1 LaTeX für Linguisten: IPA, Glossing und Baumstrukturen

1.1 Einleitung

In diesem Kapitel wird beschrieben wie Linguisten sich LaTeX zu Nutzen machen können, um anfallende fachspezifische Aufgaben aus den Teilbereichen *Phonetik*, *Morphologie* und *Syntax* schnell, sauber und formschön zu erledigen.

Zuerst wird gezeigt, wie das **IPA** in LaTeX benutzt werden kann um adäquate Lautrepräsentationen umzusetzen. Darauf folgend wird nach einer Erläuterung des besonders für die Morphologie relevanten **Glossings** mit LaTeX auf das Realisieren von **Baumstrukturen**, welche entscheidend dazu beitragen syntaktische Phänomene darzustellen, eingegangen. Einige Beispiele zeigen für jeden Anwendungsbereich konkrete Möglichkeiten der Verwendung auf und helfen einen schnellen Überblick zu gewinnen.

1.2 Installationshinweise

Gentoo [1] -Benutzer können Ebuilds für die LaTeX-Pakete *tipa*, *gb4e*, *qtree* und *covington* unter `http://homepage.rub.de/Alexander.Linke-2/linux/ebuilds/LaTeX/` herunterladen und sich den Aufwand einer manuellen Installation sparen.

Auch *FreeBSD*-User werden zumindest im Hinblick auf das *tipa*-Paket in den Ports fündig: der „Quellcodebauplan“ findet sich in „print/latex-tipa“.

Allgemeine Informationen zur Installation von LaTeX-Paketen können Kapitel ADD-REF-HERE entnommen werden - Microsoft Windows™-Benutzer sollten dieses Kapitel aufmerksam studieren, bevor sie sich an die Installation der hier vorgestellten Pakete begeben.

1.3 IPA

1.3.1 Einleitung

Das Internationale Phonetische Alphabet (IPA) wird in der *Phonetik* verwendet, um Laute einheitlich darzustellen. Jeder Laut wird durch ein entsprechendes Symbol repräsentiert. Mit Hilfe des Pakets *tipa.sty* können diese Symbole auch in LaTeX umgesetzt werden. Eine sehr ausführliche Dokumentation sowie das Paket selbst sind unter `http://www.l.u-tokyo.ac.jp/~fkr/` abrufbar.
Eingebunden wird das Paket wie gewohnt durch:

```
\usepackage{tipa}
```

[1] Eine GNU/Linux Metadistribution deren Paketsystem auf einem *BSD ähnlichen Portsansatz („Portage“) beruht - Nähere Informationen sind unter `http://www.gentoo.org` verfügbar.

1.3.2 Makro oder Umgebung?

Mit *tipa* steht ein ausgesprochen umfangreiches und vielfältiges Paket zur Darstellung von IPA-Zeichen zur Verfügung. Fast alle Zeichen können auf mehr als nur eine Weise referenziert werden, nämlich entweder durch ein parameterfreies, zeichenspezifisches Makro, ein Makro für eine längere Zeichenkette oder eine eigene Umgebung. Die folgenden drei Beispiele sind in Folge dessen bedeutungsgleich und liefern somit das gleiche Ergebnis: 'alphabet' in phonetischer Trankription ([ælfəbɛt]).

```
[\ae{}lf\textschwa{}b\textepsilon{}t]        % einzelne Makros
\textipa{[\ae{}lf@bEt]}                     % "Makrogruppierung"
\begin{IPA}[\ae{}lf@bEt]\end{IPA}           % Umgebung
```

Bei längeren Transkriptionspassagen ist es empfehlenswert, entweder `\textipa` zu verwenden oder gleich in eine IPA-Umgebung zu wechseln - für wenige zu transkribierende Zeichen bietet sich die intuitivere Makroschreibweise an.

1.3.3 Makro- und Umgebungsübersicht

Die in der Tabelle abgebildeten Symbole stellen nur einen Ausschnitt der durch das IPA definierten Lautentsprechungen dar - allerdings handelt es sich bei diesen um die erfahrungsgemäß am häufigsten benötigten. Eine 26 (!) Seiten lange Liste aller in *tipa* implementierten Symbole kann bei Bedarf der Dokumentation des Pakets entnommen werden.

