

Linguistische  
Arbeiten

365

Herausgegeben von Hans Altmann, Peter Blumenthal, Herbert E. Brekle,  
Gerhard Helbig, Hans Jürgen Heringer, Heinz Vater und Richard Wiese



*Jean-Yves Lalande*

# Verbstellung im Deutschen und Französischen

Unter Anwendung eines CAD-basierten  
Expertensystems

Max Niemeyer Verlag  
Tübingen 1997



Die Deutsche Bibliothek – CIP-Einheitsaufnahme

**Lalande, Jean-Yves:** Verbstellung im Deutschen und Französischen : unter Anwendung eines CAD-basierten Expertensystems / Jean-Yves Lalande. – Tübingen : Niemeyer, 1997  
(Linguistische Arbeiten ; 365)

NE: GT

ISBN 3-484-30365-4      ISSN 0344-6727

© Max Niemeyer Verlag GmbH & Co. KG, Tübingen 1997

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen. Printed in Germany.

Gedruckt auf alterungsbeständigem Papier.

Druck: Weihert-Druck GmbH, Darmstadt

Buchbinder: Industriebuchbinderei Hugo Nädele, Nehren

# Inhaltsverzeichnis

Vorwort.....	IX
1. Einleitung.....	1
1.1 Zielsetzung.....	1
1.2 Gliederung der vorliegenden Arbeit.....	2
1.3 Linguistische Hypothesenbildung in GB.....	2
1.4 Modellierung der linguistischen Forschungsarbeit mit GBX als CAD-orientiertem Expertensystem.....	4
1.5 Abgrenzung von GBX zu Systemen ähnlicher Zielsetzung.....	6
1.5.1 Linguistische Tools.....	6
1.5.2 Stablers Implementation der GB-Theorie.....	7
1.5.3 Die Sprache CLAT.....	8
2. Das System GBX.....	9
2.1 GBX und das System LPS.....	9
2.2 Überblick über die graphische Oberfläche von GBX.....	10
2.3 Erzeugung und Repräsentation von Baumstrukturen mit GBX.....	12
2.3.1 Erstellen der Baumstrukturen.....	12
2.3.1.1 Attribute und Attributwerte.....	13
2.3.1.2 Beschreibung einer Grundstruktur.....	14
2.3.1.3 Angabe der Projektionsebene.....	15
2.3.1.4 Anreicherung der Regeln durch Merkmale.....	16
2.3.1.5 Lexikalische Angaben.....	17
2.3.1.5.1 'Leere' lexikalische Angaben.....	17
2.3.1.5.2 Interaktion mit Lexika.....	18
2.3.2 'Parsen' der Regeln.....	21
2.3.3 'Darstellen' der Baumstruktur.....	22
2.3.4 Konvertierung und teilautomatische Generierung von Strukturbäumen.....	24
2.3.4.1 Die 'Str- Dateien'.....	24
2.3.4.2 Automatische Konvertierung von Strukturbäumen.....	25
2.3.4.3 Teilautomatische Generierung von Strukturbäumen.....	26
2.4 Erzeugung linguistischen Wissens mit GBX.....	27
2.4.1 Von der linguistischen zur prädikatenlogischen Formulierung der GB-Prinzipien.....	27
2.4.2 Implementation der GB-Prinzipien in LPS-Prolog.....	30
2.4.2.1 Programmierung in LPS-Prolog.....	30

2.4.2.2	Die vordefinierten Prädikate von LPS-Prolog.....	34
2.4.2.2.1	Built-in Prädikate .....	34
2.4.2.2.2	Zugriffsprädikate .....	37
2.4.2.2.3	Ausgabepredikate.....	38
2.4.2.3	Ein Beispiel: 'traditionelle' Implementation des Prinzips der Rektion.....	39
2.4.3	Objektorientierung und GB(X) .....	41
2.4.3.1	Grundbegriffe des objektorientierten Ansatzes .....	42
2.4.3.2	Klassendefinition und statische Vererbung .....	44
2.4.3.3	Instantiierende Vererbung .....	47
2.4.3.4	Dynamische Vererbung .....	48
2.4.3.5	Ein einfaches Beispiel: Die Überprüfung des Thetakriteriums.....	49
2.4.3.6	Obligatorische vs. fakultative Methoden.....	54
2.4.3.7	Das Prinzip der Rektion im objektorientierten Ansatz .....	56
2.4.3.7.1	M-Kommando und dynamische Vererbung.....	58
2.4.3.7.2	Barrieren und Objektorientierung .....	58
2.4.3.7.2.1	Barrieren .....	59
2.4.3.7.2.2	Barriere und dynamische Vererbung .....	60
2.4.3.7.3	Objektorientierte Überprüfung von Rektionsverhältnissen .....	62
2.4.3.8	Späte Instantiierung .....	65
2.4.3.9	Fazit: GB und Objektorientierung .....	68
2.5	Verwaltung des linguistischen Wissens .....	69
2.5.1	Modularisierung des linguistischen Wissens .....	69
2.5.1.1	Modularisierung der linguistischen Prinzipien.....	69
2.5.1.2	Modularisierung der Klassendefinitionen.....	70
2.5.2	Verwaltung des modularisierten linguistischen Wissens.....	70
2.6	Interaktive Evaluierung von Strukturen anhand des Expertenwissens .....	72
2.6.1	Automatische Evaluierung von Strukturen.....	72
2.6.2	Benutzergesteuerte Evaluierung .....	73
2.6.2.1	Anklicken von Methoden im Baum.....	73
2.6.2.2	Die Menü-Option 'Aktivieren' .....	75
2.7	Die Funktionen des Systems .....	76
2.7.1	Die Expertenfunktion.....	76
2.7.2	Die Visualisierungsfunktion .....	77
2.7.3	Die Datenbankfunktion.....	78
2.8	Zusammenfassung: Linguistische Forschungsarbeit mit GBX.....	79

3. Kontrastive Analyse der Verbstellung im Deutschen und Französischen unter Anwendung von GBX.....	82
3.1 Verbstellung im Deutschen .....	82
3.1.1 Standardanalyse der Verbstellung im Deutschen .....	84
3.1.2 Das Reissche Dilemma .....	87
3.1.3 Strenge Verbzweitstellung.....	88
3.2 Verbstellung im Französischen .....	89
3.2.1 Verbzweitstellung im Altfranzösischen.....	90
3.2.2 Verlust der Verbzweitstellung im Französischen .....	93
3.2.3 Verbbewegung zu Infl im Französischen .....	94
3.2.4 Verbbewegung zu Comp im Französischen .....	97
3.3 Kontrastive Analyse mit GBX.....	98
3.3.1 Analyse der komplexen Inversion im Französischen .....	98
3.3.2 Die komplexe Inversion im Französischen als 'Wurzel'-Phänomen.....	101
3.3.3 Objektorientierte Implementation der aufgestellten Hypothesen .....	104
3.3.4 Evaluierung und Erweiterung der Hypothesen anhand des Deutschen.....	107
3.4 Zusammenfassung.....	109
4. IP-Struktur, objektorientierter Ansatz und Minimalist Program.....	110
4.1 Infl: Von einer 'flachen' Struktur zur 'Barriers'-Struktur des Satzes .....	110
4.2 Pollocks Split-Infl-Analyse .....	111
4.2.1 Pollocks Aufteilung von IP .....	112
4.2.2 Pollocks thetatheoretische Analyse.....	114
4.2.2.1 Opazität vs. Transparenz zur Thetarollenzuweisung.....	114
4.2.2.2 Objektorientierte Implementation.....	114
4.2.3 Pollocks quantifikationstheoretische Analyse .....	116
4.2.3.1 Verbbewegung und Quantifikationstheorie im Französischen .....	116
4.2.3.2 Objektorientierte Implementation.....	120
4.2.3.3 Reformulierung der Aussagen .....	122
4.2.4 Anwendung der implementierten Theorie auf englische Daten .....	122
4.2.5 TP über oder unter AgrP? .....	127
4.2.6 Einige Argumente gegen eine Aufsplitterung von IP .....	127
4.3 Chomskys Erweiterung von Pollock (1989).....	129
4.3.1 AgrSP und AgrOP .....	129
4.3.2 Objektorientierte Implementation.....	131
4.3.3 Subjekt- und Objekt-Verb-Kongruenz in französischen Partizipialen .....	134
4.4 Minimalist Program und Objektorientierung .....	137
4.4.1 Von Chomsky (1989) zum Minimalist Program.....	137
4.4.2 Checking-Theory .....	138

## VIII

4.4.3 Objektorientierte Implementation der Checking Theory.....	139
4.4.4 Parametrische Variationen.....	142
4.4.5 Objektorientierte Implementation.....	143
4.5 Zusammenfassung.....	145
5. Resümee und Ausblick.....	147
5.1 Zusammenfassung der Ergebnisse .....	147
5.2 Weitere Perspektiven.....	148
6. Literaturverzeichnis.....	151
7. Sachregister .....	161
8. Autorenregister.....	165

## Vorwort

Dieses Buch ist die aktualisierte Fassung meiner Dissertation, die im Wintersemester 1995/1996 von der Philosophischen Fakultät der Universität zu Köln angenommen wurde. Die Referenten waren Prof. Dr. Heinz Vater und Prof. Dr. Jürgen Rolshoven; das Rigorosum fand am 18. November 1995 statt.

