Constraint Satisfaction for Robust Parsing of Spoken Language

Wolfgang Menzel

Fachbereich Informatik, Universität Hamburg

Vogt-Kölln-Straße 30, D-22527 Hamburg

telephone: (49 40) 54 94 - 24 35

telefax: (49 40) 54 94 - 25 15

e-mail: menzel@informatik.uni-hamburg.de

short title: Robust Parsing of Spoken Language

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 marked by different kinds of syntactic and semantic inconsistencies. Most surprisingly, these performance phenomena hardly disturb the language understanding capabilities of a human listener. 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. This uncertainty cannot 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 and
- the 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 as minimal a delay as possible. At every time point, each component has to carry out its processing with the body of data available so far. Delayed decisions must be avoided, as 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 treat robustness as an add-on feature, are clearly inadequate. Usually a special purpose routine is activated in case of failure and therefore requires that an erroneous situation be detected before an appropriate treatment can be attempted. There is obviously no way to explain the impressive ability of a human reader/listener to ignore unusual constructions, especially when temporal pressure

or strong expectations are involved.

The following argument states that the application of Constraint Satisfaction techniques to the parsing of spoken language may well provide the desired behaviour. This covers both a robust treatment against unexpected input as well as a graceful degradation under time pressure, closely linking these two notions of robustness.

2 Parsing as Constraint Satisfaction

In order to analyse spoken language by means of Constraint Satisfaction techniques, we define parsing as 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. These relations are usually interpreted hierarchically as the subordination of a dependent form under a dominating one.

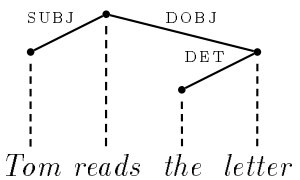

Figure 1: A syntactic dependency description

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 will be *pairs* consisting of a dominating form and a label. 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. Word forms are denoted by their position in the input string. Edges represent complex compatibility constraints between the dependency relations in the value sets. Dependency trees can be extracted from this compact representation, starting at the node linked to root (in our example only the one associated to the input form *reads*) and following reverse dependency relations down to the leaf nodes (e.g. *letter* (4) modifies *reads* (2) as either SUBJ or DOBJ). The network of figure 2 comprises eight different syntactic readings, since both the subject-object distinction and the attachment of the determiner are still ambiguous and multiple modifications by the same label have not yet been ruled out.

A set of constraints over subordination relations (the grammar) describes admissible combinations of values. 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 rela-

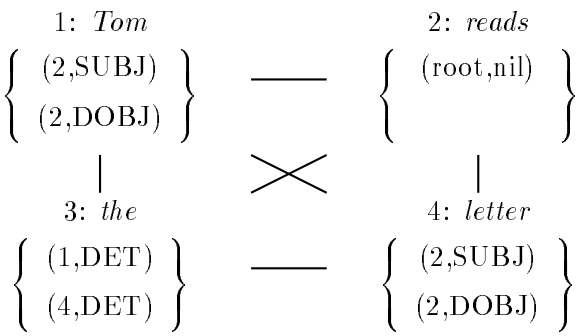

Figure 2: A constraint network for the sample input of Figure 1.

tionship. 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 a path condition. It requires all word forms of a description to belong to a single spanning path in the lattice (Harper and Helzerman, 1995).

Since parsing of natural language utterances almost always starts with considerable initial ambiguity of lexical entries, in principle we can understand lattice parsing as a parsing problem with extremely increased initial ambiguity.

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. However, a more elaborate termination criterion is required, in order to extend the treatment to graph-structured representations. This kind of situation is quite common for semantic relations where e.g. referential expressions have to be considered.

The eliminative nature of Constraint Satisfaction produces several interesting properties which constructive procedures definitely lack. First, Constraint Satisfaction introduces a rudimentary notion of robustness 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. Then,

1. any input sentence — including the clearly ungrammatical ones — will receive a complete, but possibly counterintuitive or inconsistent, 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 is possible even if not all of the constraints have yet been considered. 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 depends highly on the accidental sequence of constraint applications. However, it becomes

possible to introduce robustness in a more systematic way if we include a mechanism which is able to weight contradictory constraints against each other. Then, 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 us 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 which amounts to two for the nodes number 1, 3 and 4 of figure 2. 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 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 system's internal model provides for a certain degree of redundancy. For parsing, this redundancy can be introduced by a strict modularization into two almost independent description levels, namely a syntactic and a semantic one (Menzel, 1995). Additional representational levels, such as a referential one which might include discourse information, can be added in a similar way. Likewise, the approach can be extended easily to include additional sources of evidence (e.g. prosodic cues) in a modular fashion.

