GTU - eine Grammatik Testumgebung mit Testsatzarchiv

MARTIN VOLK, HANNO RIDDER
Universität Koblenz-Landau Institut
für Computerlinguistik Rheinau 3-4
5400 Koblenz 0261-9119-469 E-
Mail volk@brian.uni-koblenz.de 14.
Januar 1992

Abstract: GTU ist eine Entwicklungs- und Testumgebung für Phrasenstrukturgrammatiken und ID /LP (Immediate Dominance/Linear Precedence) Grammatiken im Rahmen des Unifikationsparadigmas. GTU enthält eine morphologische Komponente, Lexikon und Testsatzarchiv für ausgewählte Probleme des Deutschen, sowie einen speziellen Editor zur Grammatikentwicklung und eine Ausgabe-Routine. In diesem Aufsatz diskutieren wir in erster Linie die Probleme beim Aufbau des Testsatzarchivs sowie die Stellung von GTU als linguistischer Lehrsoftware.

1 Einleitung

Für Linguisten ist der Computer zum unersetzlichen Werkzeug für Experimente mit den verschiedensten Theorien geworden. Besonders im Bereich der Syntaxanalyse natürlicher Sprache ist das Experimentieren mit Grammatikimplementationen in verschiedenen Formalismen zum integralen Bestandteil der Ausbildung geworden. Diese Experimente erfolgten traditionell mithilfe der Programmiersprachen LISP und Prolog. Dabei nimmt das Zusammenstellen des Lexikons und die Programmierung von übersichtlichen Ausgabestrukturen viel Zeit in Anspruch, die für die eigentliche Aufgabe - dem Erstellen einer Grammatik - nebenseitig sind. Darüberhinaus ist es oft mühsam, Testsätze zusammenzustellen, die ein Phänomen in der jeweiligen Sprache ausreichend beschreiben. GTU ist mit dem Ziel entstanden, diese Probleme zu lindern. Lexikon und Testsätze sind vorgegeben. Das Lexikon enthält z. Zt. rund 250 Stammformen, denen ein Vielfaches an Wortformen entspricht. Die Bestände des Lexikons sind auf die Testsätze abgestimmt. Der Benutzer kann also sofort mit der Formulierung der Syntaxregeln beginnen und muß diese dann lediglich mit Interface-Regeln an das Lexikon anbinden. Wie dies geschieht, wird im Folgenden erläutert.

2 Das Erstellen einer Grammatik mit GTU

Mit GTU kann man Grammatiken (Phrasenstrukturregeln im DCG Formalismus oder Regeln im ID /LP Formalismus) entwerfen und mit einfachen Interface-Regeln an das Lexikon anbinden. Zum Editieren der Grammatik steht ein Editor zur Verfügung, der spezielle Hilfefunktionen für die Grammatikentwicklung bereitstellt. Die Grammatikregeln können mit Merkmalstrukturen versehen werden. Die Syntax für die jeweiligen Formalismen ist flexibel anpaßbar an linguistische Konventionen.

Beispiel 1: Eine Grammatikregel für Präpositionalphrasen
PP[typ=P] -+ Prep[typ=P, kas=K]
NP[kas=K].

PP, Prep und NP sind dabei Kategoriennamen, die Ausdrücke in den eckigen Klammern sind Merkmalstrukturen (Mengen von Merkmal-Wert-Paaren). Merkmalnamen und Bezeichnungen für Merkmalwerte können frei gewählt werden. Die Anzahl und Reihenfolge der Merkmale ist beliebig. Die Regeln können optionale Kategorien

und Terminalsymbole enthalten. Mit den Lexikon-Interface-Regeln legt der Benutzer fest, welche Information zu der jeweiligen Wortart aus dem Lexikon entnommen werden soll.

Beispiel 2: Eine Lexikon-Interface Regel für Präpositionen

```
If_in_lex (wortart=prep) then_in_gram
    Prep[ typ=#lemma, kas=#kasus].
```

Die Regel in Beispiel 2 besagt, daß für Präpositionen das Merkmal typ als Wert das jeweilige Lemma der morphologischen Analyse erhält, und daß das Merkmal kas den von der Präposition geforderten Kasus als Wert erhält. Die lexikalischen Regeln für die auftretenden Präpositionen werden dann während der morphologischen Analyse automatisch generiert.

3 Das Testsatzarchiv

Ist die Grammatik fertig editiert und geladen, muß überprüft werden, ob sie die erwarteten Strukturen erzeugt. Die Wichtigkeit der systematischen Überprüfung und Bewertung einer Grammatik kann leicht durch ein Zitat von Friedman (1989) motiviert werden: "Grammar writing is much more difficult than rule writing. The intricate interrelations of the individual rules of a grammar make grammar writing a complex and error-prone process, much like computer programming."