Die vorgenommene Zusammenstellung der Symbole der obenstehenden Tabelle basiert auf der in *Davenport/Hannahs* angeführten Liste [2].

Generell können die vom *tipa*-Paket zur Verfügung gestellten Makros `\*`, `\;`, `\:` und `\!` benutzt werden, um die Eingabe von IPA-Symbolen abzukürzen, die kein eigenes Kürzel haben [3]. Die folgende Übersicht soll ihre Anwendung exemplarisch verdeutlichen:

`\*`	umgedrehtes Symbol	f,k,r,t,w	ɟ,ʞ,ɹ,ɻ,ɯ
	Spezialsymbol	j,n,h,l,z	ʝ,ɲ,ħ,ɬ,ʒ
	normaler Font	alle restlichen	a,b,c
`\;`	Kapitälchen	A,B,H,L,E,...	ᴀ,ʙ,ʜ,ʟ,ᴇ
	(funktioniert nicht mit allen Buchstaben!)		
`\:`	Retroflexe	d,l,n,r,s,z	ɖ,ɭ,ɳ,ɽ,ʂ,ʐ
`\!`	Implosive/Clicks	b,d,g,j,G,o	ɓ,ɗ,ɠ,ʄ,ɢ,ʘ

Spezielle Belegungen der Makros sind der nachfolgenden Tabelle zu entnehmen.

[2] Siehe *Davenport/Hannahs*, „Introducing phonetics and phonology“, Arnold: 1998
[3] Siehe auch: „Anmerkungen“, Seite 4

Sym.	Makro	Umgeb.
Vorderzungenvokale		
i	i	i
y	y	y
ɪ	\textsci	I
ʏ	\textscy	Y
e	e	e
ø	\o	\o
ɛ	\textepsilon	E
œ	\oe	\oe
æ	\ae	\ae
a	a	a
ɶ	\textscoelig	\OE
Mittelzungenvokale		
ɨ	\textbari	1
ʉ	\textbaru	0
ɘ	\textreve	9
ɵ	\textbaro	8
ə	\textschwa	@
ɜ	\textrevepsilon	3
ɞ	\textcloserevepsilon	
ɐ	\textturna	5
ɯ	\textturnm	M
u	u	u
ʊ	\textupsilon	U
Hinterzungenvokale		
ɣ	\textbabygamma	G
o	o	o
ʌ	\textturnv	2
ɔ	\textopeno	O
ɑ	\textscripta	A
ɒ	\textturnscripta	

Konsonanten: Plosive		
p	p	p
b	b	b
t	t	t
d	d	d
ʈ	\textrtailt	\:t
ɖ	\textrtaild	\:d
c	c	c

Sym.	Makro	Umgeb.
ɟ	\textbardotlessj	
k	k	k
ɡ	\textscriptg	g
q	q	q
ɢ	\textscg	\;G
ʔ	\textglotstop	P
Nasale		
m	m	m
ɱ	\textltailm	M
n	n	n
ɳ	\textrtailn	\:n
ɲ	\textltailn	
ŋ	\textipa{N}	N
ɴ	\textscn	\:N
Trills		
ʙ	\textscb	\;B
r	r	r
ʀ	\textscr	\;R
Taps/Flaps		
ɾ	\textfishhookr	R
ɽ	\textrtailr	\:r
Frikative		
ɸ	\textphi	F
β	\textbeta	B
f	f	f
v	v	v
θ	\texttheta	T
ð	\textipa{D}	D
s	s	s
z	z	z
ʃ	\textesh	S
ʒ	\textyogh	Z
ʂ	\textrtails	\:s
ʐ	\textrtailz	\:z
ç	\c{c}	\c{c}
ʝ	\textctj	J
x	x	x
ɣ	\textgamma	G
χ	\textchi	X
ʁ	\textinvscr	K