The representational 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 mutual compensation for information deficits or 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 form which a semantic interpretation can be derived.

In order to achieve robust behaviour Shimazu et al. (1994) devise a similar, fully symmetrical architecture. It uses different autonomous components which cooperate to build a structural description for an utterance. Starting from a usual codescriptive unification grammar they derive three distinct constraint sets for syntactic, semantic and combined regularities which operate independently from each other, but provide partial interpretations which possibly can be combined. Since the approach has no mechanism to arbitrate between conflicting evidence, ill-formed input can only be accepted if one component completely fails to obtain an analysis.

Figure 3 shows a sample grammar consisting of syntactic, semantic, and mapping constraints. Variables (upper case `X` and `Y`) refer to subordination relations, `role` indicates the description level, `lab` is the attached label, and $\uparrow$ 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

1. syntactic constraints

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

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

   sy2: X.role=syn $\wedge$ X.lab=SUBJ $\rightarrow$ X$\downarrow$num = X$\uparrow$num

   A subject agrees with its modifiee with respect to number.

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

   The subject precedes the finite verb.

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

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

2. semantic constraints

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

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

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

   Animals do eat.

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

   Plants are to be eaten.

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

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

3. mapping constraints

   ss1: X.role=syn $\wedge$ Y.role=sem $\wedge$ X$\uparrow$pos=Y$\uparrow$pos

   $\rightarrow$ (X.lab=SUBJ $\leftrightarrow$ Y.lab=AG)

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

   ss2: X.role=syn $\wedge$ Y.role=sem $\wedge$ X$\uparrow$pos=Y$\uparrow$pos

   $\rightarrow$ (X.lab=OBJ $\leftrightarrow$ 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

information is defeasible. Mapping relations mediating between syntactic and semantic descriptions are modelled as preferential constraints as well.

Assigning penalty factors to constraints introduces the preferential nature of grammatical knowledge. 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 rather *confidence scores* ranging from zero (for impossible combinations) to one (for combinations that do not violate 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 reflects the asymmetric nature of implications and helps to avoid some normalization problems immediately arising: Most modifying relations (or combinations of them) will pass a constraint simply because it is irrelevant for that particular configuration. An increase of goodness estimates for these cases would yield a highly undesirable, unjustified reinforcement. On the other hand, it has to be observed carefully that under this definitions high confidence scores by no means can be interpreted

as providing strong support for a partial solution, but only as indicating the absence of contradicting evidence.

If desired, penalty factors may be computed dynamically. Such an extension can be used to model 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, we can assign a distribution of penalty factors according to Table 1. This assignment was based on human judgements, although it can be derived from tree-annotated corpus data also. Note that the introduction of evidential reasoning allows the specification of overconstrained satisfaction problems. From the viewpoint of grammar modelling, this certainly is an advantage and not a fault, since it opens up the possibility to model and treat contradictory evidence in a systematic way. Therefore, it provides the prerequisites not only for the incorporation of semantic expectations (e.g. selectional constraints) but additionally, it facilitates the handling of unexpected input, which by definition introduces a contradiction to some model information.

In order to establish a suitable mechanism of arbitration between different sources of evidence, Constraint Propagation has to be extended from the classical case of strictly

syntax	semantics	mapping
pf(sy1)=0.0	pf(se1)=0.0	pf(ss1)=0.2
pf(sy2)=0.1	pf(se2)=0.7	pf(ss2)=0.3
pf(sy3)=0.3	pf(se3)=0.1	
pf(sy4)=0.0	pf(se4)=0.5	

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

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. The consequences of such a local pruning decision are then propagated over the network by a standard filtering algorithm. In order to avoid pruning promising hypotheses, the assessment procedure should only take into account modification relations characterized by the following three criteria:

1. being close to the global minimum for all modification relations, combined with

2. a maximal contrast with respect to alternative relations and

$v \in V$	a variable of the CS problem
$D(v)$	the domain of v
$x, y \in D(v)$	subordination relations
$c \in C$	a graded constraint
$p(c)$	the penalty factor of c
$f(\,), g(\,), h(\,)$	scores

FORALL pairs of subordination relations (x, y) DO

%%% constraint application, scoring of pairs of subordination relations

FORALL $c \in C$ DO

$$f(x, y, c) = \begin{cases} 1 & \text{if } c \text{ evaluates to TRUE for } (x, y) \\ p(c) & \text{otherwise} \end{cases}$$

ENDDO

$$g(x, y) := \prod_{c \in C} f(x, y, c)$$

ENDDO

DO UNTIL $\forall\ v$: $|D(v)| = 1$

%%% scoring of subordination relations

FORALL pairs of variables (v_1, v_2) DO

FORALL values $x \in D(v_1)$ DO

$$h(x, v_1, v_2) := \frac{\sum_{y \in D(v_2)} g(x,y)^2}{|D(v_2)|}$$

ENDDO

ENDDO

%%% selection and pruning of dispreferred subordination relations

$$x_i := \arg \min_{v_1 \neq v_2, x \in D(v_1)} h(x, v_1, v_2)$$

FORALL variables v DO

REMOVE x_i FROM $D(v)$

IF $\exists v \in V : |D(v)| = 0$ THEN EXIT: no consistent solution

ENDDO

ENDDO

Figure 4: The Constraint Satisfaction algorithm

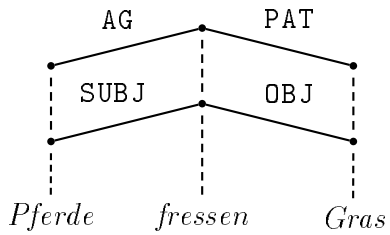

Figure 5: A two level description for a simple German sentence.

3. a minimal contrast between all the confidence scores supporting the relation in question.

For experimental purposes, we use a selection procedure based on the sum of quadratic errors for setting scores to zero. It satisfies criterion (1) and to a certain degree criterion (3) as well, but is missing the second one completely. Nevertheless, it guarantees that structural interpretations violating a high number of rather strong constraints are pruned first. Figure 4 shows a non-optimized version of the Constraint Satisfaction algorithm. In this simplified version Waltz-Filtering is subsumed by the general pruning procedure as the special case of removing values with zero scoring $h(x, v_1, v_2) = 0$.

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

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

both layers uniformly support the single interpretation shown in Figure 5.

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 support from semantic constraints in the grammar 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 headline-style utterance, syntactic evidence will gain the upper hand against the violation of two selectional constraints. However, this interpretation 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

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

Table 2: Parsing results for gradually deviating input.

result according to Figure 5 has been obtained. The distribution of positive results clearly demonstrates the compensatory effects between syntactic and semantic contributions. The procedure can easily be biased towards a particular semantic expectation by introducing specifically tailored constraints into the disambiguation process. Such expectations may be derived from contextual considerations or task specific constraints (Jacobs, 1992). It is important to note that even such a dynamic addition of constraints is completely modular and does not require any modification of the existing kernel grammar.

Syntactic or semantic deviations may be unveiled by monitoring the filtering activity in order to register serious conflicts between local pruning decisions and their consequences elsewhere in the network. Note that compared to simply ignoring an error, this monitoring activity requires additional efforts not directly contributing to the understanding process. They will be saved if time is short and enough semantic or contextual evidence available. This eventually leads to a mechanism resembling the human model's process characteristics rather closely.

5 Adaptive Parsing

In order to react sensibly to external time pressure, a system needs the ability to compromise between the desired quality of its output results and the resources allocated to carry out the computation (Boddy and Dean, 1994). 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 is then 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 riskier 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 obtain an output quality as high as possible given the time yet available. Three different strategies for achieving this kind of behaviour come to mind, two which essentially 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) makes use of dynamic penalty factors as described above. It maintains a threshold for penalty factors $\texttt{pf}_{\texttt{min}}$, 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. Again, increasingly risky decisions are taken.

Following this strategy, overconstrained problems are easily transformed into inconsistent ones. This happens systematically if unexpected input is combined with temporal pressure. In such a case, the procedure will fail to provide any consistent output at all and language understanding breaks down completely.

Despite the comparatively simple nature of the approach, preliminary experiments have shown its potential to supply the desired time-adaptive behaviour. Figure 6 shows the effects of increased temporal pressure on the parsing procedure and its results. The syntactic and semantic dimensions correspond exactly to those in Table 2. Temporal pressure (stress) is measured by the value of the threshold $\mathtt{pf_{min}}$. The height of the columns indicate the degree of pressure up to which a consistent solution has been obtained. Finally, the gray scale denotes the number of pruning cycles required to achieve a unique interpretation, where in each cycle the *globally* most disfavoured dependency relation is removed. Note that in a parallel implementation this relates to a point of process synchronization and therefore gives a first, albeit coarse, measure of temporal effort.

Two observations can be made instantly.

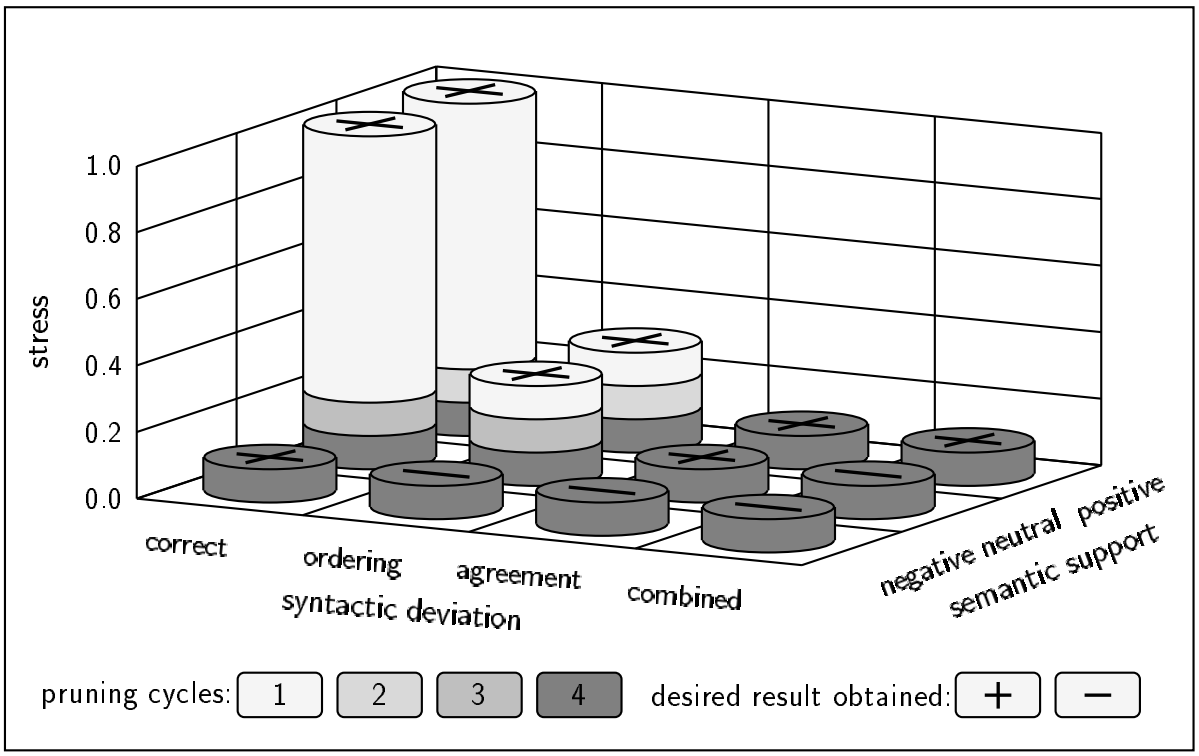

Figure 6: Parsing results under temporal pressure.

1. An increase in stress results in a reduction of temporal resources. For the four input variants with minor deviations, the number of required pruning cycles went down from four to a single one.

2. The ability to analyse deviant input breaks down under temporal pressure. Only the same four cases resisted a relatively weak pressure and delivered a consistent interpretation for $\mathtt{pf}_{\mathtt{min}} > 0.1$

These observations indicate at least a simple form of anytime-behaviour where time is exchanged for quality (namely robustness against deviant input), thereby linking two different notions of robustness in a very close manner. Robustness against temporal pressure

and the ability to tolerate syntactic and/or semantic distortions represent two orthogonal dimensions of system behaviour which have to be properly balanced in order to obtain an acceptable trade-off.

Unfortunately, this property is given only as long as the necessary degree of temporal pressure can be estimated prior to constraint application itself. Under more realistic conditions, scheduling will have to become a dynamic process which must 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 memorization mechanism for constraint applications and therefore gives rise to a substantial overhead in the organization of the propagation procedure.

Furthermore, the approach described so far shows the desired temporal adaptivity only during its second stage, namely the pruning procedure. To introduce a similar behaviour into the constraint application phase as well, the idea of explicit scheduling has to be pursued a little bit further. Based on a reliability estimation for constraints, the procedure can be extended to handle constraint sets with different degrees of reliability which might range from very general regularities to brute-force criteria. While the first group is usually used during early stages of the disambiguation procedure, the latter is invoked only under extreme pressure (c.f. Menzel (1994)). Note that it is possible to maintain such a classification of constraints 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 and determining the time point at which a particular group of constraints has to be considered.

6 Further Requirements: Temporal Incrementality and Parallel Propagation

Incremental processing along the time axis is a basic requirement for Spoken Language Systems and it is quite difficult to obtain. On the other hand there is a close interdependence between incrementality and anytime-behaviour for natural language understanding tasks: In general, only incremental processing environments will allow to estimate temporal pressure in a satisfactory manner by constantly comparing the progress of disambiguation against the amount of incoming recognition results.

Incremental parsing requires word-form hypotheses to be integrated immediately into the already existing structure and the constraint satisfaction problem will no longer remain a static one. The network is constantly being expanded by new variables, 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 makes them totally unsuitable for the task of structural disambiguation. Another difficulty results from the path condi-

tion in lattice parsing. Harper et al. (1995) present an algorithm for constraint satisfaction over lattice structures which, again, is non-incremental.

A similar observation can be made for concurrent approaches to the parsing problem. Constraint satisfaction techniques bear considerable potential for parallel execution on multiprocessor arrangements. This potential 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 produce a serious bottleneck. Helzerman 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 if any conditions these partially 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 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 and

4. incremental Constraint Satisfaction.

A coherent solution to these requirements will contribute significantly to the robustness of spoken language systems against a variety of adverse factors including ungrammatical input, recognition uncertainty and external temporal constraints. The unified treatment of such a diversity of phenomena by a single processing mechanism constitutes a novel level of achievement for Natural Language Processing systems. Even in its currently embryonic stage the system exhibits interesting robustness features which traditional constructive approaches definitely lack. Moreover, the desired behaviour turns out to be a direct consequence of the computational model. By avoiding special purpose add-on mechanisms it much closer resembles the cognitive reality of the human model than previous attempts.

None of the above-mentioned decision strategies has been evaluated to a sufficient degree yet. Therefore, an environment for extensive experimentation is still 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

Boddy, Mark and Thomas L. Dean. 1994. Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, 67(2):245–285.

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 1995. Extensions to constraint dependency parsing for spoken language processing. *Computer Speech and Language*, 9(3):187–234.

Harper, Mary P., Randall A. Helzerman, Carla B. Zoltowski, Boon-Lock Yeo, Yin Chan, Todd Steward and Bryan L. Pellom 1995. Implementation issues in the development of the PARSEC parser. *Software – Practice and Experience*, 25(8):831–862.

Helzerman, 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.

Jacobs, Paul S. 1992 Parsing Run Amok: Relation-Driven Control for Text Analysis. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 315–321, Seattle.

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.

Shimazu, Akira, Kiyoshi Kogure and Mikio Nakano. 1994 Cooperative Distributed Processing for Understanding Dialogue Utterances. In *Proceedings of the 1994 International Conference on Spoken Language Processing*, pages 99–102, Yokohama.

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