

## How to Parse Gaps in Spoken Utterances

G. Goerz, C. Beckstein  
Univ. Erlangen-Nuernberg, RRZE  
Martensstr. 1, D-8520 Erlangen, W. Germany  
Phone: (09131) 85-7031, Network: Goerz@SUMEX

### ABSTRACT

We describe GLP, a chart parser that will be used as a SYNTAX module of the Erlangen Speech Understanding System. GLP realizes an agenda-based multiprocessing scheme, which allows easily to apply various parsing strategies in a transparent way. We discuss which features have been incorporated into the parser in order to process speech data, in particular the ability to perform direction independent island parsing, to handle gaps in the utterance and its hypothesis scoring scheme.

#### I. GLP, A GENERAL LINGUISTIC PROCESSOR

GLP (Goerz 1981, 1982a,b) is a multi-strategy chart-parser, which has special features for the analysis of fragmentary and defective input data as it is the case with speech. GLP, a descendant of a version of GSP by M. Kay (1975), has been implemented in InterLISP. It can be used as a stand-alone system, to e.g. perform experiments, test various parsing strategies, or assist in the development of a linguistic data base. While for this purpose it got a cooperative, user-friendly interface, we also implemented an interface to the Erlangen Speech System (Niemann 1982). The Speech System's architecture is similar to that of HEARSAY-II, so that it employs a variety of knowledge sources, among which are modules for phonological, syntactic, semantic and pragmatic analysis. Although the structure of GLP does not limit its ability to perform syntactic analysis only - it is suitable for morphological or the non-inferential part of semantic analysis as well (see the similar system UCP, Sagvall-Hein (1982)) -, its role in the Speech System is constrained to the first mentioned task.

#### II. THE ARCHITECTURE OF GLP AND ITS EXTENSIONS FOR SPEECH ANALYSIS

The chart parsing idea was originally conceived and further developed by Martin Kay (1980). Its basic design extends the Well Formed Substring Table, a device used in many parsers to store intermediary re-

sults, which is represented as a directed graph, and makes it into an active parsing agent. Initially, the chart is set up as a set of vertices which mark beginning and end of an utterance and the boundaries between words. The vertices are connected by (inactive) edges which carry the lexical information of the resp. words. Whenever a constituent is found during the parsing process, a new inactive edge is added to the chart. In contrast to that, active edges represent incomplete constituents; they indicate an intermediate state in the search for a phrase. Using this data structure, GLP simulates internally a multiprocessing scheme by means of agendas. An agenda is a list of tasks to be carried out over the chart. Tasks are processing steps of different kinds, e.g. genuine analysis processes (Syntax- and Scan-Tasks), input/output with the outside world (Listen- and Talk-Tasks), and supervision to govern the analysis process in the large. In order to achieve a clear modularization, GLP is currently employing three agendas: Main for Syntax- and Scan-Tasks, Communication for Listen- and Talk-Tasks, and Control for Supervisor-Tasks. Whenever edges are added to the chart, any new tasks that can be created as a result, are scheduled on an agenda. The selection of tasks from an agenda is performed by its selector, which can, in the extreme cases, either perform a depth-first (agenda as a stack) or a breadth-first (agenda as a queue) search strategy. The question of the rule invocation strategy (or parsing strategy) is independent of the choice of the search strategy. Different parsing strategies such as top-down or bottom-up are reflected in different conditions for the introduction of empty active edges. An empty edge represents the task to search a constituent; it points to the same vertex where it is emerging from, indicating the search direction.

Scheduling of tasks on an agenda is performed by its scheduler which assigns priorities to tasks. GLP's operation in general is controlled by Supervisor-Tasks on the Control agenda, while the other tasks are executed by specific processors (interpreters).