Sym.	Makro	Umgeb.
Frikative		
ħ	\textcrh	
ʕ	\textrevglotstop	Q
h	h	h
ɦ	\texthth	H
Laterale Frikative		
ɬ	\textbeltl	
ɮ	\textOlyoghlig	
Approximanten		
ʊ	\textscriptv	V
ɹ	\textturnr	
ɻ	\textturnrrtail	\:R
j	j	j
ɰ	\textturnmrleg	
Laterale Approximanten		
l	l	l
ɭ	\textrtaill	\:l
ʎ	\textturny	
ʟ	\textscl	\;L

Sym.	Makro	Umgeb.
Suprasegmentalia		
ˈ	\textprimstress	"
ˌ	\textsecstress	""
ː	\textlengthmark	:
ˑ	\texthalflength	;
Diakritika		
x^{h}	x^{h}	x\super h
x̼	\textseagull{x}	\\|m{x}
x̟	\textraising{x}	\\|´x
x̠	\textlowering{x}	\\|‘x
x̺	\textsubring{x}	\r*x
x̻	\textsubwedge{x}	
x͡y	\textbottomtiebar{xy}	\t*{xy}
x͡y	\texttoptiebar{xy}	\t{xy}
x̩	\textsyllabic{x}	\s{x}
x̯	\textsubarch{x}	
x̂	\textroundcap{x}	\\|c{x}

1.3.4 Anmerkungen

- Das Makro \textipa kann durch den kürzeren Befehl \ipa ersetzt werden - so lässt sich bei langen Texten etwas Tipparbeit einsparen ;)

```
\let\ipa\textipa
```

- Sollte es zu Problemen mit den Makros *, \:, \; und \! kommen, so empfiehlt es sich, *tipa* auf eine sichere Weise einzubinden:

```
\usepackage[safe]{tipa}
```

  Oben genannte Makros sind nämlich bereits durch LaTeX definiert und werden von *tipa* umdefiniert, um einen schnelleren Zugriff auf die IPA-Symbole zu ermöglichen - die ursprüngliche Belegung der Makros ist daher nicht mehr verfügbar. Dieses potentielle Problem wird durch das gezeigte Einbinden von *tipa* mit 'safe' als Option behoben - allerdings sind die betreffenden Makros in diesem Fall nicht mehr durch *tipa*-Funktionalität belegt!

- *tipa* beherrscht nicht nur das IPA, sondern kann zudem zur Darstellung von Tonverläufen benutzt werden. Nähere Informationen finden sich in Kapitel 3.2.8 der *tipa*-Dokumentation.

- Es stehen mehrere Schriftarten zur Auswahl:

Roman (Standard)	`\textipa{ziN@n}`	ziŋən
Slanted	`\textsl{\textipa{ziN@n}}`	*ziŋən*
Bold	`\textbf{\textipa{ziN@n}}`	**ziŋən**
Sans Serif	`\textsf{\textipa{ziN@n}}`	ziŋən
Typewriter	`\texttt{\textipa{ziN@n}}`	ziŋən

1.4 Glossing

1.4.1 Was ist Glossing?

Glossing bezeichnet in der linguistischen Fachterminologie den Vorgang der Segmentierung von sprachlichen Entitäten in ihre bedeutungstragenden Einheiten [4] und deren jeweilige funktionale und/oder semantische „Übersetzung". Das Glossing kommt aber nicht nur in der *Morphologie* zur Anwendung, sondern gleichwohl in den benachbarten Disziplinen der *Phonologie* und *Syntax*. Ein morphologisches Beispiel:

(1) Kolja durak. (*Russisch*)
Kolja-*Nom* fool-*Nom*
'Kolja is a fool.'

1.4.2 Installation und erste Schritte

Die hier vorgestellten Methoden zur Erstellung von Glossen basieren auf dem Paket *gb4e.sty* das auf *CTAN* zum Download bereit steht. Eine alleinige Installation dieses Pakets ist allerdings nicht ausreichend: es bindet das Paket *cgloss4e.sty* ein - hängt also von diesem ab. Auf *CTAN* wird man jedoch auch in diesem Falle fündig.