Zu diesem Zweck kann der Benutzer von GTU Testsätze manuell eingeben oder aus dem mitgelieferten Testsatzarchiv auswählen. Das Testsatzarchiv der GTU enthält rund 350 Testsätze in 15 Klassen. Der Aufbau des Testsatzarchivs geschah nach folgenden Kriterien:

I> Der Wortschatz der Testsätze kann beschränkt sein, da Methoden der Grammatikentwicklung eingeübt werden sollen und die Lexikonkomponente nur eine Servicefunktion hat.

I> Es gilt der Grundsatz *Vom Einfachen zum Komplexen,* da GTU als Lehr- und Lernsoftware konzipiert ist. Die Testsätze sollen aufeinander aufbauen, um eine inkrementelle Grammatikentwicklung zu fördern.

Die Testsätze der GTU wurden nach folgenden linguistischen Kriterien zusammengestellt: Zunächst gehen wir von einfachen Aussagesätzen aus, Nebensätze und andere Satztypen (Fragesätze) behandeln wir in fortgeschrittenen Lektionen. Innerhalb der Aussagesätze werden zunächst Kongruenzen (Subjekt-Prädikat, Artikel-Nomen) und Verbsubkategorisierungsrahmen variiert. Bei der Subkategorisierung sind anfangs nur obligatorische, später auch fakultative Mitspieler und freie Ergänzungen zu berücksichtigen. Im nächsten Schritt werden Verbformen und Nominalphrasen komplexer. Bei den Verbformen beginnen wir mit einfachen Vollverben, gehen dann zu zusammengesetzten Verbformen in Perfekt und Futur sowie Modalverb- und Kopulakonstruktionen. Nominalphrasen werden durch Adjektiv- und Genitivattribute erweitert. Danach kommen Besonderheiten des Deutschen wie abtrennbare Verbpräfixe, die verschiedenen Funktionen von *es* sowie Wortstellungsphänomene. Dann werden die bisher erarbeiteten Konstituenten durch Konjunktionen verbunden und zu komplexeren Konstruktionen zusammengefügt. Schließlich enthält das Archiv Satztypen, die andere Wortstellungen aufweisen als der Aussagesatz. Wir haben beispielhaft Fragesätze, daß-Sätze und Relativsätze ausgewählt.

Die Bereitstellung von vorbereiteten Testsätzen bietet eine Reihe von Vorteilen: 0 Die Testsätze können auf die Bestände des Lexikons abgestimmt werden.

0 Syntaktische Phänomene können systematisch aufgelistet werden.

0 Die Anzahl der Testläufe wird im Vergleich zu unspezifischen Eingabetexten minimiert.

0 Standardisierte Testsätze ermöglichen vergleichende Untersuchungen zwischen verschiedenen Formalismen bezüglich Effizienz und Kürze.

Bei der Einteilung der Testsätze in Problemklassen treten verschiedene Komplikationen auf: Die beschriebenen Phänomene sind mit zunehmender Komplexität schwieriger zu isolieren. So ist die Wortstellung ein Phänomen, das sich durch alle Testsätze zieht, aber von uns erst da aufgegriffen wird, wo es um variable Wortstellung geht (z.B. Mittelfeldanordnung der Nominalphrasen im Deutschen). Andere Phänomene, die wir ganz zu Anfang behandeln, treten im fortgeschrittenen Lektionen in anderer Form wieder auf (z.B. Kongruenz zwischen Artikel und Adjektiv in einer NP bzgl. der Adjektivdeklination). Man kann ein Phänomen nicht erschöpfend abdecken, ohne anderen Phänomenen vorwegzugreifen. Dennoch muß eine Einteilung erfolgen, damit ein schrittweises Vorgehen möglich ist.

Außer grammatisch korrekten Sätzen müssen auch ungrammatische Ketten aufgenommen werden, um eine Übergenerierung der Grammatik zu erkennen. Die Konstruktion der ungrammatischen Ketten kann spezifischer für das aktuelle Phänomen erfolgen als die Wahl eines grammatisch korrekten Satzes. Man wählt die Kette derart, daß ihre Ungrammatikalität lediglich von dem gewünschten Phänomen abhängt.

Beispiel 3: Eine ungrammatische Kette bezüglich Artikel-zu-Adjektiv Kongruenz

```
*Peter sieht das schönes Haus.
```

Darüber hinaus ist die Wahl der ungrammatischen Ketten abhängig von dem zu bearbeitenden Grammatikformalismus. Ist beispielsweise eine Grammatik im Rahmen des **ID** /LP Formalismus zu erstellen, so sind ungrammatische Ketten, die Wortstellungsregularitäten überprüfen, wichtiger als beim DCG Formalismus. Das ist darin begründet, daß beim ID/LP Formalismus die LP-Regeln explizit die Wortstellung fixieren, während die Wortstellung bei DCG implizit in den Regeln festgelegt ist.

Damit das Testsatzarchiv auch für große Sammlungen operational bleibt, müssen folgende Kriterien bedacht werden:

- Das Testsatzarchiv muß modular aufgebaut und leicht erweiterbar sein.
- Die Gründe für die Aufnahme eines Testsatzes in das Archiv müssen dem Benutzer transparent gemacht werden.
- Der Benutzer muß verschiedene Sichtweisen zur Verfügung haben, um sich auf unterschiedlichen Ebenen einen Überblick über das Archiv verschaffen zu können.
- Der Benutzer muß nach verschiedenen Gesichtspunkten Testsätze (auch über die gewählte Klasseneinteilung hinaus) auswählen und testen können.
- Protokollierung und Auswertung der Testergebnisse müssen dem Benutzer übersichtlich dargeboten werden. Das umfaßt die Präsentation der jeweils erfolgreich und fehlerhaft akzeptierten sowie abgelehnten Testsätze.

GTU erfüllt die meisten dieser Bedingungen. GTU unterstützt die Erweiterbarkeit des Testsatzarchivs um weitere Testsätze und Testsatzklassen. Zu diesem Zweck steht eine Musterdatei zur Verfügung, die das Eingabeformat für neue Testsätze erklärt. Die Datei wird automatisch geladen, wenn eine neue Testsatzklasse angelegt wird. Um zu dokumentieren, warum ein Testsatz aufgenommen wurde, wird jedem Testsatz eine Liste von Keywords mitgegeben, die auf ausführliche Begründungen verweisen, die an anderer Stelle abgelegt sind. Die Sammlung der Begründungen gibt einen Überblick über die behandelten Phänomene. Der Benutzer kann das GTU- Testsatzarchiv auf verschiedene Art einsehen. So kann er ein Verzeichnis aller Testsatzklassen, einzelne Testsatzklassen oder einzelne Testsätze mit ihrer jeweiligen Begründung einsehen. Übersichtliche Menüs erleichtern dabei die Auswahl. Mit diesen Optionen wählt der Benutzer auch Testsätze aus und übergibt sie dem System zur weiteren Verarbeitung. Protokollierung und Auswertung der Testergebnisse sind derzeit noch nicht implementiert.

4 Verarbeitung der Testsätze

Nach der Auswahl der Testsätze werden diese nacheinander abgearbeitet. Dazu wird zunächst eine morphologische Analyse sämtlicher Worte des Eingabesatzes durchgeführt und die entsprechenden lexikalischen Regeln werden erzeugt. Dann kann die eigentliche Syntaxanalyse beginnen. Grammatiken im DCG Formalismus werden dem Prolog Top-Down Left-To-Right Parser übergeben. Grammatiken im **ID** /LP Formalismus werden durch einen Bottom-Up Left-To-Right Chart-Parser abgearbeitet.

Bei erfolgreicher Analyse wird automatisch eine übersichtliche Baumstruktur erzeugt, ohne daß der Benutzer spezielle Parameter in seine Grammatik einbauen muß. Diese Ausgabestruktur enthält die Kategorienamen und Terminalsymbole. Danach wird in eingerückter Form die gesamte Information jeder Konstituente ausgegeben. Das heißt, die komplette Merkmalstruktur für jede Konstituente wird dem Benutzer dargeboten, damit er kontrollieren kann, ob die gewünschte Struktur erzeugt wurde. Ist ein Satz syntaktisch mehrdeutig, werden alle möglichen Strukturen erzeugt. Wird die Eingabe von der Grammatik nicht akzeptiert, erscheint eine entsprechende Meldung. Der Benutzer kann die Analyse einsehen, indem er die morphologische Analyse oder auch den Parsingvorgang im Trace-Modus verfolgt.

5 Der didaktische Ansatz von GTU

In den letzten Jahren sind in der Syntaxtheorie zwei Trends zu beobachten. Zum einen wird die Unifikation über Merkmalstrukturen zur wichtigsten Operation innerhalb der Theorien. Zum anderen haben neuere Formalismen (wie GPSG und HPSG) zunehmend deklarativen Charakter. Diesen Trends versucht GTU Rechnung zu tragen. Zwar kann mit GTU z. Zt. noch keine vollständige Implementierung von GPSG oder HPSG durchgeführt werden, aber die Studierenden lernen einerseits den Umgang mit Merkmalstrukturen und den Auswirkungen der Unifikation, und andererseits das deklarative Vorgehen mit DCG- und ID /LP-Grammatiken. Letztere stellen eine interessante Alternative für Sprachen mit variabler Wortstellung dar und sind somit zur Analyse der deutschen Sprache besonders geeignet. GTU ist als Hilfswerkzeug für Lernende konzipiert. Es wendet sich an StudentInnen der (Computer-)Linguistik und verwandter Fächer, die mit der Arbeit am PC in den Grundzügen vertraut sind und schon grundlegende Programmiererfahrung haben. Durch die automatische Visualisierung von Satzstrukturen wird die Motivation zur Arbeit in der Syntaxana-

lyse gestärkt. GTU erlaubt ein inkrementelles Vorgehen bei der Grammatikentwicklung, sodaß Zwischenergebnisse leicht überprüft werden können.

