be resolved by means of contextual inference, e.g. by resolving references, by generating pragmatics-tasks which trigger the inference-processor. For this we chose the FRL-system [6], which offers a unique and - for our purpose - sufficient knowledge representation formalism and reasoning framework.

3. References

[1] Bobrow R.
"The RUS System". BBN-Report No.3878, Cambridge, MA, 1978, 28-58

[2] Hein H.-W.
"Automatic Understanding of Continuous German Speech". In Niemann H. (Ed.), Work on Pattern Recognition at Lehrstuhl f. Informatik 5 - Mustererkennung. Arbeitsberichte d. Instituts f. Mathematische Maschinen und Datenverarbeitung (Informatik), Erlangen 1981

[3] Kay M.
"Syntactic Processing and the Functional Sentence Perspective". Proc. TINLAP-1, Cambridge, MA, 1975, 6-9

[4] Kay M.
"Syntactic Processing". Proc. 17th Ann. Meeting ACL, La Jolla, 1979, T=2

[5] Miller P.
"An Adaptive Natural Language System that Listens, Asks and Learns"., Proc. IJCAI-75, Tbilissi, 1975, 406-413

[6] Roberts R.B., Goldstein I.P.
"The FRL Manual". MIT-AI-Memo 409, Cambridge, MA, 1977

[7] Woods W.
"Theory Formation and Control in a Speech Understanding System with Extrapolations towards Vision". In
Hanson, Riseman (Eds.), Computer Vision Systems, New York: Acad. Press, 1978, 379-390

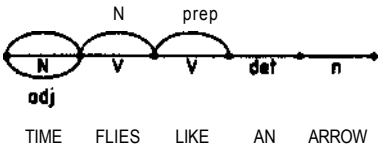

Fig. 2 Initial Chart

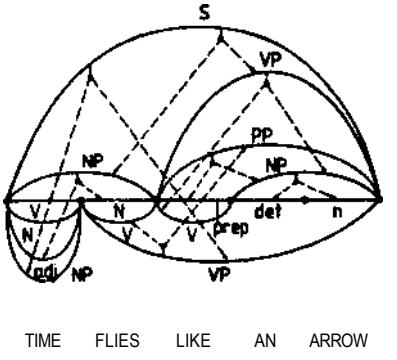

Fig. 3 Final Chart

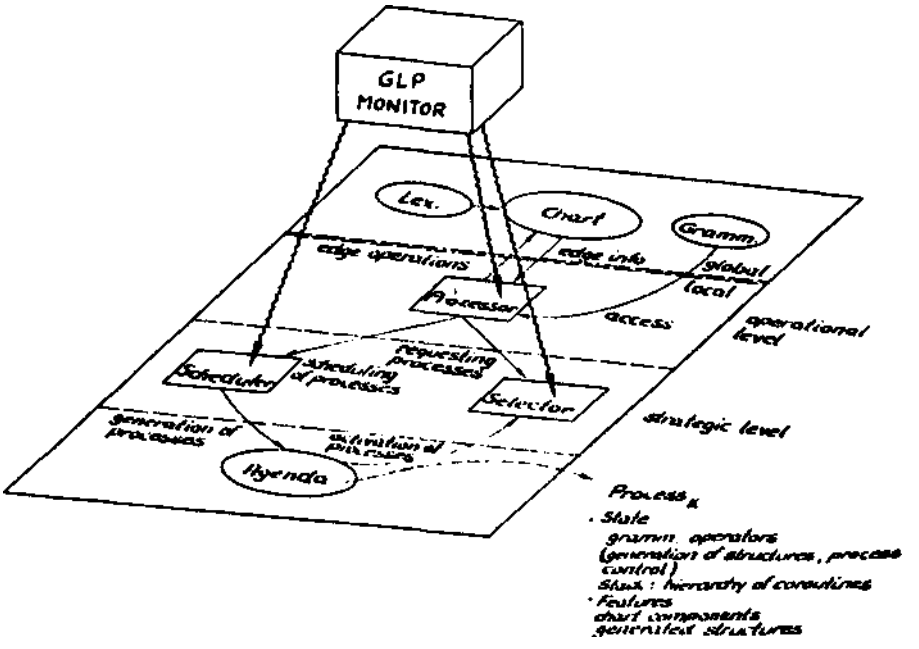

Fig. 1 The Structure of GLP