Eine noch ausführlichere Dokumentation des Pakets liegt unter folgender URL bereit: `http://www.ling.ohio-state.edu/~dm/02/latex-tutorial/sources/styles/gb4e`. Nachdem die Installation erledigt wurde, muss das Paket nun nur noch eingebunden werden:

```
\usepackage{gb4e}
```

[4] *Morphem*: Kleinste **bedeutungstragende** Einheit.

1.4.3 Befehlsübersicht

exe	Umgebung	Von dieser Umgebung werden **alle** weiteren zum Glossing gehörenden Befehle/Umgebungen umschlossen.
xlist	Umgebung	Diese Umgebung kann benutzt werden, um Glossierungen zu schachteln, d.h. nicht nur ganzzahlig nummerierte Glossen zu verwenden, sondern in diesen noch Unterabschnitte zu definieren, welche nach den in LaTeX üblichen Konventionen gehandhabt werden. Sämtliche Befehle, die in der **exe**-Umgebung verwendet werden dürfen, sind auch hier erlaubt.
ex	Befehl	Erstellt ein Beispiel mit (fortlaufender) Nummerierung. Als *optionale Argumente* können sowohl „?“ alsauch „∗“ übergeben werden, welche in der Regel benutzt werden, um einen Satz als ungrammatisch, bzw. fragwürdig zu markieren.
sn	Befehl	Erzeugt ein neues Beispiel ohne Nummerierung. (Ansonsten wie **ex**)
gll/glll	Befehl	Mit diesem Befehl wird angegeben, wie viele Zeilen (***l****ines*) die Glosse in Anspruch nehmen wird. **gll** reserviert den Platz für zwei, **glll** für drei Zeilen. Eine mit dem **trans**-Befehl beginnende Zeile wird nicht mitgezählt. Die einzelnen Entitäten der einzelnen Linien werden sauber untereinander gesetzt („Tabbing“), sodass die relevanten Merkmale der Glossierung besonders ins Auge fallen können.
trans	Befehl	**trans** erzeugt eine für Übersetzungen reservierte Zeile deren Anfang an den der vorangegangenen **gll/glll**-Zeilen angepasst ist.

1.4.4 Beispielanwendung

(2) a. Marry had a little lamb.
Marry-*Nom-SG* have-*3sg-Prät* a-*INDEF* little lamb-*Akk-SG*
„Marry hatte ein kleines Lamm.“

b. Marry likes the little lamb.
Marry-*Nom-SG* like-*3sg-Präs* the-*DEF* little lamb-*Akk-SG*.
Marry mag das klein Lamm.
„Marry mag das kleine Lamm.“

Um diese Glossierung zu erzeugen ist folgender LaTeX-Quellcode erforderlich:

```
\begin{exe}
\ex
\begin{xlist}
\ex
 \gll Marry had a little lamb.\\
 Marry-\emph{Nom-SG} have-\emph{3sg-Prät} a-\emph{INDEF} little
    lamb-\emph{Akk-SG}\\
 \trans "‘Marry hatte ein kleines Lamm."’\\

\ex
 \glll Marry likes the little lamb.\\
 Marry-\emph{Nom-SG} like-\emph{3sg-Präs} the-\emph{DEF} little
    lamb-\emph{Akk-SG}.\\
 Marry mag das klein Lamm.\\
 \trans "‘Marry mag das kleine Lamm."’\\
\end{xlist}
\end{exe}
```

Kann eine aus mehreren Wörtern bestehende Phrase nur durch ein einzelnes Wort glossiert werden, so kann ein **Platzhalter** eingesetzt werden. Genauso muss eine mehrgliedrige Glossierung durch geschweifte Klammern gruppiert werden.
Beispiel:

(3) I bought an appetizer in the shopping mall.
Ich kaufte ein appetitanregendes Mittel im Einkaufszentrum .

```
 \begin{exe}
  \ex
   \gll I bought an appetizer in the shopping mall.\\
   Ich kaufte ein {appetitanregendes Mittel} im {}
    Einkaufszentrum {}.\\
 \end{exe}
```

Da es erforderlich ist auch ungrammatische Sätze zu Beispielzwecken anzuführen, hat sich die Konvention etabliert, diesen Sätzen ein „$*$“ voranzustellen. In *tipa* lässt sich dies wie folgt umsetzen.