6 Abgrenzung zu existierenden Programmen

GTU unterscheidet sich von anderen Entwicklungsumgebungen durch seine Konzeption als Hilfswerkzeug für Lernende. Vergleichbare Programme wie z.B. GIATN (Brockmann, Fuchs 1990) für ATNs oder TAGDevEnv (Schifferer 1988) für Tree Adjoining Grammars sind dagegen als Werkzeuge für den Linguisten gedacht, der komplexe linguistische Systeme entwickeln will, die u. U. lexikalische, grammatische, semantische und pragmatische Wissensrepräsentation erfordern. GTU ist beschränkt auf die Entwicklung von Syntaxregeln und eignet sich besonders für die schnelle Einarbeitung in einen kleinen Problembereich.

Ein neuerer Ansatz von Erbach (1991) beschreibt ein System, das zum Experimentieren mit unterschiedlichen Parsingstrategien dient. Dabei kann der Benutzer Prioritäten für verschiedene Parsingaufgaben spezifizieren und damit den Parsingprozeß inkrementell optimieren. Wir glauben jedoch, daß ohne standardisierte Testsätze kein wirklicher Vergleich auf breiter Ebene möglich ist und sehen unseren Ansatz als komplementär zu dem von Erbach vorgestellten.

Einige Problembereiche sind in GTU durch die Testsatzklassen des Testsatzarchivs vorgegeben, wobei wir gleichzeitig die Verwaltung weiterer Testsatzklassen ermöglichen. Das war bei bisherigen, uns bekannten Systemen nicht der Fall. Nerbonne (1991) et al. beschreiben eine Arbeit, die in dieselbe Richtung geht. Sie sind dabei, eine
große Satzsammlung für das Deutsche in einer Datenbank zu organisieren. Auch in diesem System werden die Sätze zunächst nur nach syntaktischen Gesichtspunkten klassifiziert.

Schließlich sehen wir einen großen Vorteil unseres Systems in der Anbindung unterschiedlicher Formalismen an ein einheitliches Lexikon. Andere Systeme sind entweder monoformalistisch oder erfordern unterschiedliche Lexika für unterschiedliche Formalismen. Unser Vorgehen bietet nicht nur Vorteile für die Grammatikentwicklung, sondern spiegelt auch unsere Überzeugung wider, daß die Entwicklung von natürlich-sprachlichen Systemen nur durch systematische Nutzung der Lexikonresourcen vorangetrieben werden kann.

7 Ausblick

GTU wurde mehrfach erfolgreich in der Übung zu *Methoden der Syntaxanalyse* an der Universität Koblenz eingesetzt. Nach Aussage der betroffenen Studierenden ist das Programm leicht zu erlernen und stellt gegenüber einer Grammatikentwicklung in Prolog eine spürbare Erleichterung dar. Letzteres wird begründet mit der Verwendung der Lexikonkomponente sowie des automatisch erzeugten Strukturbaums bei der Ausgabe. Eine genaue Beschreibung der Funktionalität liefern Volk, Ridder (1991). Der Anschluß von weiteren Grammatikformalismen (LFG, GPSG) .ist in der Planung.

Literatur

[1] Brockmann, S.; Fuchs, U.: Eine ATN-Werkbank als erste Ausbaustufe für eine Graphic Interactive ATN Workstation. Diplomarbeit. Koblenz: Universität Koblenz-Landau. April 1990.

[2] Erbach, G.: An Environment for Experimentation with Parsing Strategies. (IWBS Report 167) Stuttgart: Wissenschaftliches Zentrum der IBM Deutschland. April 1991.

[3] Friedman, J.: Computational Testing of Linguistic Models in Syntax and Semantics. In: Batori, I. et al. (Eds.): Computational Linguistics. An international handbook on Computer Oriented Language Research and Applications. Berlin: Walter de Gruyter, 1989.

[4] Nerbonne, J. et al.: A diagnostic tool for German syntax. (Research Report RR-91-18) Saarbrücken: DFKI. Juli 1991.

[5] Schifferer, K.: TAGDevEnv. Eine Werkbank für TAGs. In: Batori, I. et al. (Hgg.): Computerlinguistik und ihre theoretischen Grundlagen. Berlin: Springer Verlag, 1988.

[6] Volk, M.; Ridder, H.: GTU (Grammatik Test Umgebung) Benutzerhandbuch. (Manuskript) Institut für Computerlinguistik. Universität Koblenz-Landau. 1991.