



Corpusbasierte Intonationsforschung

Etikettieren und Abfragen
prosodischer Labels in EMU/R

Corpora

- Sammlung zusammenhängender Daten
 - z.Bsp. Sprachdatensammlungen
 - Sprachsignale / abgeleitete Signale
 - Etikettierungen
- Daten
 - durch Zahlen und Buchstaben diskret abgebildeter Teil der realen Welt (Nachrichten und Zeichen)
 - Signale digitalisieren (Abtasten und Quantifizieren)
 - Signalstücke etikettieren

Corpora/Datenbanken

■ Organisation in Datenbanken

„Eine Datenbank ist eine selbständige, auf Dauer und für flexiblen und sicheren Gebrauch ausgelegte Datenorganisation, die einen Datenbestand (Datenbasis) und die dazugehörige Datenverwaltung (Datenbankverwaltungssystem=DBMS) umfasst.“
Zehnder(2005:35)

Das DBMS ist ein leistungsfähiges Programm für die flexible Speicherung und Abfrage strukturierter Daten“ Zehnder(2005:35)

heutige Datenbankbeispiele

- **Datenbasis:**
 - Kiel Corpus
 - GToBI Corpus
- **Datenbankverwaltungssystem:**
 - EMU Speech Database System
 - erweiterte Analysemöglichkeiten durch EMU/
R einer Schnittstelle zur R Programming
Language

Intonation

- Tonhöhenverlauf über die Zeit

„Intonation has traditionally been regarded as a problem“ Fox (2000:269)

- es gibt Schwierigkeiten Intonation durch Zeichen abzubilden und zu diskretisieren
- kein Beschreibungsverfahren hat sich allgemein durchgesetzt
- vorhandene Beschreibungsverfahren = Modelle

Fox, A.(2000): Prosodic Features and Prosodic Structure: the Phonology of Suprasegmentals. Oxford:O.U.P.

Modelle

- Modelle:

„[...] die Darstellung eines Untersuchungsgegenstandes“¹

„A model is an abstraction of reality [...]“²

¹Churchman, C.W. et al.(1971):Operations research: eine Einführung in die Unternehmensforschung.Oldenbourg: Wissenschaftsverlag.S.135.

²Hazelrigg, G. A.(2000): Introduction. In Carl W. Hall: Laws and models. Science, engineering and technology. Boca Raton, Fla.: CRC Press.S.vii.

Intonationsmodelle

- konturbasierte Modelle (Chrystal 1969; Halliday 1970; ...)
 - Beschreibung der Intonationsverläufe

Kielers **I**ntonations **M**odell

- Ebenen-Modelle (Tone-Level) (Pike, 1945)
 - Beschreibung der Tonlagen bestimmter strukturell relevanter Silben

German **T**ones and **B**reak **I**ndices

Inhalt

- Corpora:

GToBI DB

Kiel Corpus

- Intonationsmodelle:

GToBI

KIM

- leistungsfähige Programm (DBMS) für ...
die **Abfrage strukturierter** Daten:

- (die Daten)
- Struktur der Daten
- Abfrage der Daten

Emu Speech Database System

- **Datenbankspeicherung**
 - lokal oder serverbasiert außerhalb des EMU Systems
 - Sicherung der Daten und der Zugriffe durch System organisiert
- **Datenbankstruktur**
 - ggb. durch eingeschränkte Datenbanktemplate
 - Speicherorte der Signale und Etikettierungen
 - Struktur der Etikettierung
 - Darstellungsoptionen
- **Datenbankorganisation**
 - über Zugriff auf Template und Templateinhalte

DBMS

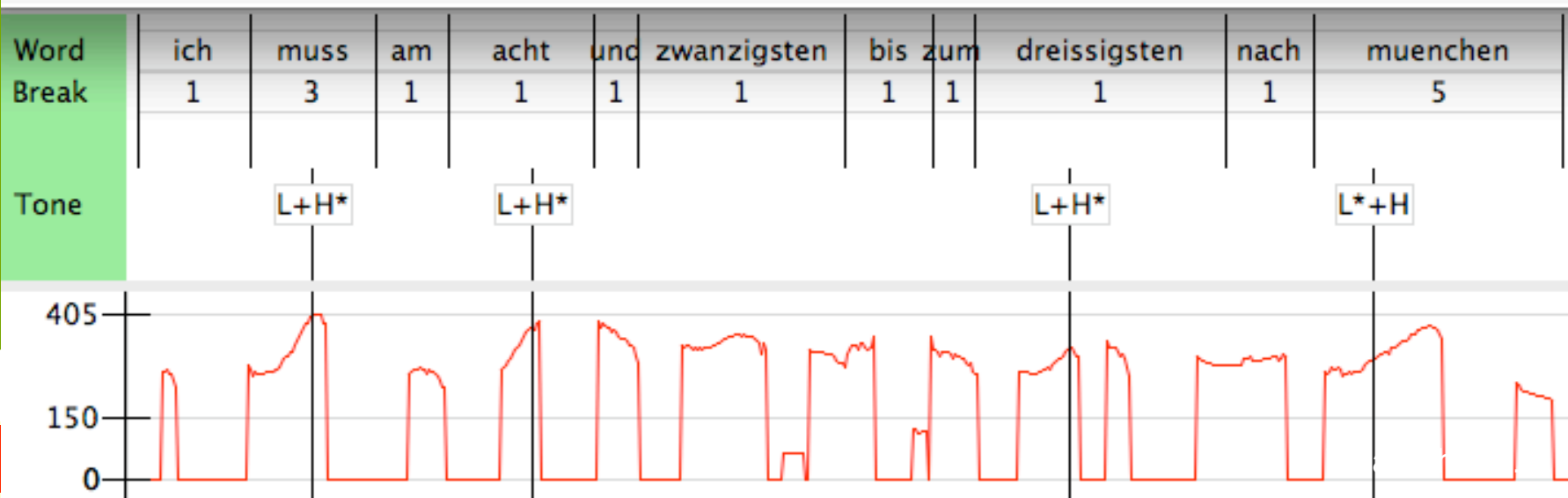
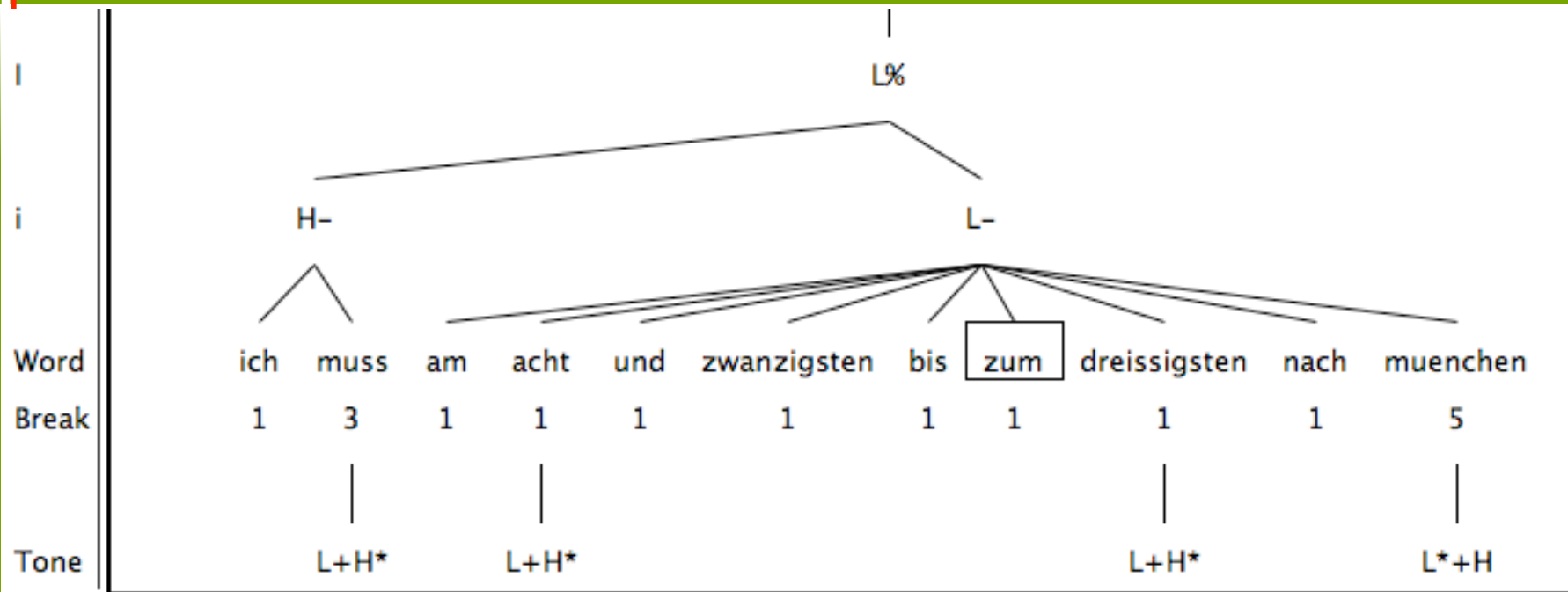
- realisiert durch:
 - EMU-Core & EMU Applikationen
 - EMU/R
 - R
- Datenbankfunktionen
 - Darstellung
 - Funktionalität auf Datenbanken
- Datenbankabfragen
- Analysefunktionen

Beispiel – GToBI DB

- Darstellung: Labeller
- über Abfrage in R:

```
library(emu);  
phrase = emu.query("gt", "*", "i != x")  
phrase.toene=emu.requery(phrase, "i", "Tone")  
phrase.f0 = emu.track(phrase, "F0")  
par(mfrow = c(3,5))  
for (i in 1:nrow(phrase)) {  
  dplot(phrase.f0[i,],  
        main = paste(label(phrase.toene)[i],  
                      utt(phrase)[i],  
                      sep="\n")  
}
```