(4) * These is sentence terrible wrong.
Diese ist Satz schrecklich falsch.

```
\begin{exe}
 \ex[*] { \gll These is sentence terrible wrong.\\
               Diese ist Satz schrecklich falsch.\\}
\end{exe}
```

1.4.5 Anmerkungen

- Am Ende einer jeden Zeile ist es wichtig an den **Zeilenumbruch** zu denken, da sonst Fehler autreten und die Glossierung nicht richtig formatiert werden kann.

1.4.6 Links zu weiteren Paketen

- **Covington**: `http://www.ai.uga.edu/~mc/info.html#LATE`
  Auf der Website von Prof. Michael Covington findet sich ein in komprimierter Form vorliegendes Archiv *covington.zip* samt Dokumentation. Das *covington* Paket kann vielseitig eingesetzt werden - z.B. zum Setzen von Beispielen, einfachen Merkmalsstrukturen und Diskursrepräsentationsstrukturen.

1.5 Baumstrukturen

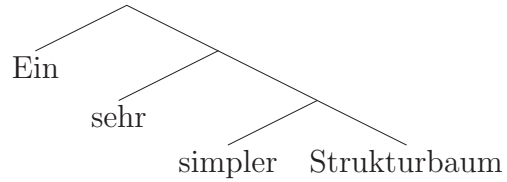

Abbildung 1: Ein sehr simpler Strukturbaum

1.5.1 Einleitung

Strukturbäume als Spezialform graphischer Darstellung sprachlicher Strukturen sind besonders für syntaktische Untersuchungen ein weit verbreitetes Hilfsmittel. Aber auch zur Verdeutlichung morphologischer Strukturen lässt sich diese Darstellungsart oftmals gewinnbringend einsetzen.

Mit dem auf *CTAN* verfügbaren LaTeX-Paket *qtree* lassen sich verschiedenste Strukturbäume relativ einfach und dennoch formschön setzen.

1.5.2 Ein erstes Beispiel

Im Gegensatz zu Abbildung 1 ist im folgenden Beispiel bereits einiges an syntaktischer Information ergänzt worden: Die Knoten weisen nun **Bezeichner** auf und der Strukturbaum wird anders aufgebaut.

Der Baum in Abbildung 2 ist im Vergleich zu anderen Paketen [5] sehr schnell und einfach generiert, sodass die Konzentration weniger auf das Setzen an sich gerichtet werden muss.

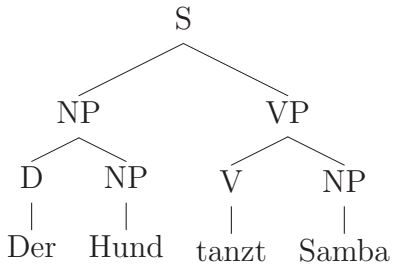

Abbildung 2: Ein erstes Beispiel

Der folgende Quellcode zu Abbildung 2 verdeutlicht die Simplizität der von *qtree* geforderten Syntax:

```
\usepackage{qtree}
...
\begin{document}
...
\Tree
[ .S
 [ .NP
  [ .D Der ]
  [ .NP Hund ]
 ]
 [ .VP
  [ .V tanzt ]
  [ .NP Samba ]
 ]
]
```

Die Einrückungen und Zeilenumbrüche sind nicht wirklich notwendig - es wäre demzufolge auch problemlos möglich die komplette Baumstruktur mit nur einer Zeile zu generieren. Für die Übersichtlichkeit ist es aber ratsam, möglichst strukturierten Code zu produzieren - gerade bei komplexen Bäumen verliert man sonst leicht den Überblick.

qtree verwendet keine Umgebung, sondern stellt den Makro-Befehl `\Tree` zur Verfügung, dem alle Komponenten des späteren Baums als Parameter übergeben werden.