The overall control mechanism is embedded in a general interrupt system. Interrupts are caused when the Main agenda - or even a particular task - is done or when the currently available resources are used up, in particular time and number of tasks. Whenever an interrupt occurs, the currently active task is finished and control is passed to the selector of the Control agenda. Then and only then input/output operations can be performed, new resources can be assigned, and GLP's strategy can be changed (see IV).

We do not claim any psycholinguistic validity for this kind of system architecture, although M. Kay (1980) argues that an agenda-based model may lead to significant insights in cognitive psychology.

### III. SCORING

In general, there are two parts of the problem of syntactic and semantic analysis: Judgment or decision (whether a given string is grammatical or not) and representation or interpretation (to decide how the pieces of the utterance fit together and what they mean). In a speech understanding system, hypotheses in all levels of abstraction carry quality scores, which play an important role in the overall strategy of the system. GLP receives word hypotheses from the Speech System's blackboard, which have been produced by the word hypothesizer, inserts appropriate word edges into its chart, extracts their quality scores and attaches derived priority scores to the resp. edges as features. If gaps in the utterance are recognized (i.e. there are no word hypotheses in a certain time interval with a score larger than a given threshold value), edges are introduced which are marked with the universal category GAP and a score feature which has the threshold as its value.

During parsing, GLP assigns scores to phrases. We are currently developing an explicit focussing strategy which is similar to Woods' (1982) Shortfall Scoring method. This method assigns priorities to partial interpretations, the so called islands, by comparing the actual score for an island with the maximum attainable score for the time period covered by the island and adding to it the maximum attainable scores for its environment. It can be shown that this priority scheme guarantees the discovery of the best matching interpretation of the utterance. In the special case of a GAP edge, a task is scheduled automatically looking for matching word hypotheses which have possibly been generated in the meantime. With each attempt to find a matching word hypothesis the GAP edges' score is reduced by a certain percentage

until it falls below a second threshold. In this case of a failure GLP constructs an incomplete phrase hypothesis out of the available information including the pattern which characterizes the missing word(s). In addition, while building phrase hypotheses, GLP can also take into consideration preference scores (or weights) for different branches in the grammar, but our grammar does not employ this feature at the present time.

### IV. INCREMENTAL PARSING

Incremental parsing is a salient feature of GLP. There is no distinct setup phase; GLP starts to work as soon as it receives the first (some ten) word hypotheses with a sufficient quality score. Whenever an interrupt occurs, new word hypotheses can be incorporated into the chart. These hypotheses are provided by the Speech System's word hypothesizer, either continuously or as an answer to a request by GLP, resulting from gap processing, that has the form of an incomplete word hypothesis which is to be filled. In the latter case active edges act as demons waiting for new information to be imbedded in already generated partial structures in such a way that no duplicate analysis has to be performed. Since the Speech System's overall strategy can decide when new word hypotheses are delivered, a data-driven influence on GLP's local strategy is achieved.

The required input/output processes for hypotheses are performed by Listen- and Talk-Tasks, which are activated by the selector attached to the Communication agenda. The Communication selector is triggered by interrupt conditions, which are due to the mentioned overall parsing strategy. The communication channel to the outside world can be parameterized by a general feature, the Wait list. Whenever the name of a processor, e.g. Listen or Talk, is put on the Wait list, this processor is blocked until it is removed from the Wait list. Because blocking of any processor causes a redistribution of the available resources, it effects in consequence GLP's strategy. Direct influence on the parsing strategy is achieved by temporarily blocking the Syntax or Scan processors. Furthermore, the strategy can be modified explicitly by attaching a new selector to the Main agenda and by setting Various global strategy parameters. These include threshold values, e.g. for gap processing, as well as limits for resources, the most important of which is time. This flexibility in strategy variation is important for an empirical evaluation of our approach. Although we have not yet analyzed GLP's parsing complexity in general, some limiting factors for chart parsing are well known by investigations on

the context free case by Sheil (1976): The number of steps is of  $O(n^3)$ , the space requirements of  $O(n^2)$  independent of the parsing strategy, where  $n$  is the length of the input sentence. The size of the grammar does not influence complexity, but its branching factor, which is a measure for its degree of nondeterminism, acts as a proportionality factor.