Darstellung

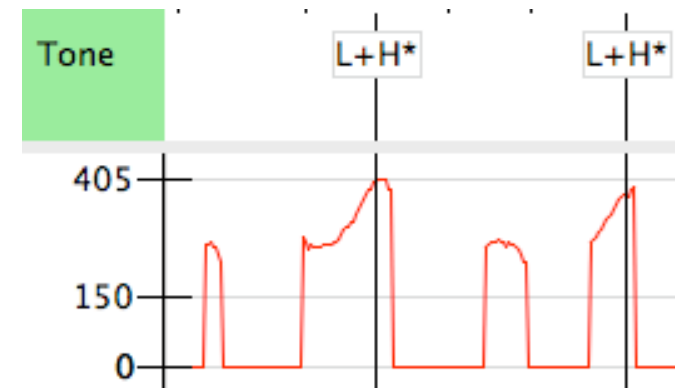


Etikettierungsstruktur

- sequenziell: ein Label folgt einem anderen auf einer Ebene

- zeitgebundene Ebene

- SEGMENT (Intervalle)
- EVENT (Ereignisse)



- zeitlose Ebenen



Verbindungen/Links

- zwischen Ebenen

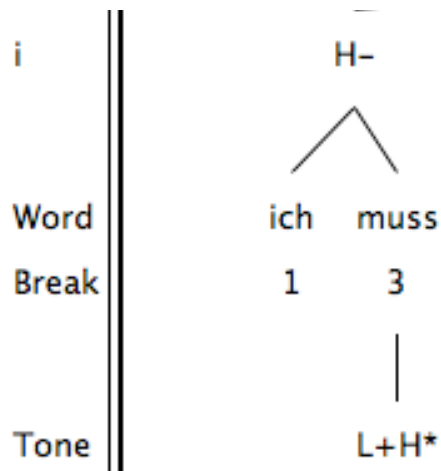
- linear:

- Parallelebene / Labellinks

Word Break	ich	muss	am	acht
	1	3	1	1

- zu jedem Label gibt es ein Label auf der anderen Ebene und umgekehrt.

- hierarchisch:



- ein Label auf einer Ebene (hyper) dominiert ein oder mehrere Labels auf einer zweiten Ebene (hypo)

Datenbankabfrage

- Emu Query Language (EQL)
 - Alphabet, Syntax und Semantik
- ist an die Etikettierungsmöglichkeiten angepasst:
 - einfache Abfragen
 - Abfragen von linearen Verbindungen
 - Abfragen von nichtlinearen Verbindungen
 - Positionsabfragen
 - Reihenfolgenabfragen (Sequenzen)
 - Anzahlabfragen

EQL - Alphabet

- Operatoren:
 - für einfachen Abfragen: = !=
 - innerhalb einfacher Abfrage: | &
 - zwischen Abfrage: ^ ->
- Funktionen und ihre Operatoren:
 - Start() End() < >
 - Medial() <= >=
 - Num() =
- Zeichen mit besonderer Funktion [] , ' #
- Buchstaben und Zahlen

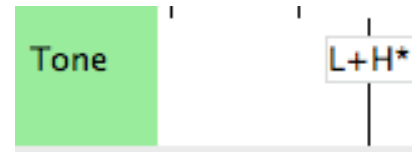
EQL – Syntax & Semantik

- Bildung einfacher Abfragen: =

- $LEV = LAB$

$Tone = L+H^*$

- Finde alle Segmente mit dem Label LAB auf der Ebene LEV



- $LEV = LAB1 \mid LAB2 \mid LAB3 \dots$

$Tone = L \mid L^*$

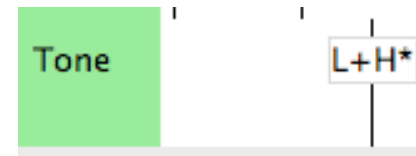
- Finde alle Segmente mit dem Label LAB1 oder LAB2, ... auf der Ebene LEV

EQL – Syntax & Semantik

- Bildung einfacher Abfragen: !=

- $LEV \neq LAB$

$Tone \neq H$



- Finde alle Segmente auf der Ebene LEV, die nicht das Label LAB haben

- $LEV \neq LAB1 \ \& \ LEV \neq LAB2 \ \dots$

$Tone \neq H \ \& \ Tone \neq H^*$

- Finde alle Segmente auf der Ebene LEV, die nicht das Label LAB1 oder LAB2 haben

EQL – Syntax & Semantik

- Erweitern einfacher Abfragen um lineare Verbindungen: &

Word	muss
Break	3

- $LEV1 = LAB1 \ \& \ LEV2 = LAB2$

$Break = 3 \ \& \ Word = muss$

- Finde alle Segmente mit dem Label LAB auf der Ebene LEV1, die mit einem Label LAB2 auf der Ebene LEV2 verbunden ist
 - für & müssen die Ebenen eine lineare Verbindung haben

EQL – Syntax & Semantik

- nicht lineare Verbindungen abfragen durch Verknüpfen einfacher Abfragen:

[EA ^ EA]

- [LEV1 = LAB1 ^ LEV2 = LAB2]

Word	muss
Break	3
Tone	L+H*

[Word = muss ^ Tone = L+H*]

- Finde alle Segmente mit dem Label LAB auf der Ebene LEV1, die mit einem Label LAB2 auf der Ebene LEV2 verbunden ist
- für ^ müssen die Ebenen eine hierarchische Verbindung haben und der Ausdruck muss geklammert sein

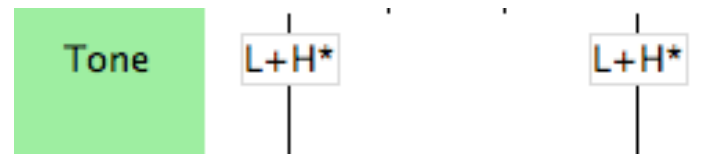
EQL – Syntax & Semantik

- Reihenfolgen (Sequenzen) abfragen durch Verknüpfen einfacher Abfragen:

[EA -> EA]

- [LEV1 = LAB1 -> LEV2 = LAB2]

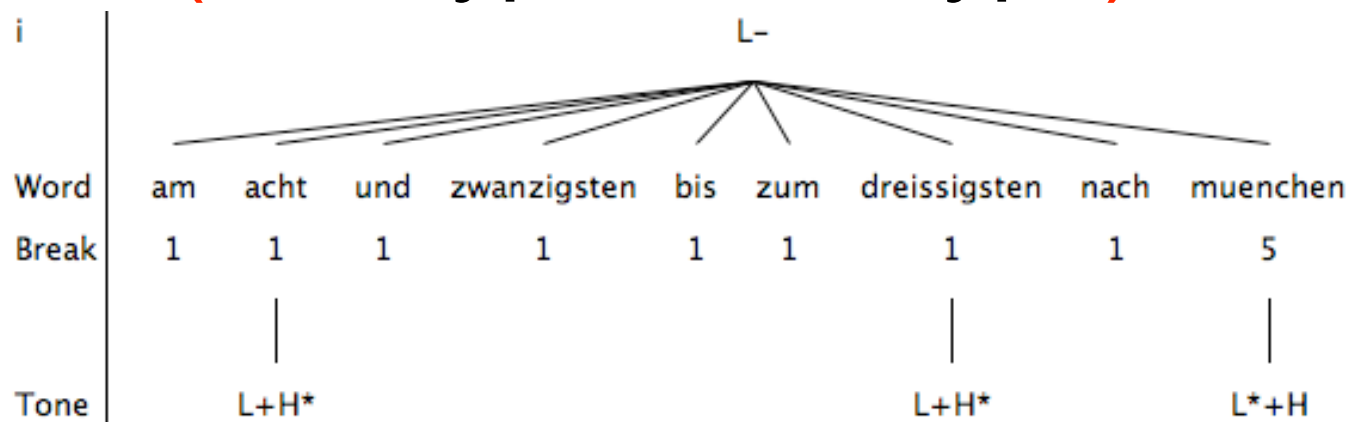
[Tone = L+H* -> Tone = L+H*]



- Finde die Reihenfolge der Labels LAB1 LAB2 auf der Ebene LEV

EQL – Syntax & Semantik

- Postionsabfragen: **Start()** **Medial()** **End()**
- **Start(LEVhyper, LEVhypo) = 1**



Start(i, Tone)=1 Medial(i, Tone)=1 End(i, Tone)=1

- Finde das jeweils erste/mittlerste/letzte Label der LEVhypo Ebene, das mit einem Label der LEVhyper Ebene verbunden sind.

EQL – Syntax & Semantik

- $\text{Start}(\text{LEVhyper}, \text{LEVhypo}) = 0$

$$\text{Start}(i, \text{Tone}) = 0$$

$$\text{Medial}(i, \text{Tone}) = 0$$

$$\text{End}(i, \text{Tone}) = 0$$

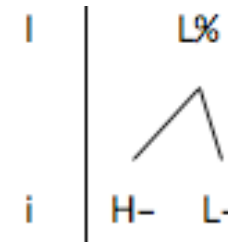
- Finde alle Labels auf der LEVhypo Ebene, die nicht das jeweils erste/mittlerste/letzte Label ist, was mit einem Label der LEVhyper Ebene verbunden sind.

EQL – Syntax & Semantik

- Anzahlabfragen $\text{Num}() = < > <= >= (\text{COM})$
- $\text{Num}(\text{LEVhyper}, \text{LEVhypo}) \text{COM Z AHL}$

$$\text{Num}(l,i) = 2$$

$$\text{Num}(l,i) < 3$$



- Finde alle Labels auf der LEVhyper Ebene, die COM Z AHL Verbindungen zur LEVhypo Ebene haben, also:
- Finde alle Intonationsphrasen, die mit 2 / weniger als 3 Intermediatephrasen verbunden sind

KIM

- Phrasen
- Akzente
- Konturen
- Deklination
- Registerwechsel
- Sprechgeschwindigkeitswechsel
- ProLab Zeicheninventar für Emu übersetzt

KIM-Beispiel

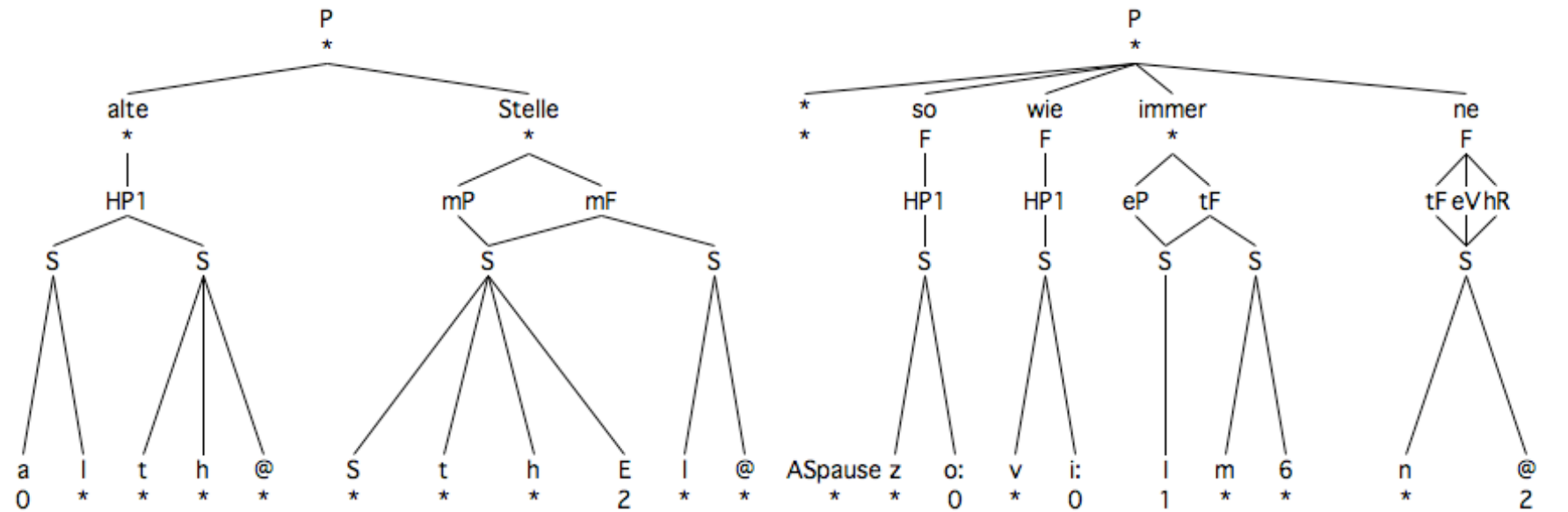
Phrase
SpeakRate

Word
Func

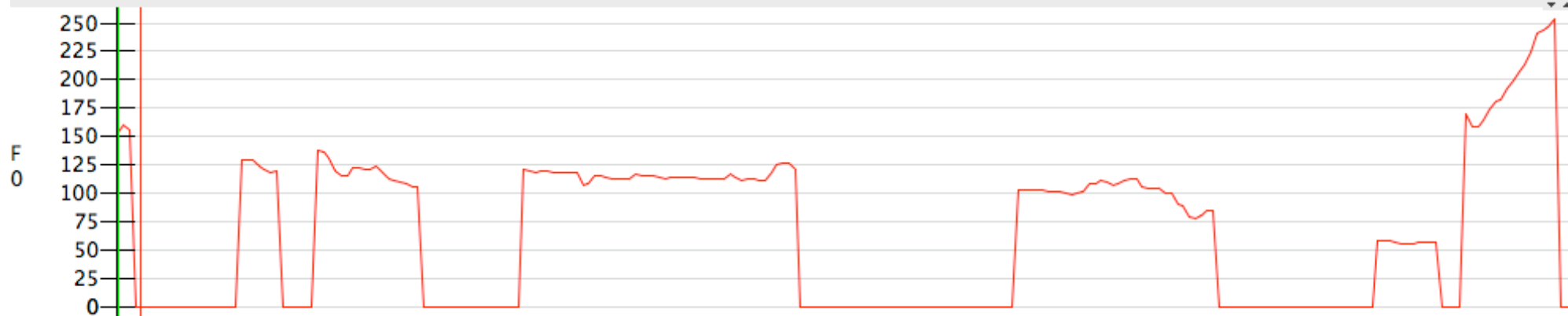
Contour

Syllable

Phonetic
Accent



Phonetic Accent	a	l	t	h	@	S	t	h	E	l	@	ASpause	z	o:	v	i:	l	m	6	n	@
	0								2				*	0	*	0	1	*	6	*	2



KIM-Bsp. alternativ

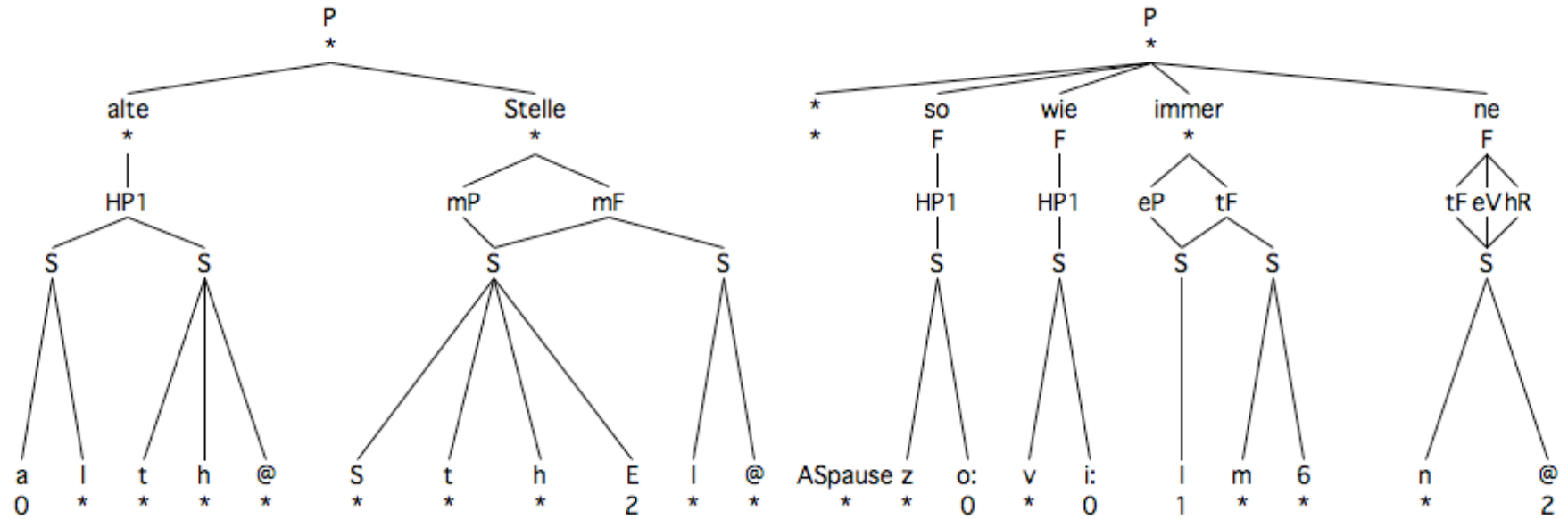
Phrase
SpeakRate

Word
Func

Contour

Syllable

Phonetic
Accent

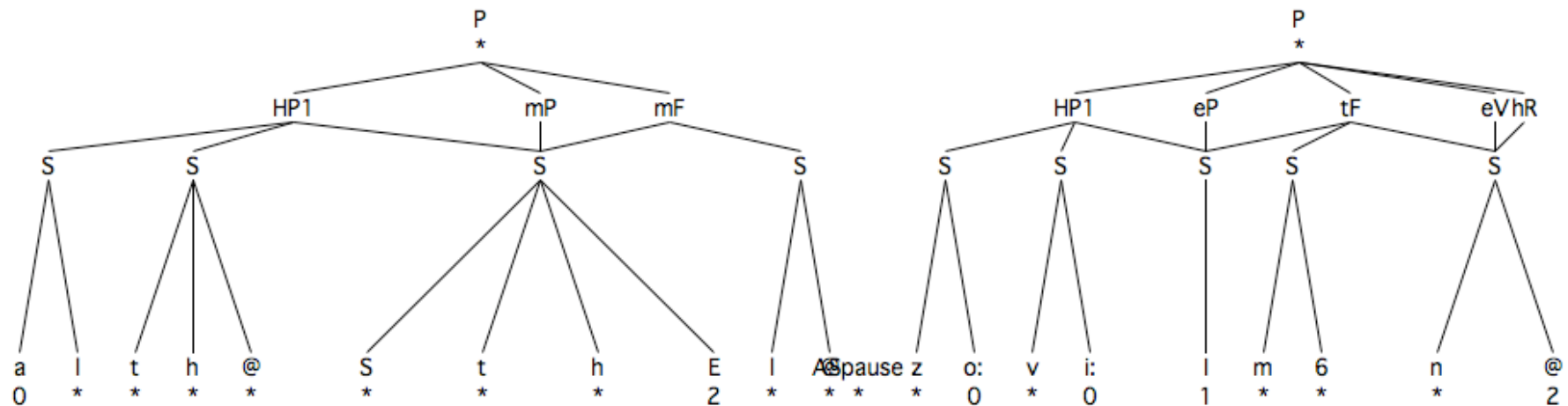


Phrase
SpeakRate

ContourC

Syllable

Phonetic
Accent



KIM-Phrasen

- einfache Abfrage:

Phrase != x5y7

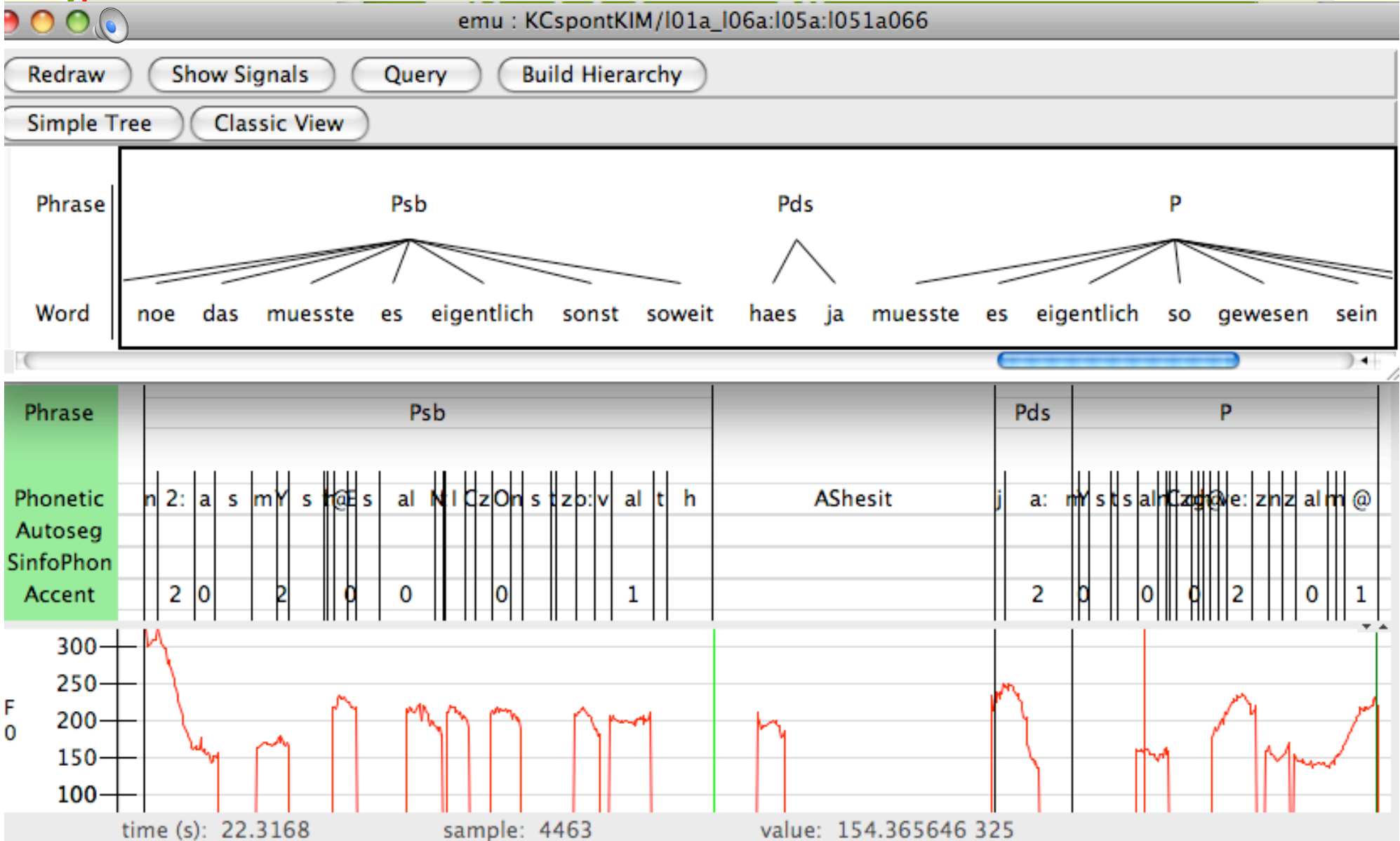
P Default Fall

Pds der erste Gipfel der nachfolgenden Phrase liegt tiefer als der letzte Gipfel der vorangehenden Phrase (d.h. Downstep über die Phrasengrenze hinweg)

Psb die Phrasengrenze fällt mit einem syntaktischen Abbruch zusammen

Ptb die Phrasengrenze wurde durch einen technischen Abbruch verursacht

Phrasen Beispiel



Finde Pds und Umgebung

der erste Gipfel der nachfolgenden Phrase liegt tiefer als der Pds letzte Gipfel der vorangehenden Phrase (d.h. Downstep über die Phrasengrenze hinweg)

```
Pdss = phrase[label(phrase) == "Pds",]  
Pds = Pdss[utt(Pdss) == "l01a_l06a:l05a:l051a066",]  
Pds.davor = emu.requery(Pds,"Phrase","Phrase", seq = -1)  
Pds.danach = emu.requery(Pds,"Phrase","Phrase", seq = 1)
```

```
Pds.f0 = emu.track(Pds,"F0")  
Pds.davor.f0 = emu.track(Pds.davor,"F0")  
Pds.danach.f0 = emu.track(Pds.danach,"F0")
```

```
Pds.davor.f0.max = max(Pds.davor.f0)  
Pds.f0.max = max(Pds.f0)  
Pds.danach.f0.max = max(Pds.danach.f0)
```

```
barplot(c(Pds.davor.f0.max,Pds.f0.max,Pds.danach.f0.max))
```

KIM/ProLab - Akzente

- einfache Abfrage:

`Accent != x5y7`

- 0 deakzentuiert
- 1 partiell deakzentuiert
- 2 default Akzentstärke
- 3 emphatisch verstärkte Akzentuierung

- Emphatische Akzente in default Phrasen

`[Accent = 3 ^ Phrase = P]`

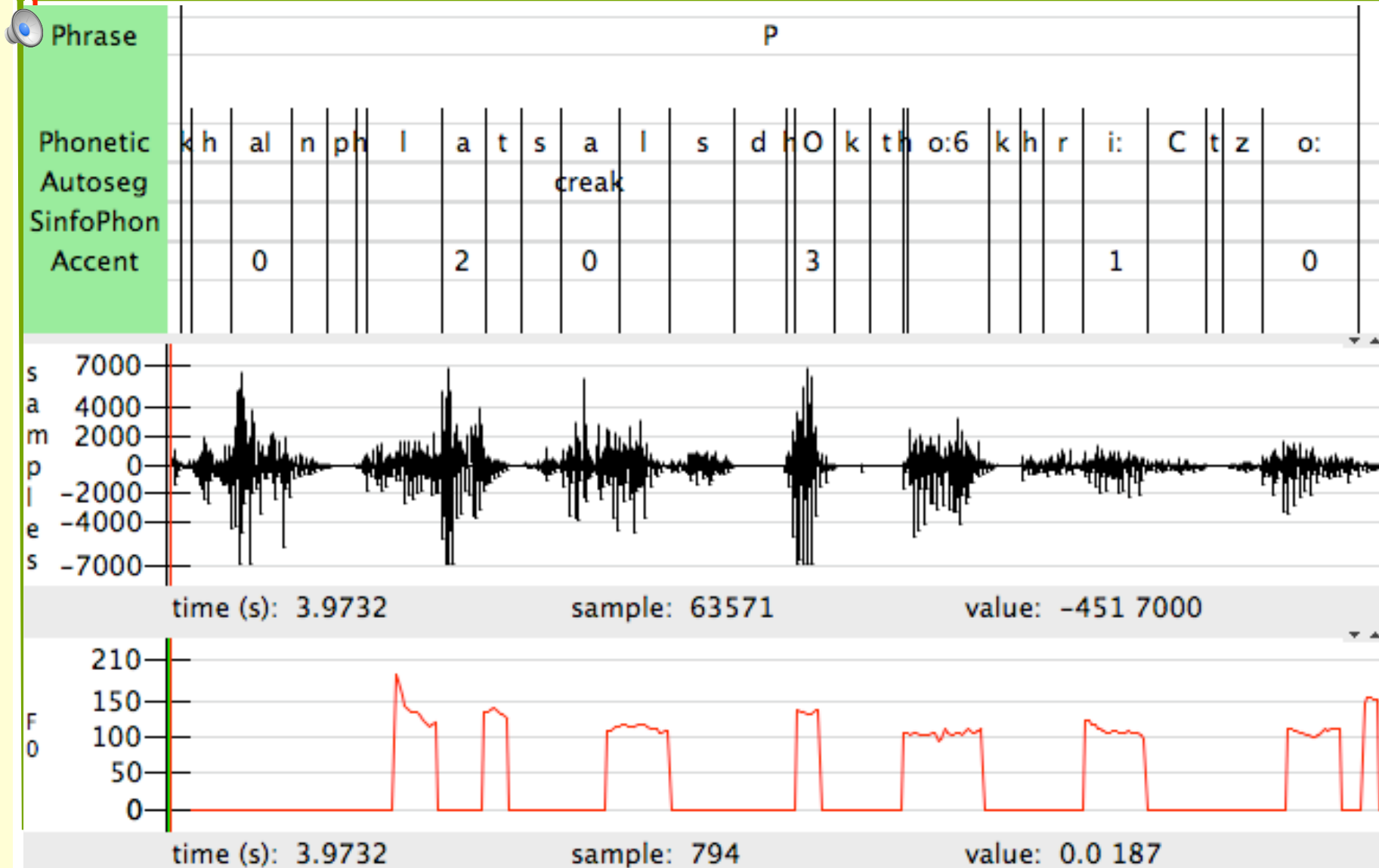
- Deakzentuierte Silben

`[Syllable != x5y7 ^ Accent = 0]`

- Phone, die mit default Akzent markiert sind

`Phonetic != x5x7 & Accent = 2`

Akzente Beispiel



KIM-Akzentkonturen

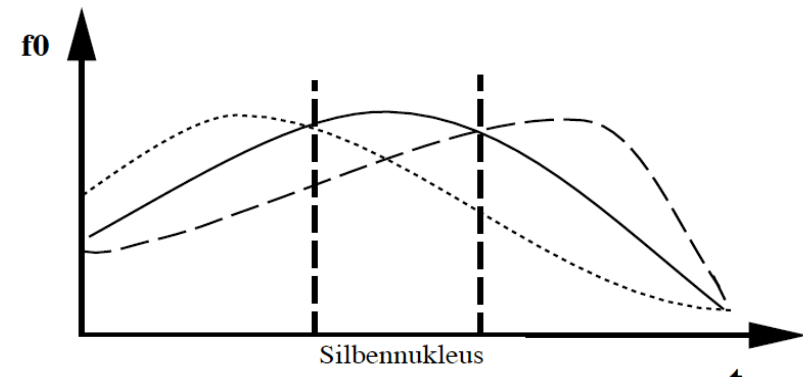
■ Abfrage:

[Contour != x5y7 ^ Accent = 1 | 2 | 3]

eP früher Gipfel

mP mittlerer Gipfel

IP später Gipfel

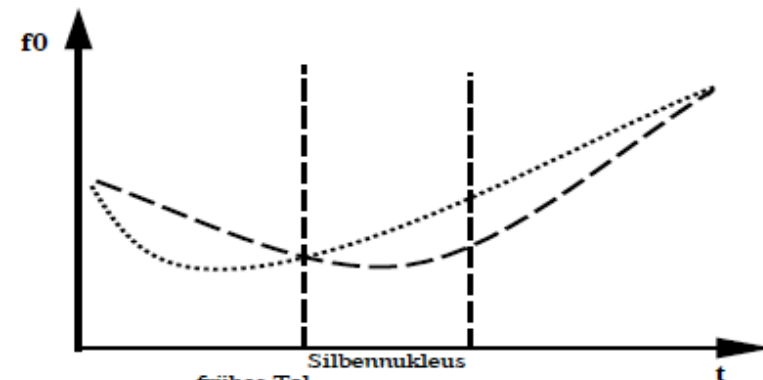


nP es bleibt eben vor, über und nach der Akzentsilbe

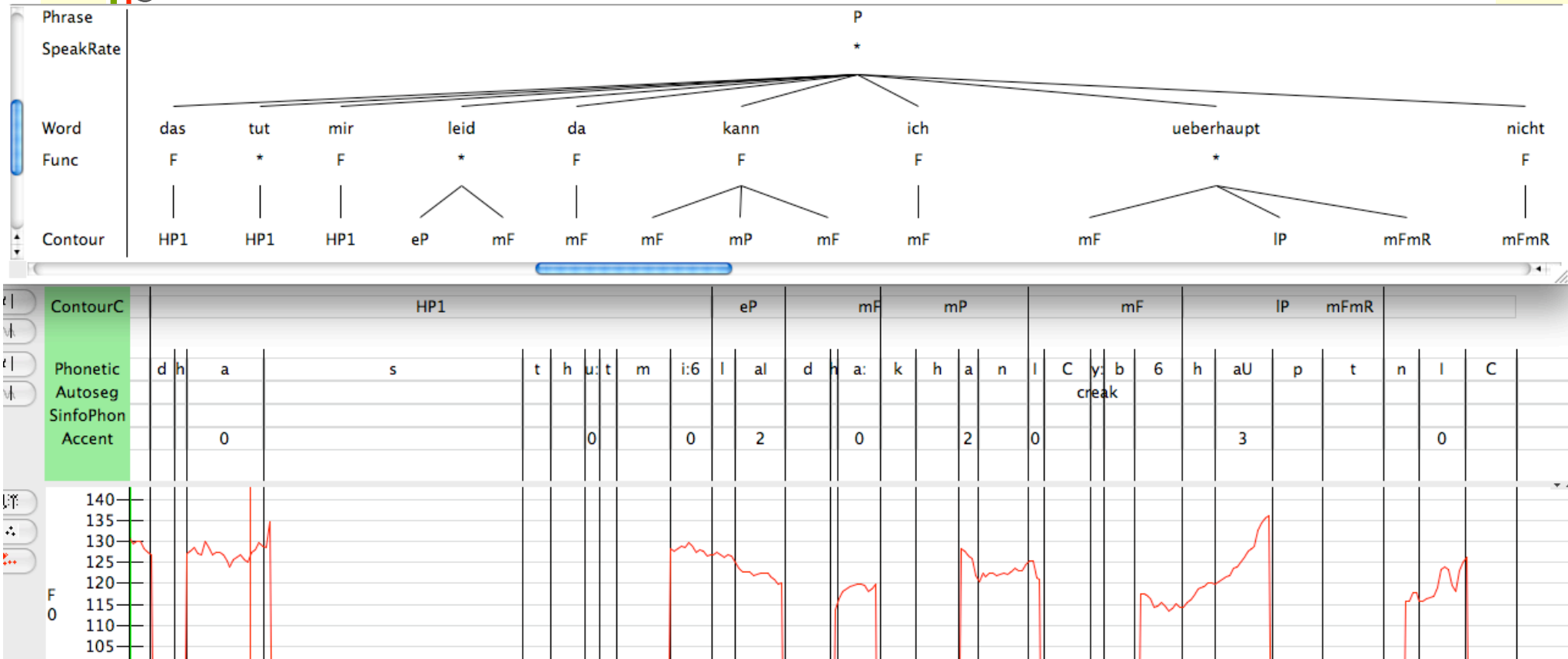
eV frühes Tal, Einsatz
liegt vor dem Akzentvokal

neV Nicht frühes Tal, Einsatz
liegt in dem Akzentvokal

U... Upstep des ...

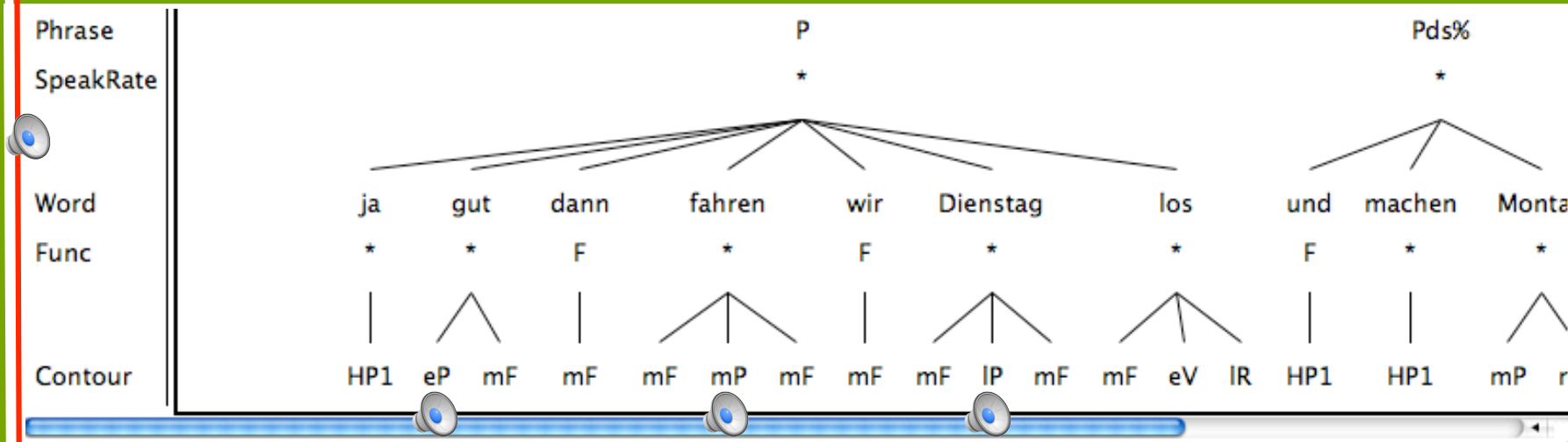


Akzentkonturen - Beispiel

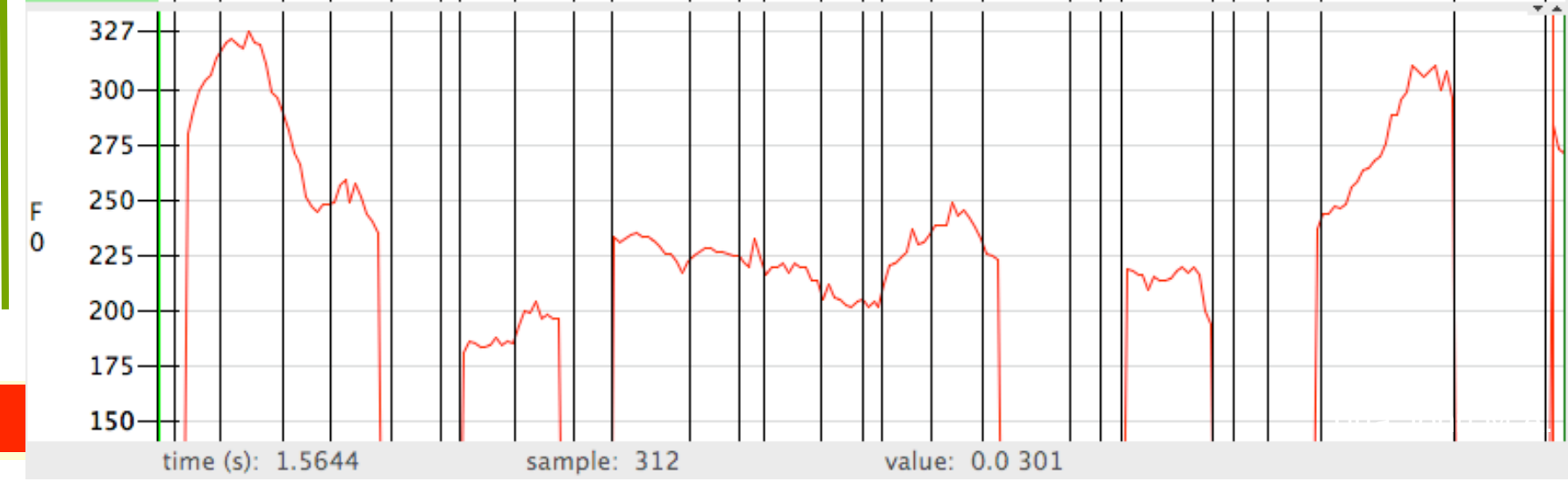


Zeitgebundene Labels der ContourC Ebene durch Emu-Module hier2sigview

Einzelbeispiel



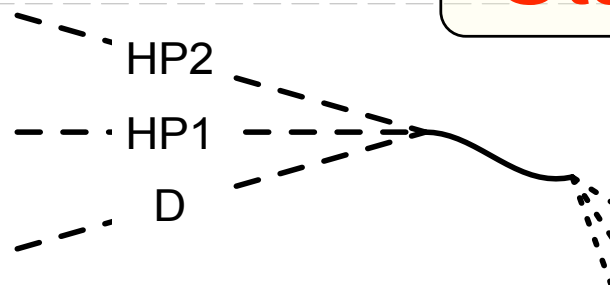
Phonetic	j	a:	g	u:	d	h	a	m	f	a:6	m	v	i:6	d	h	i:	n	s	t	h	a:	k	h	l	o:	s	
Autoseg																											
SinfoPhon																											
Accent		0		1						1																1	



KIM-Phraseninitiale K.

■ Abfrage:

Start(Phrase, Contour) = 1



HP1 der Anfang der Phrase liegt auf dem gleichen bzw. geringfügig höherem Niveau im Vergleich zum ersten Gipfel der Phrase

HP2 der Anfang der Phrase liegt auf dem bedeutsam höheren Niveau im Vergleich zum ersten Gipfel der Phrase und das F0 fällt häufig schnell zum Gipfel hin ab

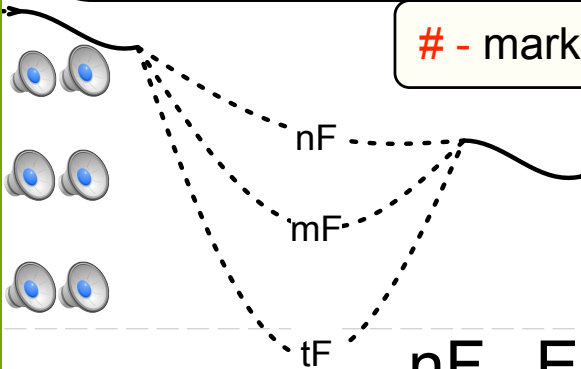
D für fehlende Verläufe im Kiel Corpus mit der Bedeutung Default

KIM-Konkatenations K.

- Abfrage:

[[Contour = eP | mP | IP | eV | neV | nP
-> **#Contour != 5x7y]** ->
Contour = eP | mP | IP | eV | neV | nP **]**

- markiert die Segmente, die das Ergebnis der Abfrage sein sollen



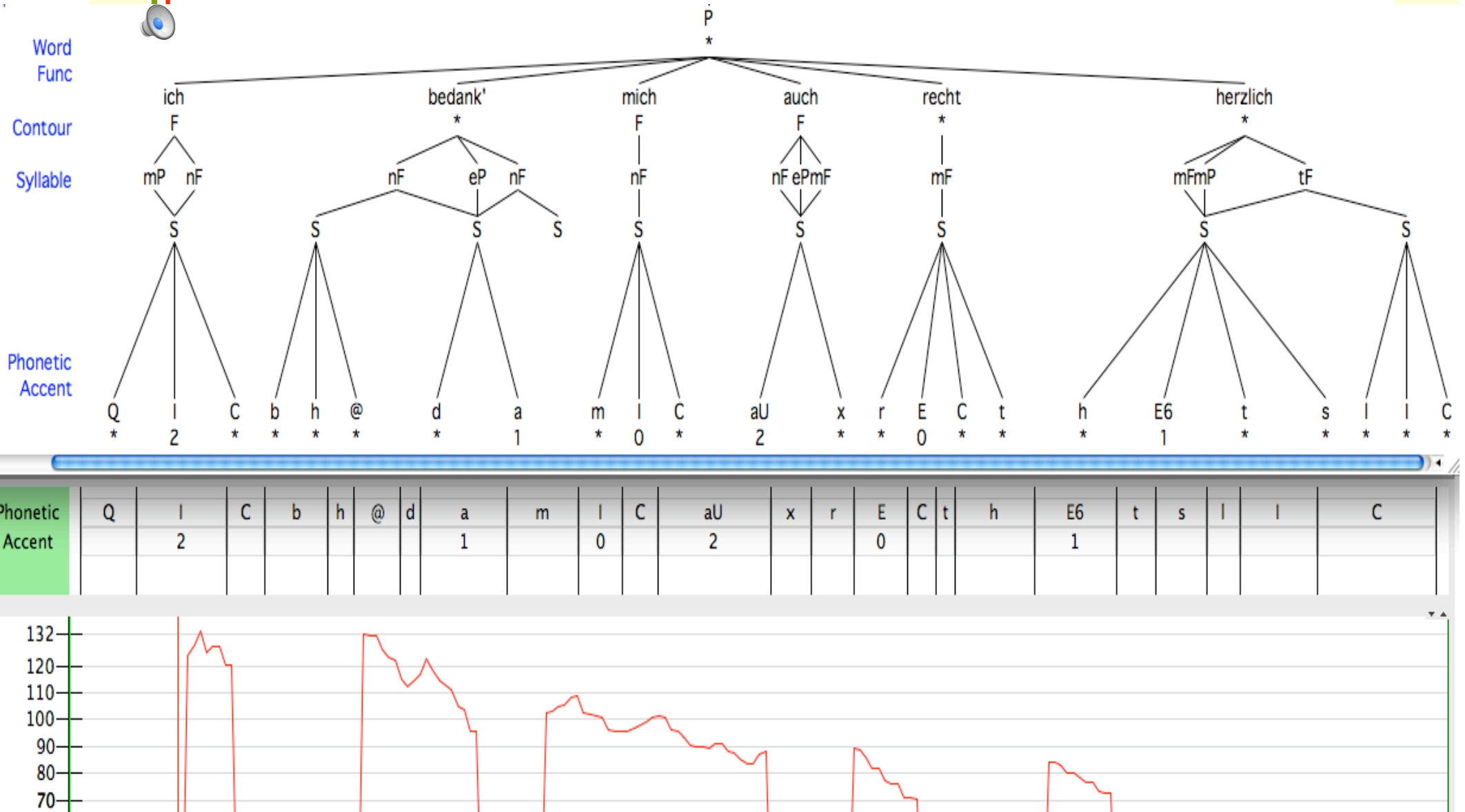
nF Ebener Verlauf bzw. kein Absinken

mF Leichtes Absinken

tF Absinken bis zur Base-Line

zw.
mP neV

Konkatenationsbeispiel



KIM/ProLab-Phrasenfinale K.

- Abfrage:

End(Phrase, Contour) = 1

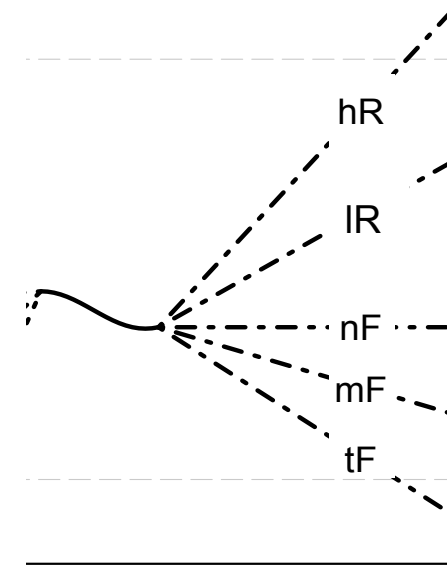
nF nach Gipfelkonturen bleibt es eben 

mF nach Gipfelkonturen fällt es leicht ab 

tF nach Gipfelkonturen fällt es bis zur Base-Line ab 

IR nach Talkonturen: Anstieg bis zum mittleren Bereich der Sprechstimme

hR nach Talkonturen: Anstieg bis zum obersten Bereich der Sprechstimme



Phrasenfinale K. Bsp.

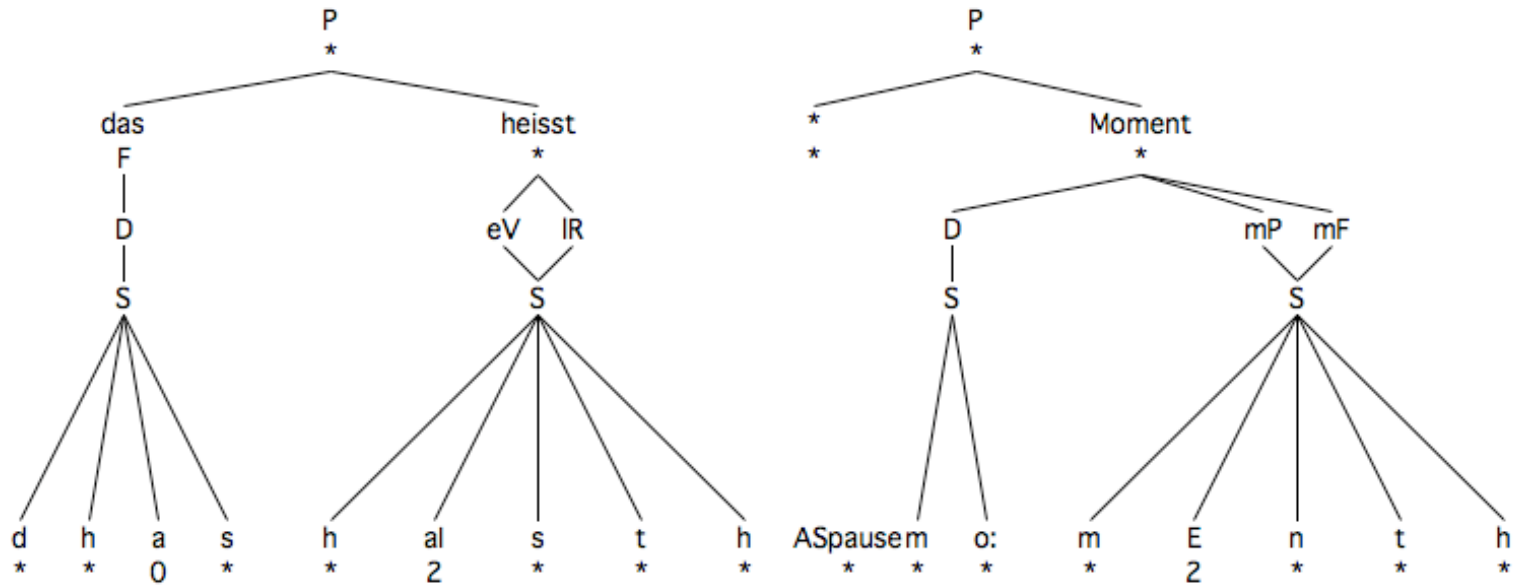
Phrase
SpeakRate

Word
Func

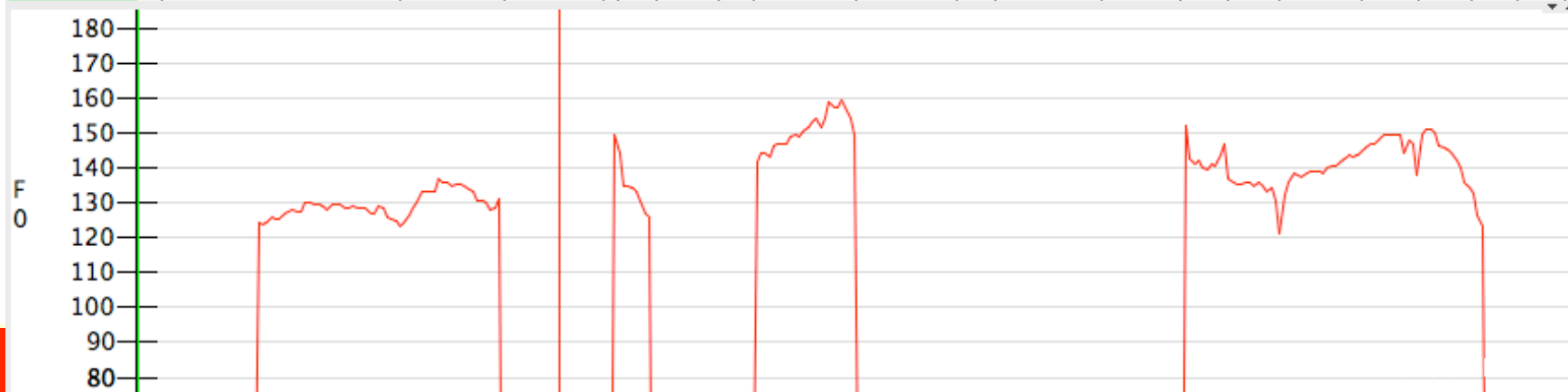
Contour

Syllable

Phonetic
Accent



Phonetic Accent	E:	m	d	h	a	s	h	al	s	t	h	ASpau	sem	o:	m	E	n	t	h
	2				0			2								2			



Beispiel (2)



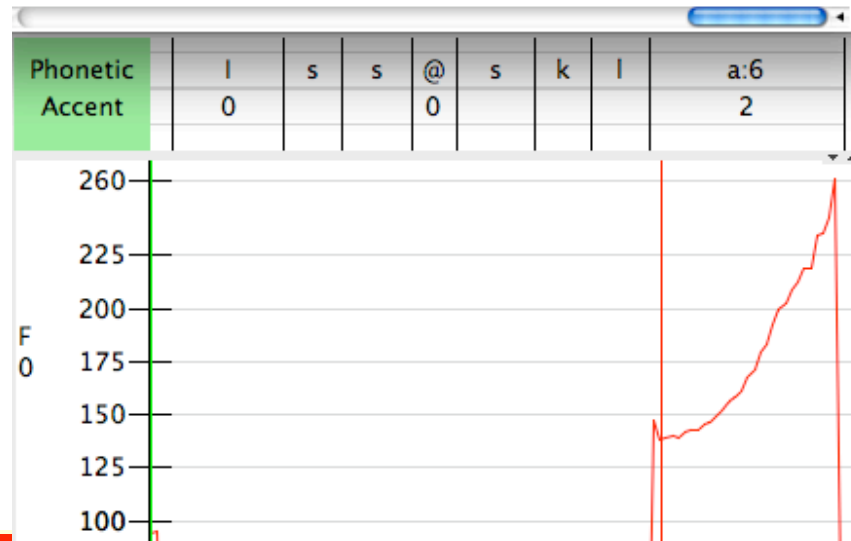
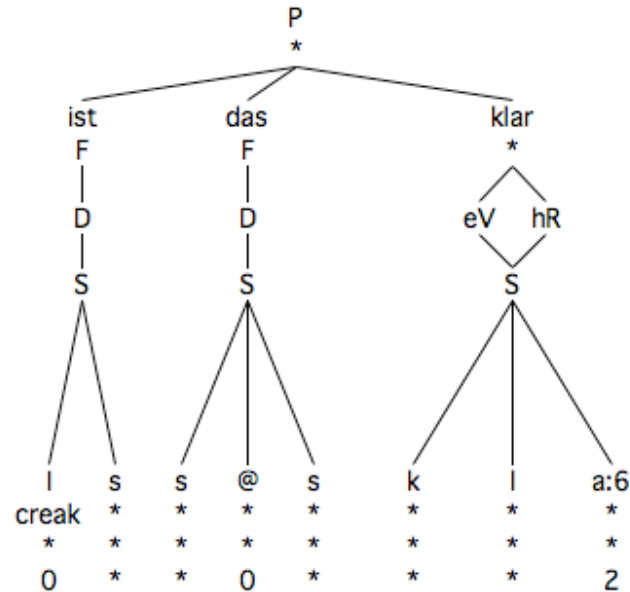
Phrase
SpeakRate

Word
Func

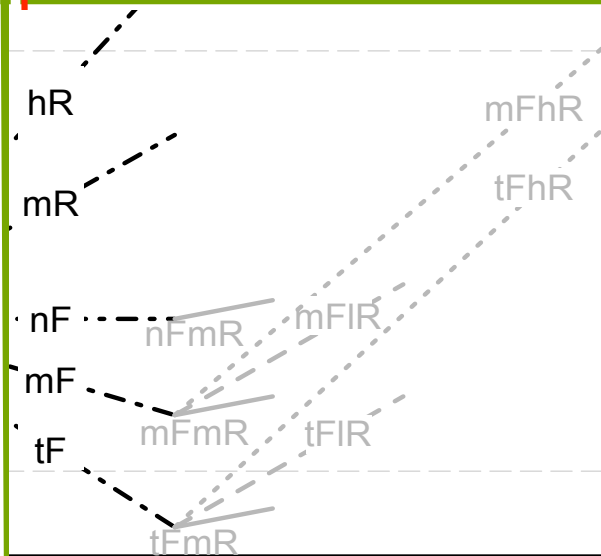
Contour

Syllable

Phonetic
Autoseg
SinfoPhon
Accent



KIM-Phrasenfinale K.



- nFmR pseudoterminal: nach ebenen leichter Anstieg
- mFmR pseudoterminal: nach einem nicht besonders tiefen Fallen leichter Anstieg
- tFmR pseudoterminal: nach einem tiefen Fallen leichter Anstieg
- mFtR fallend-steigend: nach einem nicht besonders tiefen Fallen ein deutlicher Anstieg, der nicht bis zur obersten Grenze der Sprechstimme geht
- tFtR fallend-steigend: nach einem tiefen Fallen ein deutlicher Anstieg, der nicht bis zur obersten Grenze der Sprechstimme geht
- mFhR fallend-steigend: nach einem nicht besonders tiefen Fallen ein deutlicher Anstieg, der bis zur obersten Grenze der Sprechstimme geht
- tFhR fallend-steigend: nach einem tiefen Fallen ein deutlicher Anstieg, der bis zur obersten Grenze der Sprechstimme geht

Phrasenfinale K. Bsp.

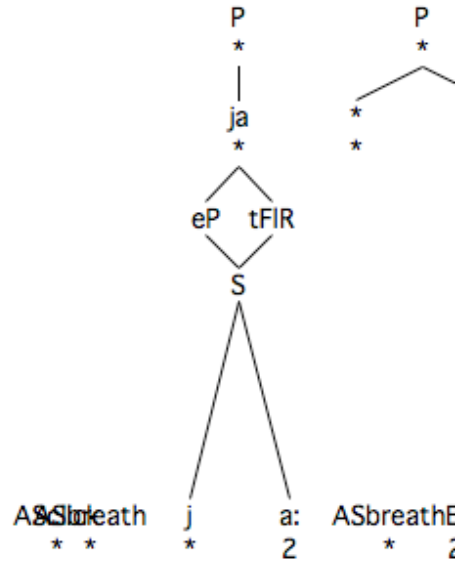
Phrase
SpeakRate

Word
Func

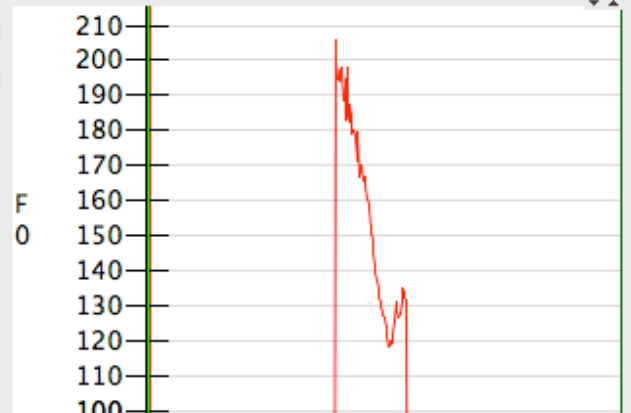
Contour

Syllable

Phonetic
Accent



Phonetic Accent	lickASbreath	j	a: 2	ASbreath



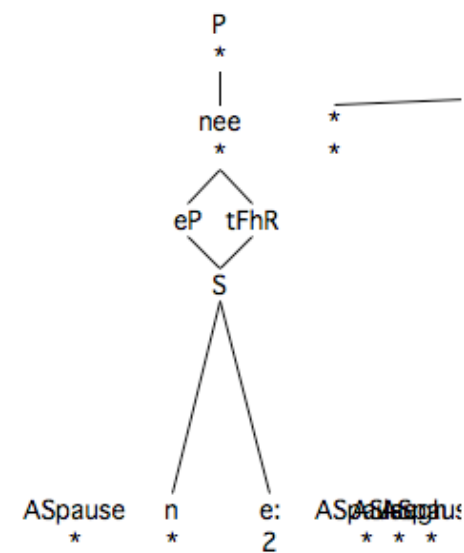
Phrase
SpeakRate

Word
Func

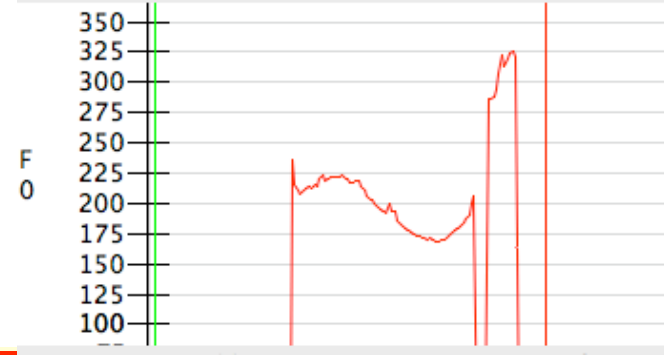
Contour

Syllable

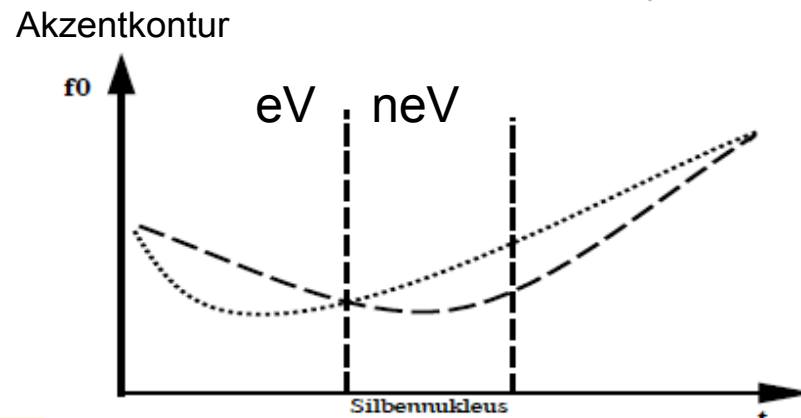
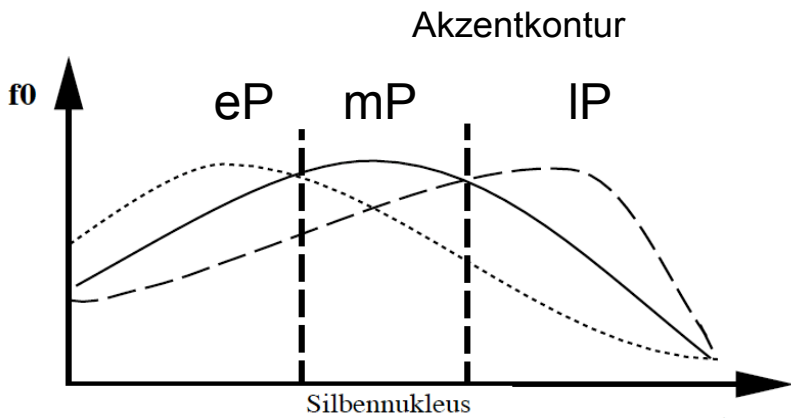
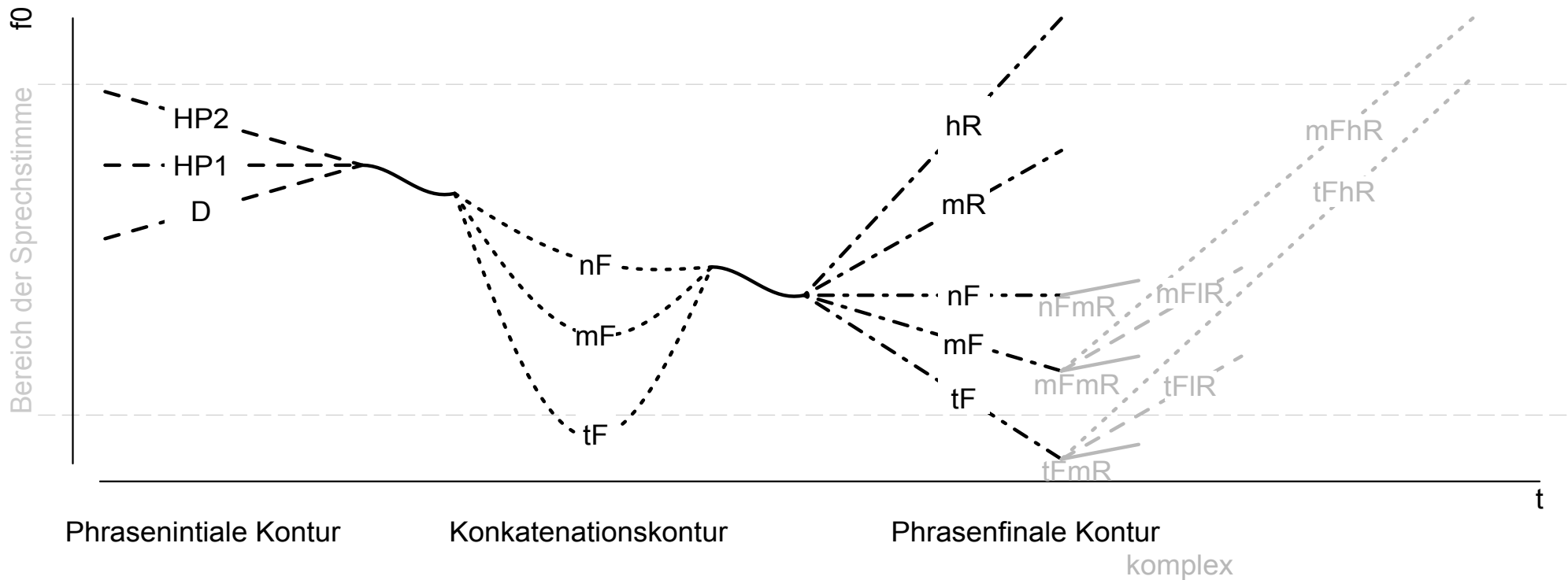
Phonetic
Accent



Phonetic Autoseg SinfoPhon Accent	n	e: 2	



KIMs Konturen



Finde Pds und Umgebung

der erste Gipfel der nachfolgenden Phrase liegt tiefer als der Pds letzte Gipfel der vorangehenden Phrase (d.h. Downstep über die Phrasengrenze hinweg)

```
Pdss = phrase[label(phrase) == "Pds",]  
Pds = Pdss[utt(Pdss) == "l01a_l06a:l05a:l051a066",]  
Pds.davor = emu.requery(Pds,"Phrase","Phrase", seq = -1)  
Pds.danach = emu.requery(Pds,"Phrase","Phrase", seq = 1)
```

```
Pds.f0 = emu.track(Pds,"F0")  
Pds.davor.f0 = emu.track(Pds.davor,"F0")  
Pds.danach.f0 = emu.track(Pds.danach,"F0")
```

```
Pds.davor.f0.max = max(Pds.davor.f0)  
Pds.f0.max = max(Pds.f0)  
Pds.danach.f0.max = max(Pds.danach.f0)
```

```
barplot(c(Pds.davor.f0.max,Pds.f0.max,Pds.danach.f0.max))
```