Mein besonderer Dank gilt Herrn Prof. Dr. Heinz Vater, der die Entwicklung von VisualGBX und die Entstehung meiner Dissertation immer unterstützte und mit großem Interesse verfolgte und der als einer der Herausgeber die Aufnahme dieses Buches in die Reihe *Linguistische Arbeiten* ermöglichte.

Die Anregung, das System VisualGBX zu entwickeln und es auf das Deutsche und Französische anzuwenden, wurde von Herrn Prof. Dr. Jürgen Rolshoven gegeben. Hierfür sowie für seine stetige Unterstützung und den fruchtbaren Austausch während der letzten Jahre möchte ich ihm an dieser Stelle herzlich danken.

Dipl. Math. Michael Holst (Regionales Rechenzentrum an der Universität zu Köln) verdanke ich wertvolle Konzepte der Programmierung graphischer Benutzeroberflächen. Gleichfalls danken möchte ich Thomas Hessel, der mir wiederholt bei der Windows-Programmierung zur Seite stand.

Dr. Guido Mensching hat mir immer wieder durch Gespräche, Anregungen und Ratschläge geholfen. Hierfür danke ich ihm sehr, desgleichen Tobias Schoofs, dessen Ausdauer, Kritik, Witz und Begeisterung für VisualGBX mir eine große Hilfe waren.

Dr. Olaf Nüsser war mir bei der Anfertigung der Druckfassung sehr behilflich.

Besonderer Dank gebührt schließlich meinen Schwiegereltern, die immer für Jana, Jules und Mirka da waren, wenn es bei der Bewältigung des Alltags manchmal zu Engpässen kam.

Anne und den drei Letztgenannten widme ich dieses Buch.

Köln, im Sommer 1996

Jean-Yves Lalande



# 1. Einleitung

## 1.1 Zielsetzung

Innerhalb der generativen Grammatik zeichnet sich die linguistische Hypothesenbildung durch eine sehr hohe Produktivität aus, welche einerseits auf die Vielfalt der im Rahmen einer allgemeinen Sprachtheorie zu erklärenden Phänomene, andererseits aber auch auf einen Mangel an Vereinheitlichung innerhalb der Theorie zurückzuführen ist. Beispielhaft dafür sind die von Pollock (1989) eingeleitete Diskussion über die Struktur der funktionalen Kategorie IP, die in den letzten Jahren zu zahlreichen alternativen Analysen der Satzstruktur geführt hat (vgl. z.B. Pollock 1989, Belletti 1989, Chomsky 1989, Schmidt 1994), sowie die innerhalb der GB-Theorie konkurrierenden Varianten der X-bar-Theorie (s. z.B. Chomsky 1986b, Larson 1988, 1990, Hoekstra 1991, Kayne 1993). Vor dem Hintergrund der Zielsetzung der GB-Theorie, ein universelles Modell zur Erklärung der menschlichen Sprachkompetenz aufzustellen, sind eine Vereinheitlichung und gleichzeitig eine Vereinfachung der Theorie jedoch unbedingt geboten und nicht zuletzt im Hinblick auf die Übernahme der herausgearbeiteten Prinzipien in computerlinguistische Anwendungen sinnvoll. Mit diesem Ziel wurde das System VisualGBX, im folgenden kurz GBX genannt, entwickelt. GBX modelliert und unterstützt die Arbeit des GB-orientierten Sprachwissenschaftlers mit Hilfe von Techniken der Expertensystemtechnologie und des Computer Aided Design (CAD) und trägt dank der Systematisierung und einer weitgehenden Automatisierung grundlegender Tätigkeiten des Linguisten zur Weiterentwicklung und Vereinheitlichung der Theorie bei. Indem es eine neue, objektorientierte Sicht der Syntaxanalyse ermöglicht, eröffnet GBX darüber hinaus innovative Perspektiven bei der Untersuchung sprachlicher Phänomene, die in vieler Hinsicht mit den neuen Entwicklungen zu Minimalität und Lokalität im Rahmen des Minimalist Program Chomskys (Chomsky 1992) kongruieren. Die oft statische Sicht der GB-Theorie, in der von fertigen Strukturen ausgegangen wird, wird durch eine dynamische ersetzt, die dem mentalen Prozeß bei der Sprachanalyse und -synthese näherkommt und somit der Theorie eine neue kognitive Plausibilität verleiht.

Ziel dieser Arbeit ist es zu zeigen, wie der Einsatz von GBX als Werkzeug zur Theorieentwicklung und -evaluierung die Forschungsarbeit des Linguisten innerhalb der generativen Grammatik effizient unterstützen und zu neuen Lösungsansätzen bei der Untersuchung syntaktischer Phänomene führen kann.

## 1.2 Gliederung der vorliegenden Arbeit

Im folgenden wird, von der Beschreibung grundlegender Arbeitsmethoden der linguistischen Hypothesenbildung ausgehend (1.3), die Konzeption des Systems GBX, das die Arbeit des GB-orientierten Sprachwissenschaftlers modelliert und spiegelt, skizziert (1.4), bevor auf besondere Eigenschaften von GBX im Vergleich zu anderen Systemen ähnlicher Zielsetzung hingewiesen wird (1.5).

In Kapitel 2 wird die Funktionsweise des Systems GBX detailliert beschrieben. Dabei wird besonderer Wert auf die Übernahme des Konzepts der Objektorientierung in die (computer-)linguistische Analyse mit GBX gelegt.

Das Kapitel 3 widmet sich der kontrastiven Analyse der Verbstellung im Deutschen und Französischen unter Anwendung von GBX. An konkreten Beispielen wird gezeigt, wie das System zur Implementation, Evaluierung und Erzeugung linguistischer Hypothesen eingesetzt werden kann.

Das Kapitel 4 beschäftigt sich mit theoretischeren Überlegungen über die Struktur des Satzes, wobei mit Hilfe von GBX gezeigt wird, inwiefern die objektorientierte Erweiterung der Theorie parallel zu aktuellen Entwicklungen im Rahmen des Minimalist Program verläuft und mit ihnen kongruiert.

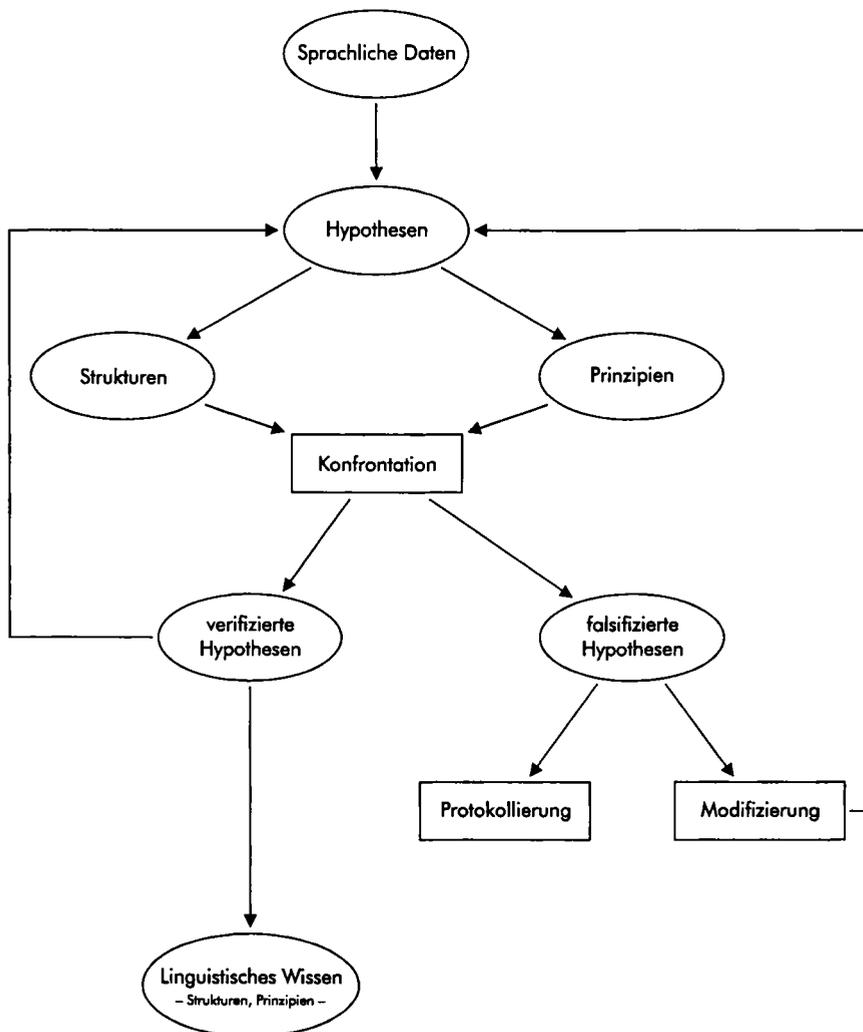
Die Ergebnisse der Arbeit werden schließlich in Kapitel 5 zusammengefaßt, in dem weiterhin einige Zukunftsperspektiven für die Weiterentwicklung und die weitere linguistische Forschungsarbeit mit GBX angeführt werden.

## 1.3 Linguistische Hypothesenbildung in GB

Die linguistische Hypothesenbildung innerhalb der GB-Theorie bedient sich syntaktischer Strukturbäume, um sprachliche Phänomene im Rahmen einer expliziten Theorie der menschlichen Sprachfähigkeit zu beschreiben und zu erklären. Diese Baumstrukturen unterliegen strengen Wohlgeformtheitsbedingungen, die universelle, z.T. aber sprachspezifisch parametrisierte abstrakte grammatische Prinzipien widerspiegeln und von der Theorie aufgedeckt und formalisiert werden. Da es um die Erforschung der universellen Eigenschaften von natürlichen Sprachen und der den Unterschieden zwischen diesen Sprachen entsprechenden Parametereinstellungen geht,<sup>1</sup> ist es dabei unumgänglich, jede von der Theorie aufgestellte Hypothese zwecks Evaluierung und ggfs. Falsifizierung mit der größtmöglichen Zahl sprachlicher Daten zu konfrontieren. Zur Formulierung und Überprüfung von Hypothesen werden daher in der linguistischen Praxis für zahlreiche Sätze aus verschiedenen Sprachen zahlreiche Baumstrukturen unter stetiger Berücksichtigung der von der Theorie herausgearbeiteten Prinzi-

<sup>1</sup> Die Begriffe 'Prinzip' und 'Parameter' sind innerhalb der GB-Theorie nicht einheitlich definiert. Vgl. hierzu Haas (1993:13ff.) sowie Müller (1993:12ff.) u.v.a. die in Bondre-Beil (1994) vorgeschlagene Parametertheorie.

pien erstellt und auf ihre Wohlgeformtheit hin überprüft. Figur 1 stellt die übliche Verfahrensweise bei der GB-orientierten linguistischen Hypothesenbildung graphisch dar:<sup>2</sup>



Figur 1

Ausgehend von den zu untersuchenden sprachlichen Daten werden Hypothesen in Form von Baumstrukturen und von Prinzipien, von denen die Wohlgeformtheit dieser Strukturen abhängt, aufgestellt. Durch die Konfrontation von Strukturen und Prinzipien werden die Hypothesen evaluiert. Verifizierte Hypothesen, in Form von Struktu-

<sup>2</sup> Ich bedanke mich bei Michael Lahaye für die graphische Visualisierung unter Figur 1 und Figur 8.

ren und/oder Prinzipien, müssen festgehalten und bei der Überprüfung weiterer als Erweiterung des linguistischen Wissens mitberücksichtigt werden. Falsifizierte Hypothesen dagegen müssen als solche protokolliert werden, um bei der weiteren Arbeit ausgeschlossen zu werden, können aber auch ggfs. zum Zwecke weiterer Tests modifiziert werden.

Diese Arbeit wurde bisher in der Regel manuell, d.h. mit 'Papier und Bleistift', durchgeführt, was angesichts der Vielfalt der zu untersuchenden Phänomene, der Komplexität der Theorie und der oben bereits erwähnten Vielzahl der alternativen Hypothesen innerhalb der Theorie in vieler Hinsicht inadäquat ist. Während der Linguist einerseits kreativ sein muß, um die Probleme lösen zu können, muß er gleichzeitig seine Lösungen stets anhand eines komplizierten Geflechtes interagierender, modularisierter Prinzipien und durch die Konfrontation mit weiteren sprachlichen Daten auf ihre Korrektheit überprüfen, darf dabei die Übersicht aber nicht verlieren. Das manuelle Entwerfen der Baumstrukturen erweist sich hierbei als zeitraubend und unökonomisch; eine auf Papier (oder Tafel) erstellte Struktur ist nur unter Mühe veränderbar und wiederverwertbar. Die Evaluierung der sprachlichen Strukturen wird durch die (stets wachsende) Komplexität der Theorie und die Vielzahl der zu berücksichtigenden Alternativen oft erschwert und fehleranfällig. Vor diesem Hintergrund wurde das System GBX entwickelt, um durch den Einsatz computationaler Techniken die Arbeit des GB-orientierten Linguisten zu systematisieren und zu automatisieren und somit sowohl Quantität als auch Qualität der linguistischen Forschungsarbeit zu steigern.

#### 1.4 Modellierung der linguistischen Forschungsarbeit mit GBX als CAD-orientiertem Expertensystem

GBX ist ein interaktives graphikbasiertes Werkzeug zur Konstruktion linguistischer Theorien, das die Arbeit des Linguisten mit Hilfe von Techniken der Expertensystemtechnologie<sup>3</sup> und des Computer Aided Design (CAD)<sup>4</sup> modelliert. Das System ermöglicht einerseits die Erzeugung und die graphische Repräsentation sprachlicher Strukturen in Form von Baumstrukturen, andererseits die Implementation der linguistischen Prinzipien der GB-Theorie in einer einfachen Notation und die automatische Überprüfung der Wohlgeformtheit der graphisch dargestellten Strukturen anhand der implementierten Prinzipien in einer durch ihre Multifunktionalität flexibel gehaltenen Benutzeroberfläche. Somit unterstützt GBX, in Analogie zu CAD-Systemen, die man aus

---

<sup>3</sup> Als Expertensystem bezeichnet man im Bereich der Künstlichen Intelligenz ein Programm, das die Tätigkeit eines Experten auf einem bestimmten Gebiet übernimmt. Typische Anwendungsgebiete findet man z.B. in der medizinischen Diagnose oder in der Analysetechnik. Vgl. hierzu z.B. Bratko (1987:343), Wachsmuth (1993).

<sup>4</sup> Mit CAD bezeichnet man die "Unterstützung von Konstruktionsprozessen durch Datenverarbeitungssysteme" (vgl. Duden:129).

den Ingenieurwissenschaften kennt (vgl. hierzu z.B. Encarnaçao et al. 1990), jeden Entwicklungsschritt bei der Modellierung linguistischer Hypothesen:

- es übernimmt die graphische Darstellung der zu untersuchenden Strukturen und der Ergebnisse der Überprüfung der Wohlgeformtheit dieser Strukturen (Visualisierungsfunktion). Hierbei fördert es durch geeignete Konvertierungsfunktionen die Aufstellung neuer Hypothesen.
- es berechnet die Kompatibilität der graphisch dargestellten Hypothesen zu den implementierten linguistischen Prinzipien (Expertenfunktion),
- es ermöglicht dank Speicherung und Wiederverwertbarkeit von Strukturen und modular organisiertem linguistischem Wissen die Berücksichtigung alternativer Lösungsansätze (Datenbankfunktion).

Wie diese Funktionen in der Praxis zur Simulation und Teilautomatisierung, und somit zur effizienten Unterstützung der linguistischen Forschungsarbeit, interaktiv genutzt werden, wird – vereinfachend – in folgender Tabelle gezeigt:

	Linguist	GBX
Visualisierungsfunktion	<ul style="list-style-type: none"> <li>• beschreibt Baumstrukturen in Form von attribuierten Regeln</li> <li>• weist System an, Struktur mit bestimmten Prinzipien zu konfrontieren</li> <li>• weist auf bereits vorhandene Strukturen als Alternativen hin</li> </ul>	<ul style="list-style-type: none"> <li>• stellt die Baumstrukturen graphisch dar</li> <li>• macht Ergebnis der Konfrontation sichtbar</li> <li>• ermöglicht die teilautomatische Konvertierung von Strukturen</li> </ul>
Expertenfunktion	<ul style="list-style-type: none"> <li>• formuliert die linguistischen Prinzipien in einem objektorientierten, speziell für Linguisten geschaffenen Prolog-Dialekt</li> </ul>	<ul style="list-style-type: none"> <li>• überprüft die Wohlgeformtheit der Strukturen anhand des implementierten linguistischen Wissens</li> </ul>
Datenbankfunktion	<ul style="list-style-type: none"> <li>• erstellt Strukturen und Prinzipien</li> </ul>	<ul style="list-style-type: none"> <li>• speichert Strukturen und Prinzipien und stellt sie für weitere Tätigkeiten zur Verfügung</li> </ul>

Durch seine hohe Funktionalität fördert GBX die kontrastive Untersuchung sprachlicher Phänomene, sowohl im Sinne der Untersuchung von Sprachkontrasten als auch im Sinne der Berücksichtigung von theoretischen Kontrasten innerhalb der GB-Theorie. Darüber hinaus können, dank des in dem System integrierten objektorientierten Ansatzes, neue Lösungswege bei der syntaktischen Analyse eingeschlagen und ausgetestet werden, so daß GBX nicht nur die herkömmliche Forschungsarbeit innerhalb der GB-Theorie unterstützt, sondern auch neue Perspektiven im Hinblick auf eine interessante Erweiterung der Theorie eröffnet (vgl. hierzu v.a. Kapitel 4).

## 1.5 Abgrenzung von GBX zu Systemen ähnlicher Zielsetzung

GBX zeichnet sich dadurch aus, daß es vielfältige Funktionen in einer flexiblen graphischen Benutzeroberfläche verbindet und in einer Weise, die es mit CAD-Systemen vergleichbar macht, dem Linguisten zur Entwicklung und Evaluierung linguistischer Hypothesen zur Verfügung stellt. Die wichtigsten Funktionen dieses Systems sind die bereits erwähnten Visualisierungs-, Experten- und Datenbankfunktion. Charakteristisch für GBX sind weiterhin die Einfachheit der bei der Implementation linguistischer Prinzipien verwendeten linguistischen Programmiersprache, die eng an die GB-Notation angelehnt ist und eine direkte Überführung der linguistischen Prinzipien in eine prädikatenlogischen Klauseln ähnliche Notation ermöglicht, und schließlich die Übernahme des Konzepts der Objektorientierung in die linguistische Theorie

### 1.5.1 Linguistische Tools

In der Literatur wird eine Vielzahl von Systemen beschrieben, die eine bzw. mehrere Funktionen von GBX in verschiedenster Weise erfüllen. Darunter fallen beispielsweise Baumeditoren, wie *Arboreal for Windows* (1995) oder *RF Flow* (vgl. Broadwell 1995), sowie Grammatikentwicklungsumgebungen wie *TAGDevEnv* von Schifferer (1988) oder *GTU* von Volk (1994).

Baumeditoren haben lediglich den Teil der Visualisierungsfunktion, der die graphische Repräsentation der Strukturbäume betrifft (s. 2.3.3), mit GBX gemeinsam, sind aber mit einem linguistischen Werkzeug wie dem hier beschriebenen nicht vergleichbar.

Eine Übersicht über Entwicklungswerkzeuge für Grammatiken findet man in Backofen et al. (1993), der detaillierte Beschreibungen der jeweiligen Systeme enthält, sowie in Volk (1994:150ff.), der eine kurze Klassifizierung der wichtigsten Grammatikentwicklungsumgebungen entwirft. Eine ausführliche Beschreibung dieser vielfältigen Systeme erfolgt im Rahmen dieser Arbeit aufgrund ihrer unterschiedlichen Zielsetzungen nicht. Die meisten Systeme arbeiten nicht im Rahmen der GB-Theorie, was ihren

Vergleich mit GBX, dessen Konzeption stark GB-orientiert ist, wenig sinnvoll macht. Bei der Mehrzahl der z.T. doch GB-orientierten Systeme handelt es sich um Parser (und/oder Generatoren), die daher von vornherein aus dem Vergleich mit dem hier beschriebenen Werkzeug zur Theorieentwicklung und -evaluierung ausscheiden, bei dem es – selbst wenn die Ergebnisse der Forschungsarbeit mit GBX im Rahmen eines GB-orientierten Parsers/Generators natürlich genutzt werden können (und sollen, vgl. hierzu 2.1 und 2.8) – nicht um Probleme des GB-Parsings geht.

Zwei Ansätze kommen jedoch dem mit GBX verfolgten sehr nah, da sie zumindest über zwei der in GBX enthaltenen Funktionen verfügen, die Visualisierungs- und die Expertenfunktion. Ersterer ist der von Stabler (1988, 1992), zweiterer der von Felix & Palm (1994), die Sprache *CLAT*. Sie werden im folgenden kurz beschrieben und mit dem in der vorliegenden Arbeit vorgestellten Ansatz verglichen.