#### V. ISLAND PARSING WITH A CHART

In the following we like to point out why we think that GLP's mechanism has several advantages over traditional island parsing schemes (e.g. Woods 1976). In order to process defective input data, the parser must be able to start its operation at any point within the chart. In general, our main parsing direction is from left to right. With respect to the expansion of islands, in particular from right to left, our mechanism is simpler, because, for example, there is no explicit representation of paths. For Syntax-Tasks, which are proceeding in the usual way from left to right, this information is already attached to their corresponding active edges. Scan-Tasks, which are seeking to the left of the island, access information attached to the vertex they are starting from. Phrase hypotheses are only generated by Syntax-Tasks; if an island cannot be expanded to the right, a Scan-Task which seeks an anchor point for an active edge to the left of the island is scheduled automatically. While in the usual island parsing schemes the focus of attention is not shifted left of an island before appropriate hypotheses are generated, (e.g. if there is a gap - of arbitrary duration - left of the island), GLP seeks for an anchor point, attaches an active edge to it and schedules a corresponding Syntax-Task. This task will then and only then generate a phrase hypothesis. Furthermore, we think that our scheme is combinatorially more efficient, because fewer hypotheses are generated. This fact results from a more adequate representation of an island's left context: In usual island parsing expansions to the left are performed without regarding the left context of the island as long as only predictions exist and no hypotheses are available.

The goal of the parsing strategy we are developing now is that semantic analysis at the constituent level can be started as soon as a local constituent is syntactically recognized (bottom-up). The resulting semantic hypotheses, produced by the SEMANTICS module and delivered through the Speech System's blackboard, which contain semantically based predictions, are then matched against the chart. This process will lead to the generation of new tasks, which in turn may produce new word and

phrase hypotheses, so that present islands can be expanded and merged.

#### VI. ACKNOWLEDGEMENTS

Thanks to Prof. G. Nees, who continuously encouraged us in our work on GLP, and to Prof. K.M. Colby, Roger Parkison and Dan Christinaz of the Neuropsychiatric Institute, UCLA, where the first author learnt a lot on robust parsing during a research stay sponsored by the German Academic Exchange Service (DAAD).

#### VII. REFERENCES

- Goerz G. (1981): GLP: A General Linguistic Processor. Proc. IJCAI-81, Vancouver, B.C., 1981, 429-431
- Goerz G. (1982a): GLP: The Application of a Chart Parser to Speech Understanding. SIGART Newsletter No. 79, Jan. 1982, 52-53
- Goerz G. (1982b): Applying a Chart Parser to Speech Understanding. Proc. European A.I. Conference, Orsay, 1982,
- Kay M. (1975): Syntactic Processing and Functional Sentence Perspective. Proc. TINLAP-1, Cambridge, Mass., 1975, 6-9
- Kay M. (1980): Algorithm Schemata and Data Structures in Syntactic Processing. Xerox Report CSL-80-12, Palo Alto, Calif., 1980
- Niemann, H.: The Erlangen System for Recognition and Understanding of Continuous German Speech. In: Nehmer J. (Ed.): GI - 12. Jahrestagung, Berlin: Springer IFB-57, 1982, 330-348
- Sagvall-Hein A. (1982): An Experimental Parser. In: Horecky J. (Ed.): Proc. COLING-82, Prague, 1982, 121-126
- Sheil B. (1976): Observations on Context Free Parsing. Stat. Meth. in Linguistics 6, 1976, 71-109
- Woods W. (1976): Speech Understanding Systems, Final Report, Vol. IV. Syntax and Semantics. BBN Report 3438, Cambridge, Mass., 1976
- Woods W. (1982): Optimal Search Strategies for Speech Understanding Control. A.I. Journal 18, 1982, 295-326