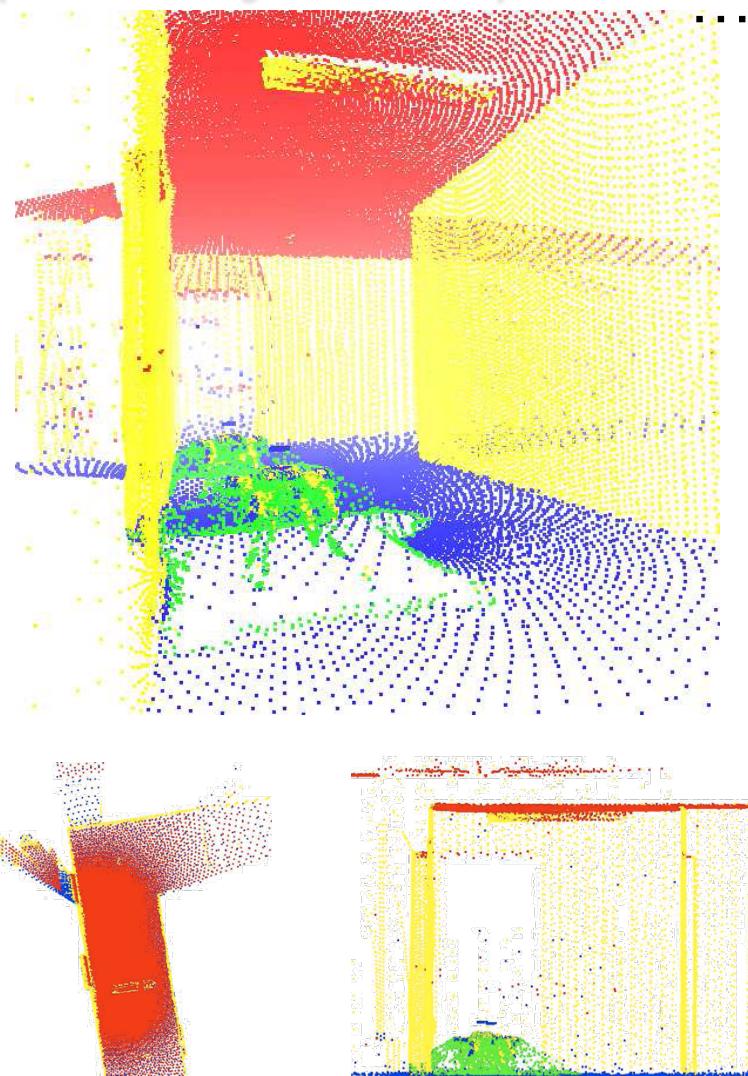


Einzelscan (vertikal zylindrisch)



Bsp. Ebenen-Erkennung

... Beispiel für reinen bottom-up Prozess!

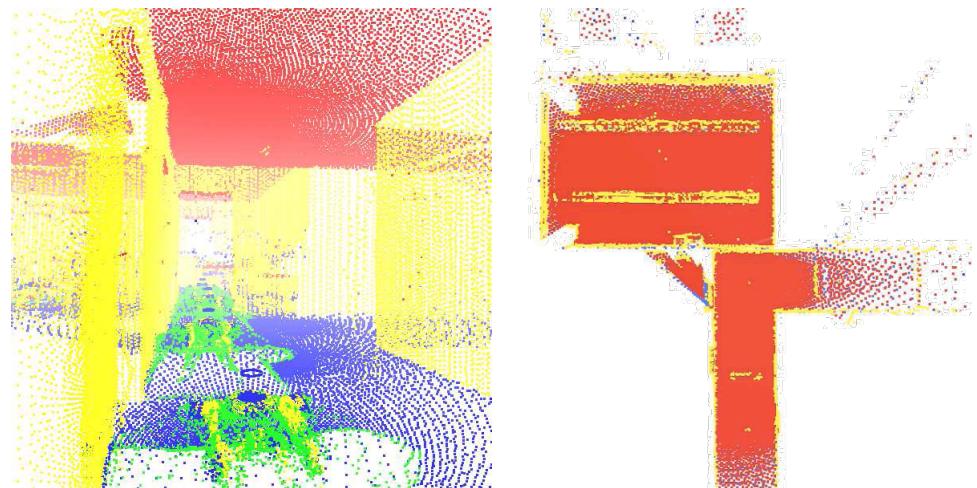
blau: Fußbodenpunkte: „Unterste“ Ebene

rot: Deckenpunkte: „Oberste“ Ebene

gelb: Wände: senkrechte Ebenen

grün: Artefakte (gescannte Roboterecken)

Fünf Scans, registriert

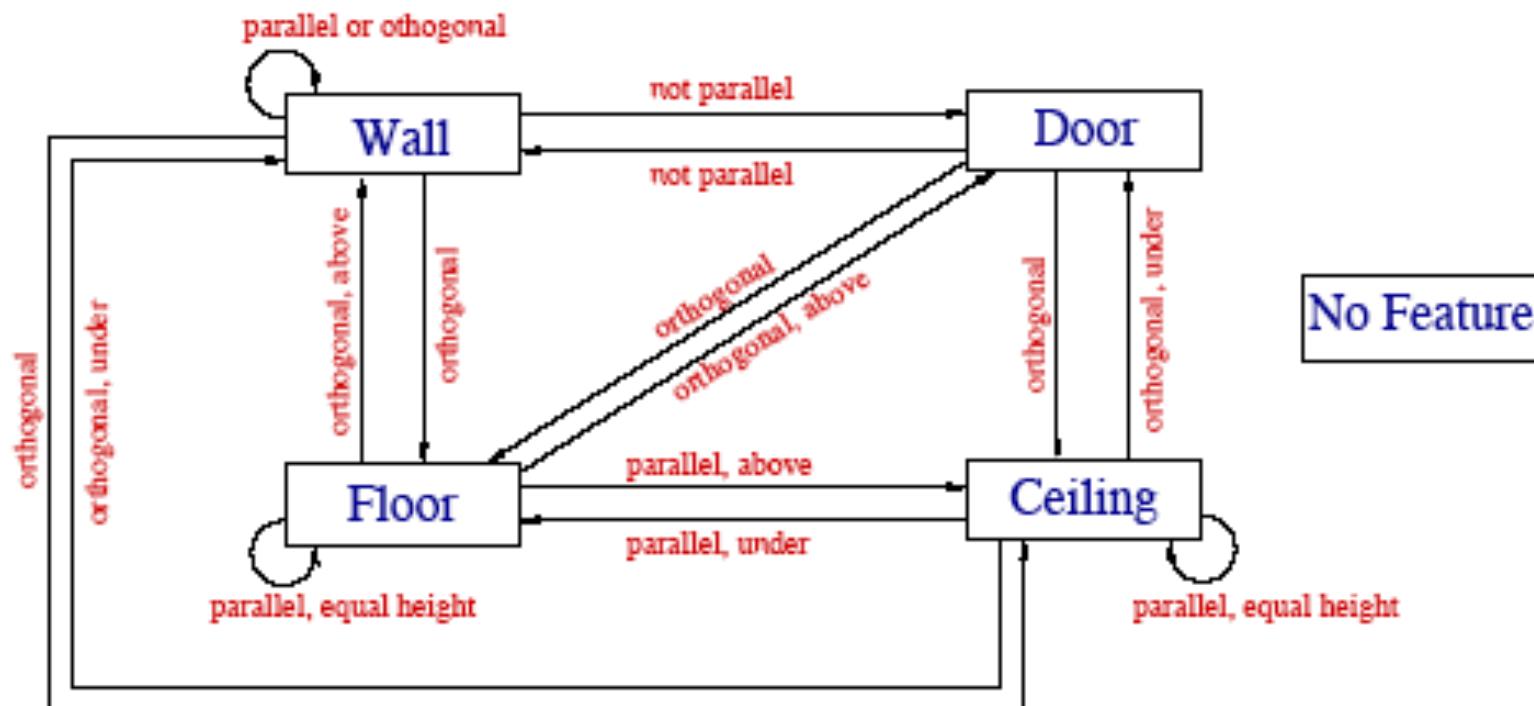


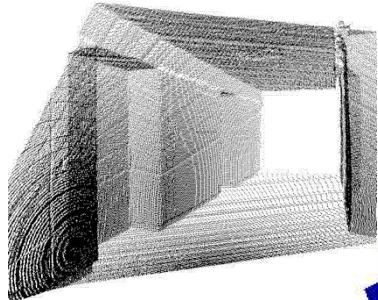
Beispiel: Gebäudeinnenflächen top-down

[Nüchter et al, 3DIM-2003]

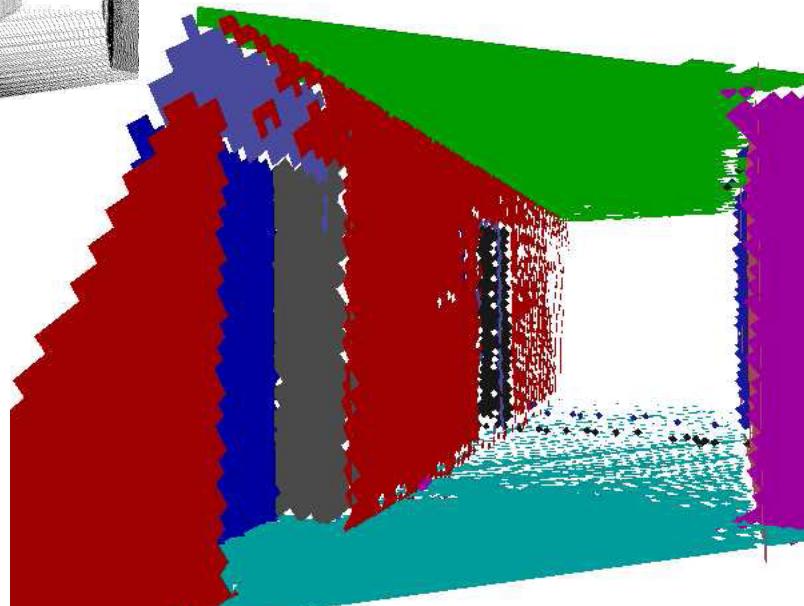
Finde Fußboden, Decken, Wände und Türen gemäß erkannten Flächen und „semantischen“ Relationen zwischen ihnen

Formulierung: Konsistente Zuordnung manuell definierter Raumrelationen zwischen benachbarten planaren Bereichen

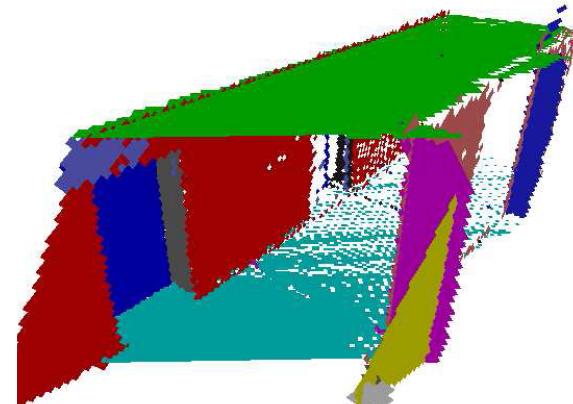




Bsp. Gebäudeflächen: Ergebnis

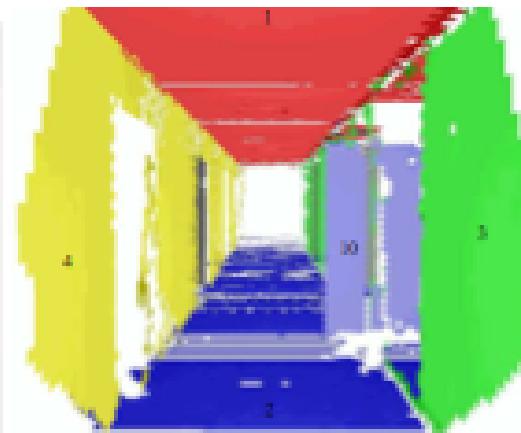


1	Wall
2	Ceiling
3	Wall
4	Door
5	Wall
6	Floor
7	Wall
8	Wall
9	Wall
10	Door
11	Wall
12	Wall



„Interpretierendes Löschen“

von Punkten, die nicht zum Gebäude gehören



Wissen hinter semantischen Karten

- ... kann in jedem beliebigen Format vorliegen
- ... liegt im Gebäudeflächenbeispiel als (Mini-)Constraintnetz vor
- ... löst das Bootstrap-Problem in der Regel so, dass eine Bereichstheorie vorgegeben wird
(hier kein Lernen – grundsätzlich aber möglich)
- ... wird in den wenigen existierenden Beispielen meist als Theorie in einer Beschreibungslogik formuliert:
 - Repräsentation von Raumrelationen
 - Teil-Ganzes-Beziehungen (hierarchisch)
 - Anzahl-Restriktionen („Ein Essplatz hat 1 Tisch und 4-6 Stühle“)
 - Typische Ereignisse mit ihren zeitlichen Bestandteilen

Beispiel (Neumann/Möller): Tischdecken

- Elem. Objektklassen: Teller, Tasse, Untertasse, Gabel, ...
- Aggregate: Gedeck, ...
- El. Ereignisse: bewegen (Objekt),
absetzen (Teller, Gabel, ...), ...
- Komplexe Ereignisse: Tisch_decken, ...
- Raumrelationen: nahe_bei,
auf, unter, ...
- Zeitrelationen: vor, nach,
aufeinanderfolgend, ...
(\rightarrow 6(+6)+1 Allen-Relationen)



TBox fürs Tischdecken: Elementarobjekt

name:	plate
parents:	:is-a scene-object
parts:	pl-body :is-a body with pl-body-preds pl-view :is-a view with pl-view-preds
constraints:	(constraints between pl-body-preds and pl-view-preds)



- Modelliere Elementarobjekte als Aggregate aus einem Körper (x -body-preds) und einer Ansicht (x -view-preds)
- Nur für **starre Elementarobjekte**(!) ist Ansicht definiert
- Die Constraints stellen sicher, dass das Bild zum Objekt passt (Teller muss rund sein, Durchmesser $\approx n$, Innenfläche weiß, außen farbiger Ring, 12 Striche, möglicherweise glänzend, ...)
→ **Objektverankerung!**

TBox fürs Tischdecken: Aktion

name: place-cover
parents: :is-a agent-activity
parts: pc-tt :is-a table-top
pc-tp1 :is-a transport with (tp-ob :is-a plate)
pc-tp2 :is-a transport with (tp-ob :is-a saucer)
pc-tp3 :is-a transport with (tp-ob :is-a cup)
pc-cv :is-a cover

time marks: pc-tb, pc-te :is-a timepoint **Beginn&Ende der Aktion**

constraints: pc-tp1.tp-ob = pc-cv.cv-pl
pc-tp2.tp-ob = pc-cv.cv-sc
pc-tp3.tp-ob = pc-cv.cv-cp

...

pc-tp3.tp-te \geq pc-tp2.tp-te **Tassentransport endet nicht vor Untertassentransport**
pc-tb \leq pc-tp3. tp-tb

...

Szeneninterpretation nach Neumann/Möller

Eine Szeneninterpretation ist eine konsistente Theorie (ABox+TBox) in einer Beschreibungslogik über der TBox der definierten Konzepte

Unvollständigkeit von Szeneninterpretationen

- Objekte müssen nicht Aggregaten zugeordnet sein (z.B.: Es gibt Teller, der nicht zu einem Gedeck gehört)
- Objekte müssen nicht auf spezifischste Weise zugeordnet sein (z.B.: Es gibt Besteckteil, das nicht als Messer, Gabel oder Löffel identifiziert ist)
- Nicht alle Bestandteile von Aggregaten müssen als Instanzen vorliegen (z.B.: bei einem Gedeck ist die zugehörige Tasse nicht instanziert)

Grundschritte der Szeneninterpretation

- **Aggregat-Instanzierung:** Gegeben Objekte, fasse sie zu passendem Aggregat-Objekt zusammen (eine Art von Abduktion!) **Beispiel:** Erkannte Objekte der Typen Teller, Messer, Gabel, Untertasse fasse zu Gedeck zusammen (ohne Tasse)
- **Instanz-Verfeinerung** mit zwei Formen:
 - **Spezialisierung:** Übergang zu Unterklasse in der Ontologie (**Beispiel:** Frühstücksteller statt Teller)
 - **Expansion:** Instanzierung weiterer Bestandteile einer Aggregat-Instanz (**Beispiel:** Tasse-Objekt zu Gedeck-Objekt zuordnen, das noch ohne Tasse-Instanz war)
- **Instanz-Mischung:** Vermeintliche „Kopien“/Aliase in der ABox desselben physischen Objekts zusammenführen
Beispiel: Kontur- und Texturmodul haben je 1 Messer an ähnlicher Position erkannt und instanziert – mach daraus 1 Objekt!

Perception = Hallucination + Control

Alle Szeneninterpretationsschritte
erfordern heuristische Entscheidungen!

- Wähle Datum (Bildregion, Elementarobjekt/e, Aggregat/e), das bearbeitet werden soll
- Wähle Art des Interpretationsschritts
- Wähle bevorzugte Art der Ausführung des Interpretationsschritts
(Beispiel: Wozu spezialisieren? Worein expandieren?)

1. Welche dieser Entscheidungen trifft man zu welcher Zeit?
(Erst alle Bilddaten *bottom-up* auswerten, oder zwischendurch schon Aggregate suchen?)
2. Was schafft Übersicht im Raum der möglichen Schritte?

Softwarearchitektur: Die „Tafel-Metapher“

- Mehrere „Experten“ arbeiten asynchron an der Lösung eines gemeinsamen Problems
- Daten dafür lesen sie von einer gemeinsamen „Tafel“
- