Parsing Generalized ID/LP Grammars

Michael W. Daniels
Department of Linguistics
Indiana University
402 Memorial Hall
1021 E Third Street
Bloomington, IN 47405
daniels@ling.osu.edu

1 Introduction

The Generalized ID/LP (GIDLP) grammar formalism (Daniels and Meurers 2004a,b; Daniels 2005) was developed to serve as a processing backbone for linearization-HPSG grammars, separating the declaration of the recursive constituent structure from the declaration of word order domains. This paper shows that the key aspects of this formalism – the ability for grammar writers to explicitly declare word order domains and to arrange the right-hand side of each grammar rule to minimize the parser's search space – lead directly to improvements in parsing efficiency.

2 Defining GIDLP Grammars

A brief overview of GIDLP syntax is given in 1, and an example GIDLP grammar is given in 2 that recognizes a very small fragment of German, focusing on the free word order of arguments and adjuncts in the *Mittelfeld*.[1] The basic idea of this grammar is that no word order constraints apply below the level of the clause. This allows the verb's arguments and adjuncts to freely intermingle, before being compacted at the clause level, at which point the constraints on the location of the finite verb apply. It is important to note that this grammar cannot be straightforwardly expressed in the ID/LP formalism, where LP constraints only apply within local trees.

Terminal: t
Non-terminal: C
Lexical entry: $C \rightarrow t$
Grammar rule: $C \rightarrow C^{+}; LP^{*}; DD^{*}$
Start declaration: $start(S) : LP^{*}$
LP [Constraint]: $C_1\{<, \ll\}C_2$
D[omain] D[eclaration]: $\langle\{C^{+}\}, C, LP^{*}\rangle$

Figure 1: GIDLP syntax

a) start(s): []
b) s → s(cmp)
c) s → s(que)
d) s(cmp) → cmp, clause;
⟨{[0]}, s(cmp), cmp < _, _ < v(_)⟩
e) s(que) → clause;⟨{[0]}, s(que), v(_) < _⟩
f) clause → np(n), vp
g) vp → v(ditr), np(a), np(d)
h) vp → adv, vp
i) vp → v(cmp), s(cmp)
j) [np(Case)] → det(Case), n(Case);1 ≪ 2

Figure 2: Example GIDLP Grammar

3 The GIDLP Parsing Algorithm

The GIDLP parser Daniels and Meurers (2004a); Daniels (2005) is based on Earley's algorithm for context-free parsing, suitably modified to handle discontinuous constituents.

A central insight of the GIDLP parsing algorithm is that the same data structure used to describe the coverage of an edge can also encode restrictions on the parser's search space. This is done by adding two bitvectors to each edge: a *negative mask* (n-mask), which marks positions that must not be part of the edge, and a *positive mask* (p-mask), which marks positions that must be part of the edge. These masks are generated during the prediction phase and then tested during the completion phase using efficient bitvector operations. Compiling LP constraints into

[1] For compactness, categories are described in this example with prolog-style terms; the actual GIDLP syntax assumes feature structure categories.

Proceedings of the Ninth International Workshop on Parsing Technologies (IWPT), pages 192–193,
Vancouver, October 2005.

bitmasks in this way allows the LP constraints to be integrated directly into the parser at a fundamental level. Instead of weeding out inappropriate parses in a cleanup phase, LP constraints in this parser can immediately block an edge from being added to the chart.

4 Evaluation

To evaluate the effectiveness of the GIDLP formalism, a moderate-scale grammar of German was obtained from Professor Martin Volk (Stockholm University). This grammar combines ID/LP rules with PS rules, as argued for in (Volk 1996), and uses a flat structure to encode the flexibility of the German *Mittelfeld*. As an example, the rule for ditransitive verbs is given in (1).

(1) `S -> N2 V N2 N2 ADV* (ERG) (PRAEF)`

This grammar can be mechanically translated into the GIDLP format, as each of Volk's PS rules corresponds to a GIDLP rule. This translation establishes an 'initial' GIDLP grammar.[2] The grammar was then optimized in two successive steps to take advantage of the GIDLP formalism. First, a 'medial' grammar was created in which word order domains were introduced only when necessary. (In the ID/LP formalism, every local tree is an order domain.) Second, a 'final' grammar was created by reordering the RHS order of each rule so as to put the most discriminatory RHS element first – generally the finite verb.

To compare these three grammars, a testsuite of 150 sentences was constructed. The sentences were generally chosen to equally cover the sentence types recognized by the grammar. The results from parsing this testsuite with each grammar are summarized in Figure 3, which shows the average number of chart insertion attempts at each sentence length. (Chart insertion attempts have traditionally been used as an overall metric for parsing efficiency, as parse time tends to be dominated by the time taken searching the chart for blocking edges.) Overall, the final grammar shows a clear improvement over the medial and initial grammars.

[2]As Volk's parser is not available, the relative performance of the GIDLP parser on the initial grammar and of Volk's parser on his grammar cannot be determined. Thus Volk's grammar is only used as a basis for the three GIDLP grammars described here.

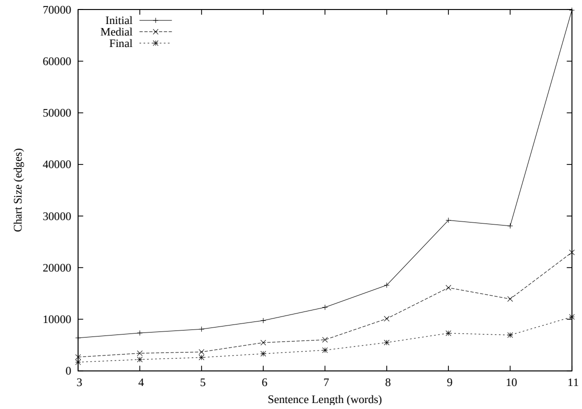

Figure 3: Average Chart Size per Sentence Length

Averaging over all 150 sentences, the final grammar sees a decrease of 69.2% in the number of chart insertion attempts compared to the initial grammar. Thus the expressive capabilities provided by the GIDLP formalism lead directly to improvements in parsing efficiency.

5 Summary

This paper has shown that two key aspects of the GIDLP grammar formalism – the ability for grammar writers to explicitly declare word order domains in the spirit of the linearization-HPSG tradition and the ability to completely order the RHS of a grammar rule to minimize the parser's overall search space – lead directly to improvements in parse efficiency.

References

Daniels, Michael W. 2005. *Generalized ID/LP Grammar: A Formalism for Parsing Linearization-Based HPSG Grammars*. Ph. D.thesis, The Ohio State University.

Daniels, Michael W. and Meurers, W. Detmar. 2004a. A Grammar Formalism and Parser for Linearization-based HPSG. In *Proceedings of the Twentieth International Conference on Computational Linguistics*, pages 169–175.

Daniels, Mike and Meurers, Detmar. 2004b. GIDLP: A Grammar Format for Linearization-Based HPSG. In Stefan Müller (ed.), *Proceedings of the Eleventh International Conference on Head-Driven Phrase Structure Grammar*, pages 93–111, Stanford: CSLI Publications.

Volk, Martin. 1996. Parsing with ID/LP and PS rules. In *Natural Language Processing and Speech Technology. Results of the 3rd KONVENS Conference (Bielefeld)*, pages 342–353, Berlin: Mouton de Gruyter.