[5] Siehe „Links zu weiteren Paketen“, Kapitel 1.5.7, Seite 14

Eckige Klammern dienen dem „Schachteln“ der einzelnen Komponenten in der Baumstruktur. Der jeweilige direkt auf den initialen *Punkt* folgende String wird als Bezeichner für den Knoten gesetzt. Öffnende und schließende eckige Klammern bilden einen Knoten.

Obwohl die Verwendung des `\Tree`-Makros simpel ist, empfiehlt es sich, den Strukturbaum zuerst einmal auf einem Blatt zu notieren und dann erst in den Quellcode aufzunehmen. Der einfachste Weg diese zu Erreichen besteht darin, vom obersten Knoten (S) zum untersten linken Knoten zu wandern und daraufhin Teilbaum für Teilbaum schließend fortzufahren.

Soll mit einem Knoten ein Bezeichner assoziiert werden, so wird dieser entweder direkt nach der öffnenden eckigen Klammer eingefügt, oder aber nach der schließenden Klammer. Beide Möglichkeiten werden im folgenden exemplarisch dargestellt und sind bedeutungsgleich:

```
\Tree [ .S Wort1 [ .XP Wort2 Wort3 ] ]
\Tree [ .S Wort1 [ Wort2 Wort3 ] .XP ]
```

Der erste Knoten muss allerdings innerhalb der ersten Klammer benannt werden.

1.5.3 Kurzdarstellungen von Phrasen

Abbildung 3: Detaillierte/Kurze Darstellung einer *NP*

Im linken Beispiel aus Abbildung 3 ist der detaillierte Baum zu sehen, der die *NP* komplett darstellt, wohingegen sich im rechten Beispiel eine deutlich verkürzte Darstellungsweise der gleichen syntaktischen Struktur findet.
Erzeugen lassen sich derartige Verkürzungen durch das Makro `qroof`, dem der zu umfassende Text als Parameter übergeben wird. Um dies zu verdeutlichen folgt nun der Code für beide Beispielstrukturbäume aus Abbildung 3.

```
%%% Detaillierte Darstellung %%%
\Tree
[ .S
 [ .NP [ .D Die ] [ .N Katze ] ]
 [ .VP [ .V ist ] [ .Adj müde ] ]
]

%%% Kurze Darstellung %%%
\Tree
[ .S
 \qroof{Die Katze}.NP
 [ .VP [ .V ist ] [ .Adj müde ] ]
]
```

1.5.4 Modifikatoren des Baumstrukturlayouts

qtree verwendet zur Darstellung von Bäumen ein akzeptables Defaultlayout. Was aber, wenn man aus einem speziellen Grund davon abweichen möchte?

Zum Glück bringt *qtree* auch zu diesem Zweck einige Makros mit unter deren Zuhilfenahme sich das Layout nach eigenen Präferenzen modifizieren lässt. Diese Makros werden im Folgenden vorgestellt:

qsetw	Abstand zwischen zwei Knoten festlegen. Erwartet den Abstand als Parameter. Beispiel: `!\qsetw{.1cm}`
qtreecenterfalse	Schaltet das automatische Zentrieren des Strukturbaums ab. Es wird kein Parameter benötigt. Beispiel: `\qtreecenterfalse`
qtreecentertrue	Schaltet das automatische Zentrieren (wieder) ein. Es wird kein Parameter benötigt. Beispiel: `\qtreecentertrue`

Manche der oben vorgestellten Makrobefehle wird der durchschnittliche Anwender vielleicht selten verwenden - jeder jedoch, der einen komplexen Strukturbaum generieren muss, wird auf lange Sicht nicht auf `\qsetw` verzichten können, wenn die Bäume einem gewissen optischen Anspruch genügen sollen. Dies sei durch ein Beispiel erläutert:

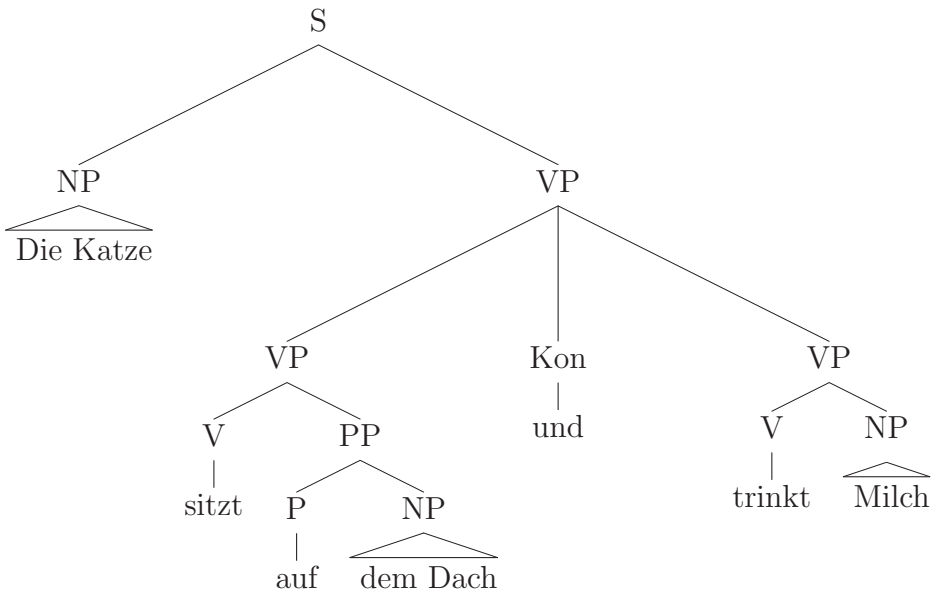

Abbildung 4: „Roher“ Strukturbaum

```
\Tree
[ .S
 \qroof{Die Katze}.NP
 [ .VP
  [ .VP [ .V sitzt ] [ .PP [ .P auf ]  \qroof{dem Dach}.NP  ] ]
  [ .Kon und ]
  [ .VP [ .V trinkt ] \qroof{Milch}.NP ]
 ]
]
```

Der Strukturbaum aus Abbildung 4 stellt zwar die relevanten Informationen dar, allerdings sieht er nicht sonderlich ansprechend aus. Die Kanten der oberen Knoten sind proportional betrachtet viel zu lang wenn man sie mit denen der unteren Knoten in Vergleich setzt. Um ein schönes Ergebnis zu bekommen, kann mit ein wenig Feinarbeit nachgeholfen werden:

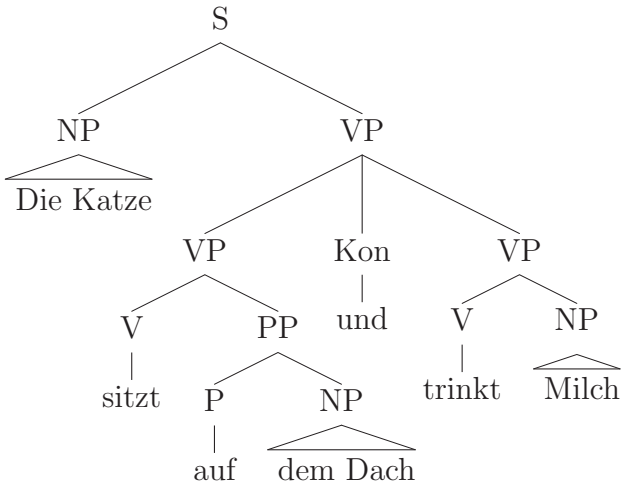

Abbildung 5: Strukturbaum mit modifizierten Abständen

```
\Tree
[ .S
 \qroof{Die Katze}.NP
 !\qsetw{.2cm}
 [ .VP
  [ .VP [ .V sitzt ] [ .PP [ .P auf ]  \qroof{dem Dach}.NP  ] ]
  !\qsetw{.2cm}
  [ .Kon und ]
  [ .VP [ .V trinkt ] \qroof{Milch}.NP ]
 ]
]
```

Der Abstand der Knoten *NP* und *VP* unterhalb des Satzknotes *S* wurde durch den Befehl `!\qsetw{.2cm}` vom Defaultwert 1 cm auf nur noch 0.2 cm verringert. Dadurch gewinnt der Strukturbaum ein geordneteres Aussehen und erscheint symmetrischer.

1.5.5 Eingebettete Glossierungen

```
\Tree
[ .Adj
 [ .Neg\textsuperscript{Pre} un-\\un- ]
 [ .Adj
  [ .V believ(e)\\glaub(en) ]
  [ .Adj\textsuperscript{Suf} -able\\-lich ]
 ]
]
```

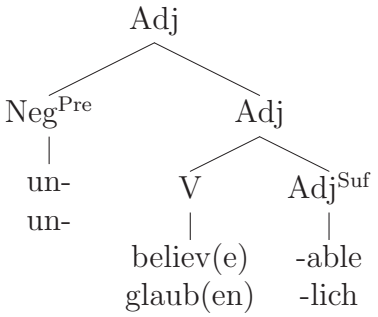

Abbildung 6: Eingebettete Glossierung

Die in Abbildung 6 dargestellte eingebettete Glossierung wird durch einen einfachen Zeilenumbruch innerhalb einer Klammer realisiert. Dabei ist jedoch peinlich genau darauf zu achten, dass zwischen den Entsprechungen kein Leerzeichen steht.

1.5.6 Anmerkungen

- Alle bekannten schönen Kleinigkeiten wie Textsuper- und Subskripte, Zeichen aus dem Mathematik-Modus und IPA-Symbole können auch in Bäume eingebettet werden.

1.5.7 Links zu weiteren Paketen

- **ps-trees**: `http://www2.sfs.nphil.uni-tuebingen.de/~wolfgang/`
  Bei *ps-trees* handelt es sich um ein von Prof. Dr. Wolfgang Sternefeld entwickeltes LaTeX -Paket zum Erstellen von Baumstrukturen. Es beherrscht nicht nur alle Features von *qtree*, sondern kann mit einigen weiteren nützlichen Funktionen glänzen, wie z.B. der Darstellung von Silbenstrukturen. Allerdings ist die zugrundeliegende Syntax bei weitem anspruchsvoller, so dass für Standardbäume *qtree* als die (für den Neuling) bessere Wahl erscheint. Für wirklich große und komplexe Bäume jedoch dürfte man an *ps-trees* kaum vorbeikommen.

- **ecltree**: `http://ling.ohio-state.edu/~dm/02/latex-tutorial/sources/styles/`
  Dieses von Hideki Isozaki entwickelte Paket ermöglicht ebenfalls das Generieren von Strukturbäumen, bietet allerdings noch einige weitere Möglichkeiten, wie z.B. das Festsetzen von Höhen- und Breitenabständen der einzelnen Knoten. Die Syntax ist komplizierter als die von *qtree*, aber immer noch sehr logisch aufgebaut.

- **pst-jftree**: `http://www.math.neu.edu/ling/tex/`
  Mit *pst-jftree* steht ein weiteres Paket zur Verfügung, welches auf *PSTricks* aufbauend alle Makros bereitstellt, um selbst komplexesten Anforderungen gerecht werden zu können. So können z.B. verschiedenste Pfeile in Strukturbäume eingefügt werden um Bewegungen anzuzeigen, Teilbäume können eingekreist und einzelne Linien in doppelter Stärke dargestellt werden um Dominanz darzustellen. Die Syntax ist jedoch dementsprechend anspruchsvoll, so dass der geneigte Benutzer

dieses Pakets mit einiger Einarbeitungszeit rechnen und die Bereitschaft mitbringen sollte, sich auf mit *PSTricks* auseinanderzusetzen. Die unter oben genannter URL zugängige Dokumentation ist aber relativ umfangreich und ermöglicht es, sich einen ersten Überblick über die Vielfalt an Möglichkeiten zu verschaffen.