Constraint Satisfaction for Robust Parsing of Spoken Language

Wolfgang Menzel
Fachbereich Informatik, Universität Hamburg
Vogt-Kölln-Straße 30, D-22527 Hamburg
e-mail: menzel@informatik.uni-hamburg.de

Abstract

The eliminative nature of Constraint Satisfaction over finite domains offers an interesting potential for robustness in the parsing of spoken language. An approach is presented, which puts unusually ambitious demands on the design of the Constraint Satisfaction procedure by trying to combine preferential reasoning, dynamic scheduling, parallel processing and incremental constraint solving within a coherent solution.

1 Introduction

Spoken language processing involves a number of novel requirements as compared to traditional techniques for the treatment of written text. Among those which deserve special attention are:

1. the frequent occurrence of deviant language in particular if utterances are produced spontaneously as part of a problem solving task. Spoken language is not only characterized by hesitations, breaks and restarts but is additionally narked by different kinds of syntactic and semantic inconsistencies. All these performance phenomena hardly disturb the language understanding capabilities of a human listener. On the contrary, very often they are absorbed unnoticed.

2. the fuzzy nature of the signal-to-symbol mapping for spoken utterances which is caused by very different factors including the nondeterministic nature of articulation, fast speech phenomena or dialectal and idiolectal variance. It makes speech recognition a difficult enterprise, characterized by an inherent uncertainty of recognition results. By no means this uncertainty can be avoided. At best its degree can be estimated and coded into a kind of confidence score.

3. the temporal characteristics of speech understanding capabilities including
   - the consideration of varying speech rates, which requires the processing components to adapt to external temporal constraints
   - an immediate response capability, necessary to take part in the usual kind of mixed-initiative dialogues
   - a parallel processing on various levels of language in order to integrate knowledge from quite different sources in a coherent fashion.

   These properties require the speech understanding system to work in an overall incremental manner analysing incoming portions of the signal with a delay as minimal as possible. At every time point, each component has to carry out its processing with the body of data available so far. Delayed decisions have to be avoided, because they may hinder the progress of other components. The input stream is always open to continuation, hence, global optimization criteria based on the existence of a definite endpoint are lost.

In order to cope with these requirements, a processing schema has to be defined which — within the framework of interactive and incremental analysis — is expected to exhibit a robust behaviour most notably against ungrammaticality, recognition errors and external temporal pressure. Robustness, however, is required to constitute a fundamental feature of the processing mechanism itself. Traditional approaches to the robust processing of written language (c.f. the overview in Stede (1992)), which rely on additional special purpose routines activated only in case of failure, are clearly inadequate. In the following it is argued that the application of Constraint Satisfaction techniques to the parsing of spoken language may well be suited to provide the desired kind of robust behaviour.

2 Parsing as Constraint Satisfaction

In order to analyse spoken language by means of Constraint Satisfaction techniques, parsing is considered a procedure of structural disambiguation (Maruyama, 1990). A relational model of syntactic and semantic structures (usually called Dependency Grammar) serves this purpose best. It yields a structural description for input utterances as an appropriate subset of $S = W \times W \times L$ where W denotes the set of word form tokens from the input utterance and L is a set of syntactic or semantic labels. In terms of the grammar, labels can be understood as binary relation symbols over pairs of word forms constituting the desired structural description of the input. They are usually interpreted hierarchically as the subordination of a dependent form under a dominating one.

Traditionally, tree structured descriptions are of particular interest in Natural Language Processing. They are obtained if dependents are always subordinated uniquely. Using this property, parsing can be transformed into a Constraint Satisfaction problem by simply identifying input word forms (the dependents) with variables of the constraint network. Values, accordingly, will be *pairs* consisting of a dominating form and a label.

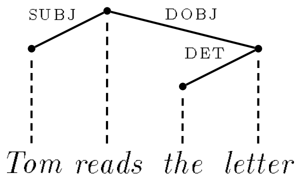

Figure 1: A syntactic dependency description

They are taken from the finite domain $W \times L$, and they always belong to a specific description level (i.e. syntax or semantics). Figure 2 shows a simple network of syntactic constraints prior to the application of linear ordering regularities. Dependency trees can be extracted from this compact representation, starting at the node linked to root and following reverse dependency relations down to the leaf nodes. The network of figure 2 comprises four different syntactic readings, since both, the subject-object distinction as well as the attachment of the determiner are still ambiguous.