### 1.5.2 Stablers Implementation der GB-Theorie

Stabler (1988 und v.a. 1992) formalisiert die GB-Theorie, wie sie in *Barriers* (Chomsky 1986b) beschrieben wird, unter Berücksichtigung wichtiger Entwicklungen von Koopman & Sportiche (1988) über Pollock (1989) bis Chomsky (1989). Stablers Zielsetzung dabei ist die mathematische Beweisführung der von der GB-Theorie aufgestellten Axiome zur Überprüfung ihrer logischen Korrektheit, nicht zuletzt im Hinblick auf die Lösung parsingtheoretischer Probleme.

Stabler benutzt sowohl die Notation der *Prädikatenlogik erster Stufe* (*First Order Logic*)<sup>5</sup> als auch eine Standard-Prolog-Notation (*Edinburgh Prolog*)<sup>6</sup>, welche beide für Linguisten zumindest gewöhnungsbedürftig sind. Die in GBX eingebundene linguistische Programmiersprache LPS-Prolog dagegen ist ein speziell für GB-Linguisten entwickelter Prolog-Dialekt, dessen Notation sehr nah an die GB-Notation angelehnt ist (s. 2.4.2).

Die zu untersuchenden Baumstrukturen werden bei Stabler mit demselben Formalismus erzeugt wie die Prinzipien, die sich auf sie beziehen, was das Editieren ersterer sehr erschwert.

Die genaue Formalisierung der GB-Theorie führt zwar zur Präzisierung und folglich zur Überprüfung der linguistischen Aussagen (vgl. hierzu auch 2.4.1), eine Erweiterung der Theorie, wie sie in GBX durch die Unterstützung der interaktiven Hypothesenbildung und durch den objektorientierten Ansatz intendiert wird, wird in keiner Weise unterstützt.

Fraglich ist auch, ob das System aufgrund seiner relativen mathematischen Komplexität als linguistisches Werkzeug zur Theorieentwicklung und -evaluierung in der linguistischen Praxis eingesetzt werden kann.

<sup>5</sup> Zur *First Order Logic* Notation vgl. Owsnicki-Klewe (1993:10ff.).

<sup>6</sup> *Edinburgh Prolog* bildet den *de facto* Standard unter den zahlreichen Prolog-Implementationen. Seine Syntax wird in Clocksin & Mellish (1987:Appendix D) z.B. beschrieben.

### 1.5.3 Die Sprache *CLAT*

Die ursprünglich von Feldmeyer (1991) entwickelte Sprache *CLAT* (*Constraint Language for Attributed Trees*) dient der Formalisierung von Wohlgeformtheitsbedingungen für attribuierte Baumstrukturen und der Überprüfung der Wohlgeformtheit solcher Bäume. *CLAT* wurde von Palm (1993) erweitert, wobei besonderer Wert auf die Vereinfachung der Formalisierung, die modulare Organisation und die Parametrisierung der implementierten Prinzipien gelegt wurde. Diese erweiterte Version ermöglicht eine kompakte Formalisierung der Prinzipien der GB-Theorie und, gestützt durch die Möglichkeit der modularen Anordnung der Prinzipien, die Entwicklung von effizienten, gut strukturierten Prüfprogrammen für Strukturbäume (vgl. Palm 1993 und Felix & Palm 1994).

Ein gravierender Nachteil dieser Sprache liegt in ihrer primären Konzeption als Programmiersprache im herkömmlichen Sinne. *CLAT* greift auf informatische Mittel zurück wie die Benutzung zahlreicher Schlüsselwörter, wie man sie aus Programmiersprachen wie Modula-2 oder C kennt, was zu einer gewissen Überlastung des 'linguistischen' Codes führt, der letztlich nicht mehr GB-nah ist.

Da die Sprache über keinen Interpreter verfügt, muß der *CLAT*-Code, der die Implementation der linguistischen Prinzipien enthält, in Modula-2 bzw. C übersetzt werden. Erst nach der weiteren Übersetzung aller zum fertigen Programm gehörenden Moduln kann dieses zur Überprüfung eingegebener Bäume eingesetzt werden (vgl. Palm 1993:110). Obwohl dank der Modularisierung der Umfang des im Zuge einer Veränderung (der Theorie z.B.) neu zu übersetzenden Codes eingeschränkt werden kann, ist es klar, daß diese Arbeitsweise eine komfortable, dynamische Programmierung von Prinzipien stark beeinträchtigt.

Hinzu kommt, daß *CLAT* eine direkte interaktive graphikorientierte Konfrontation von Baumstrukturen und linguistischen Prinzipien, wie sie durch GBX ermöglicht wird, nicht unterstützt. Ein objektorientierter Ansatz wird mit *CLAT* nicht verfolgt.