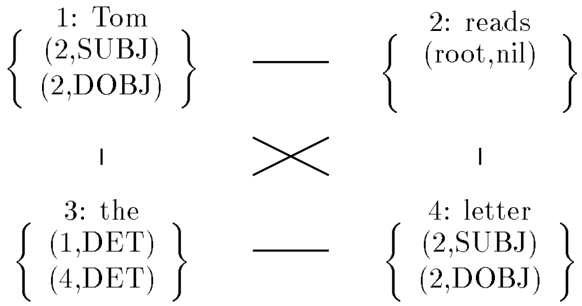

Figure 2: A constraint network for the sample input of Figure 1. Word forms are denoted by their position in the input string. Edges represent complex compatibility constraints between the dependency relations of corresponding value sets.

Admissible combinations of values are described by a set of constraints over subordination relations (the grammar). Every solution to the Constraint Satisfaction problem then represents a well-formed structural description of the input string under consideration.

Uncertainty resulting from speech recognition is usually captured by the introduction of alternative recognition hypotheses, which are partially ordered by their temporal relationship. Parsing is then to be carried out over a lattice of word form hypotheses. Every admissible representation has to satisfy both, the constraints of the grammar and the path condition, which requires all word forms included into a description to belong to a single spanning path in the lattice.[1]

Since parsing of natural language utterances almost always starts with considerable initial ambiguity of lexical entries, lattice parsing in fact is a parsing problem with extremely increased initial ambiguity.

[1]Actually, an optimal path with respect to the confidence scores from word recognition is required.

3 Why Constraint Satisfaction?

The traditional approach to the parsing of natural language is a constructive one. Partial trees as defined by the grammar are successively combined with each other until the complete input string has been successfully consumed. In contrast to this popular strategy, parsing as Constraint Satisfaction is a *candidate-elimination procedure*, which successively removes partial descriptions from the space of possible solutions. It always starts with a maximal degree of structural ambiguity (each word form can be subordinated under any other with any possible label) and tries to reduce it to a unique assignment. A more elaborate termination criterion will be required, however, if the treatment is extended to graph-structured representations. This kind of situation is quite common for semantic relations where e.g. referential expressions have to be considered.

From the eliminative nature of Constraint Satisfaction, several interesting properties follow, which constructive procedures definitely lack. First of all, a rudimentary notion of robustness is introduced into the parsing process without any additional effort. It can be obtained by a rather simple modification of the termination criterion from the usual network quiescence to a unique value-assignment condition: Constraint Propagation stops if the value set for each variable has been reduced to a singleton[2]. Then

1. any input sentence — including the clearly ungrammatical ones — will receive a complete but possibly counterintuitive structural interpretation. Note that even this extremely simple property is no matter of course with constructive procedures, where considerable additional effort is required in order to achieve a similar behaviour.

2. under favourable conditions, a completely disambiguated description can be obtained even if not all of the constraints have been considered yet. By chance, a corresponding constraint violation in the input sentence might pass through unnoticed.

Obviously, in both cases the achieved kind of robustness is of very limited use, since it highly depends on the accidental sequence of constraint applications. It offers, however, the possibility to introduce robustness in a more systematic way if it becomes possible to include a mechanism which is able to weight contradictory constraints against each other. Structural descriptions supported only by rather weak constraints can be discarded instead of simply ignoring constraints in an arbitrary manner.

A second important advantage of candidate-elimination procedures comes from the fact that they allow to measure the progress of computation by rather simple means, namely the current size of the value sets at certain variables of the Constraint Satisfaction Problem. In our parsing example, this corresponds to a local degree of remaining ambiguity. Such a measure not only provides the basis for a goal-oriented scheduling of disambiguation efforts (use the most effective constraints first!) but for the first time

[2]Lattice parsing, on the other hand, requires a suitable combination of this criterion with the path condition. Here, the majority of variables has to be removed from the Constraint Satisfaction problem by reducing the corresponding value sets to empty ones.

opens up the possibility to react sensibly to external temporal constraints (c.f. Menzel (1995)).

4 Robust Parsing

Robustness does not come for free. A reliable compensation for deviant input can only be achieved if the internal model information of the system provides for a certain degree of redundancy. For parsing, this redundancy can be introduced e.g. by a strict modularization into two almost independent description levels, namely a syntactic and a semantic one (Menzel, 1995). Both levels, are loosely coupled using mapping constraints. Thus, a symmetrical architecture results, which supports a completely autonomous processing on the distinct levels as well as a mutual compensation of information deficits in case of partial inconsistencies. Syntactic errors may be suppressed due to semantic support, whereas semantic deviations (c.f. the communication of novel concepts) can only be handled if sufficient syntactic structure is available.

A sample grammar consisting of syntactic, semantic and mapping constraints is shown in Figure 3. Variables (upper case `X` and `Y`) refer to subordination relations. `role` indicates the description level, `lab` the attached label. $\uparrow$ and $\downarrow$ point to the dominating and depending word form within the subordination relation, specifying the kind of information requested. Word forms are identified by their position index `pos`.

Preferential reasoning based on soft constraints is used throughout the system. It takes care of the fact that the majority of semantic constraints and even a vast body of syntactic information is defeasible. Constraints between the syntactic and semantic level are modelled as preferential mappings as well.

The preferential nature of grammatical knowledge is modelled by the assignment of penalty factors to constraints. Penalty factors may range from zero to one where

`pf=0`	specifies a strict constraint in the classical sense and
`0<pf<1`	indicates a soft constraint accepting contradictory cases with a confidence value proportional to `pf`

Obviously, a value of one is meaningless because it neutralizes the constraint. Penalty factors are combined multiplicatively, i.e. compatibility matrices within the constraint satisfaction problem no longer contain binary categories but *confidence scores* also ranging from zero (for impossible combinations) up to one (for combinations not even violating a single constraint).

The indirect treatment of preference by penalty factors offers a consistent extension to the basic paradigm of constraint satisfaction. It does not sacrifice the eliminative nature of constraints but simply softens it. Inappropriate readings are excluded only if they violate strict constraints. In all other cases, they are downgraded to a certain degree.

In particular, the penalty-based approach helps to tackle some normalization problems otherwise inherently connected with the constraint-satisfaction approach: Most modifying relations (or combinations of them) will pass a constraint simply because it is irrelevant

1. syntactic constraints

   sy1: $\texttt{X.role=syn} \wedge \texttt{X}\!\downarrow\!\texttt{cat=N} \rightarrow \texttt{X}\!\uparrow\!\texttt{cat=V} \wedge \texttt{X.lab}\in\{\texttt{SUBJ,OBJ}\}$

   A noun can modify a verb syntactically either as a subject or as a direct object.

   sy2: $\texttt{X.role=Y.role=syn} \wedge \texttt{X.lab=SUBJ} \rightarrow \texttt{X}\!\downarrow\!\texttt{num = Y}\!\uparrow\!\texttt{num}$

   A subject agrees with its modifiee with respect to number.

   sy3: $\texttt{X.role=syn} \wedge \texttt{X.lab=SUBJ} \rightarrow \texttt{X}\!\downarrow\!\texttt{pos < X}\!\uparrow\!\texttt{pos}$

   The subject precedes the finite verb.

   sy4: $\texttt{X.role=Y.role=syn} \wedge \texttt{X}\!\uparrow\!\texttt{pos=Y}\!\uparrow\!\texttt{pos} \rightarrow \texttt{X.lab} \neq \texttt{Y.lab}$

   A word form cannot be modified twice by the same relation.

2. semantic constraints

   se1: $\texttt{X.role=Y.role=sem} \wedge \texttt{X}\!\downarrow\!\texttt{cat=N} \rightarrow \texttt{X}\!\uparrow\!\texttt{cat=V} \wedge \texttt{X.lab}\in\{\texttt{AG,PAT}\}$

   A noun can modify a verb semantically either as as agent or as a patiens.

   se2: $\texttt{X.role=sem} \wedge \texttt{X}\!\uparrow\!\texttt{word=fressen} \wedge \texttt{X}\!\downarrow\!\texttt{prop=animal} \rightarrow \texttt{X.lab=AG}$

   Animals do eat.

   se3: $\texttt{X.role=sem} \wedge \texttt{X}\!\uparrow\!\texttt{word=fressen} \wedge \texttt{X}\!\downarrow\!\texttt{prop=plant} \rightarrow \texttt{X.lab=PAT}$

   Plants are to be eaten.

   se4: $\texttt{X.role=Y.role=sem} \wedge \texttt{X}\!\uparrow\!\texttt{pos=Y}\!\uparrow\!\texttt{pos} \rightarrow \texttt{X.lab} \neq \texttt{Y.lab}$

   A word form cannot be modified twice by the same relation.

3. mapping constraints

   ss1: $\texttt{X.role=syn} \wedge \texttt{Y.role=sem} \wedge \texttt{X}\!\uparrow\!\texttt{pos=Y}\!\uparrow\!\texttt{pos}$
   $\rightarrow (\texttt{X.lab=SUBJ} \leftrightarrow \texttt{Y.lab=AG})$

   The subject of a verb is always identical to its agens.

   ss2: $\texttt{X.role=syn} \wedge \texttt{Y.role=sem} \wedge \texttt{X}\!\uparrow\!\texttt{pos=Y}\!\uparrow\!\texttt{pos}$
   $\rightarrow (\texttt{X.lab=OBJ} \leftrightarrow \texttt{Y.lab=PAT})$

   The direct object of a verb is always identical to its patiens.

Figure 3: A sample grammar for a simple German disambiguation problem

syntax	semantics	mapping
pf(sy1)=0.0	pf(se1)=0.0	pf(ss1)=0.2
pf(sy2)=0.1	pf(se2)=0.1	pf(ss2)=0.3
pf(sy3)=0.3	pf(se3)=0.7	
pf(se4)=0.0	pf(se4)=0.5	

Table 1: A penalty factor assignment for the grammar in Figure 3.

for that particular configuration. An increase of goodness estimates for these cases would yield a highly undesirable, since unjustified reinforcement.

If desired, penalty factors may be computed dynamically. Such an extension can be used to model e.g. distance effects, where the penalty factor is lowered if the local distance between two forms in the input sentence increases (Menzel, 1994). Other applications for dynamic penalty factors include

- the incorporation of stochastic knowledge sources,
- a contextually influenced preference for certain syntactic constructions or
- a dynamic scheduling procedure for parsing under time pressure as described below.

Given the grammar of Figure 3, a distribution of penalty factors according to Table 1 has been assigned[3].

Constraint Propagation now has to be extended from the classical case of strictly binary decisions to the handling of confidence scores. The application of penalty-weighted constraints to a disambiguation problem then consists of two steps:

1. the calculation of initial confidence scores for all combinations of syntactic and semantic modification relations and
2. a selection procedure pruning the search space by sorting out unlikely interpretations

The selection procedure is based on an assessment function heuristically identifying subordination relations to be pruned. In order not to select promising hypotheses, the assessment procedure first of all should only take into account modification relations characterized by the following three criteria

- being close to the global minimum for all modification relations, combined with
- a maximal contrast with respect to alternative relations and
- a minimal contrast between all the confidence scores supporting the relation in question.

[3]This assignment was based on human judgements. Investigations are carried out, however, to directly derive penalty factors from tree-annotated corpus data.

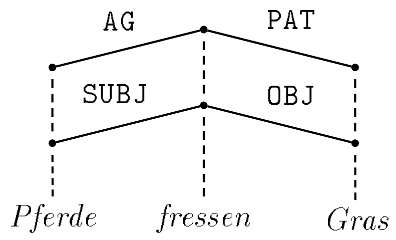

Figure 4: A two level description for a simple German sentence.

For experimental purposes, a selection procedure based on the sum of quadratic errors for setting scores to zero has been used. Hence, structural interpretations violating a high number of rather strong constraints are pruned first.

Using the toy grammar specified above together with its penalty scores, the arbitration process between syntactic and semantic evidence in simple disambiguation problems can be studied. Thus in a sentence like

Pferde fressen Gras.	*(Horses eat grass.)*	(1a)

both layers uniformly support the single interpretation shown in Figure 4.

Due to the strong semantic support, the interpretation remains unchanged if a marked ordering (topicalization of the direct object) is chosen (1b), an agreement error is introduced (1c) and both deviations are combined finally (1d).

Gras$_{PAT}$ *fressen Pferde*$_{AG}$.	(1b)
Pferd$_{AG}$ *fressen Gras*$_{PAT}$.	(1c)
Gras$_{PAT}$ *fressen Pferd*$_{AG}$.	(1d)

The interpretation is retained even if its semantic support is neutralized as in the following utterance, containing a twofold type shift.

Autos$_{AG}$ *fressen Geld*$_{PAT}$.	*(Cars eat money.)*	(2a)
Geld$_{PAT}$ *fressen Autos*$_{AG}$.		(2b)
Auto$_{AG}$ *fressen Geld*$_{PAT}$.		(2c)

It switches to the alternative interpretation only in the case of combined syntactic distortions

Geld$_{AG}$ *fressen Auto*$_{PAT}$. (2d)

Even for the counterintuitive example

Gräser$_{AG}$ *fressen Pferd*$_{PAT}$. (3a)

which, if desired, could be taken as an utterance, syntactic evidence will gain the upper hand against the violation of two selectional constraints. This interpretation, however,

	strong semantic support	missing semantic support	counterintuitive reading
wellformed	+	+	+
marked ordering	+	+	−
aggreement error	+	+	−
combined deviation	+	−	−

Table 2: Parsing results for gradually deviating input.

happens to be fragile and breaks immediately under arbitrary syntactic variation.

The results are summarized in Table 2, where a positive sign denotes that the desired result according to Figure 4 has been obtained. The distribution of positive results clearly demonstrates the compensatory effects between syntactic and semantic contributions.

5 Adaptive Parsing

In order to react sensibly to external time pressure, a system needs the ability of deliberation between the desired quality of its output results and the ressources allocated to carry out the computation. Since quality parameters for parsing results can hardly be measured directly, an indirect approach is taken, which estimates the quality level according to the quality of model information used to produce the output (Menzel, 1994). The procedure then is able to supply high-quality results if only constraints of strong certainty have been used. The quality level — and hopefully the time required for the computation — drops if more risky decisions are taken.

In contrast to the parsing schema presented so far, a kind of scheduling for constraint application becomes necessary, which selects among the available constraints to achieve an output quality as high as possible given the time yet available. Three different strategies of achieving the desired behaviour come to mind, two of them rely on the penalty factors already introduced. A rather simple technique (forced pruning) exploits a reinforcement strategy for the pruning procedure: Growing temporal pressure causes pruning to remove an increasing number of subordination possibilities at a time from the constraint network. Accordingly, increasingly hazardous pruning decisions will be taken.

A more elaborate version (dynamic penalty assignment) will make use of dynamic penalty factors as described above. It maintains a threshold for penalty factors $\mathtt{pf_{min}}$, which monotonically rises under growing time pressure. The actual size of the penalty factor is then computed according to the following condition:

$$\mathtt{pf}(c) = \begin{cases} \mathtt{pf}(c) & \text{if } \mathtt{pf}(c) > \mathtt{pf_{min}} \\ 0 & \text{if } \mathtt{pf}(c) \leq \mathtt{pf_{min}} \end{cases}$$

thereby tightening a defeasible constraint to a strict one which definitely excludes combinations of subordinating relations from the space of possible solutions. This approach, however, can be considered a straightforward solution only as long as the degree of temporal pressure can be estimated prior to constraint application itself. Under realistic conditions, scheduling will become a dynamic process which has to adjust the eliminative strength of constraints according to the actual requirements of the current disambiguation problem. Hence, dynamic penalty assignment will sometimes have to decide on a dedicated reapplication of constraints if the penalty threshold has been modified in the meantime. This requires a sophisticated memoization mechanism for constraint applications and therefore gives rise to a substantial overhead in the organization of the propagation procedure.

Finally, the idea of explicit scheduling can be pursued a little bit further, by including a reliability estimation for constraints. The procedure then maintains constraint sets with different degrees of reliability, which might range from very general regularities to brute-force criteria. Whilst the first group is usually used during early stages of the disambiguation procedure, the latter will be invoked only under extreme pressure (c.f. Menzel (1994)). Note that it is possible to maintain a classification of this kind completely independent from the penalty factors assigned to individual constraints. This enables the grammar writer to deliberately choose between controlling the balance between conflicting constraints on the one hand and determining the time point at which a particular group of constraints is considered at all.

6 Further Requirements: Incremental and Parallel Propagation

Incremental processing is a basic requirement for Spoken Language Systems, which is quite difficult to obtain. Since incoming word-form hypotheses have to be integrated immediately into the already existing structure, the constraint satisfaction problem will no longer remain a static one. The network is constantly expanded by new hypotheses, whereas completely disambiguated portions of the speech signal may be discarded after leaving a suitably adjusted time horizon.

Although algorithms for incremental constraint propagation have been published already (e.g. Freeman-Benson, Maloney, and Borning (1990)), these are restricted to domains with uniquely computable values. This restriction, however, makes them totally unsuitable for the task of structural disambiguation. Another difficulty results from the path condition in lattice parsing. Harper and Helzerman (1994) present an algorithm for constraint satisfaction over lattice structures, which, again, is non-incremental.

A very similar observation can be made for concurrent approaches to the parsing problem. Constraint satisfaction techniques bear a considerable potential for parallel execution on multiprocessor arrangements. This potential basically results from the strictly local scope of constraint application and filtering operations over the network. Pruning as described above, however, uses a global decision criterion. It therefore might contribute a serious bottleneck. Heizerman and Harper (1992) describe a massively parallel implementation based on a suitable problem transformation. Unfortunately, problem transformation is inherently restricted to static networks, whereas incremental constraint solving requires a dynamic solution to load balancing.

It therefore still remains to be examined under which conditions or whether at all these partial contradictory characteristics of an incremental *and* parallel approach to lattice parsing can be reconciled.

7 Conclusion

The application of Constraint Satisfaction techniques to the parsing of spoken language basically requires finite domain algorithms enhanced by a number of additional nonstandard features. This approach to parsing turns out to be a rather ambitious enterprise, pointing to a number of interesting and challenging areas for innovative solutions, which might even exhibit the potential to push Constraint Satisfaction technology ahead. In particular, possible solutions will need to support the following demands:

1. the propagation of soft constraints,
2. a dynamic scheduling of constraint applications
3. a parallel propagation of constraints
4. incremental Constraint Satisfaction

A coherent solution to these requirements will contribute significiantly to the robustness of spoken language systems against a variety of adverse factors including ungrammatical input, recognition uncertainty and external temporal constraints. None of the above-mentioned decision strategies has been evaluated to a suffient degree yet. Therefore, an environment for extensive experimentation is under development. Currently investigations focus on the design of the semantic layer (including an interface to customary knowledge-based systems for domain specific concepts) and the exploitation of stochastic information as it can be elicited from spoken language corpora.

References

Freeman-Benson, B., J. Maloney, and A. Borning. 1990. An incremental constraint solver. *Communications of the ACM*, 33(1):54–63.

Harper, Mary P. and Randall A. Helzerman. 1994. Managing multiple knowledge sources in constraint-based parsing of spoken language. Technical Report EE-94-16, School of Electrical Engineering, Purdue University, West Lafayette.

Heizerman, R. A. and M. P. Harper. 1992. Log time parsing the MasPar MP-1. In *Proceedings of the 6th International Conference on Parallel Processing,*, pages 209–217.

Maruyama, H. 1990. Structural disambiguation with constraint propagation. In *Proceedings 28th Annual Meeting of the ACL*, pages 31–38.

Menzel, Wolfgang. 1994. Parsing of spoken language under time constraints. In A. Cohn, editor, *Proceedings 11th European Conference on Artificial Intelligence*, pages 560–564, Amsterdam.

Menzel, Wolfgang. 1995. Robust parsing of natural language. In *KI-95: Advances in Artificial Intelligence*, pages 19–34, Berlin. Springer-Verlag.

Stede, Manfred. 1992. The search for robustness in natural language understanding. *Artificial Intelligence Review*, 6(4):383–414.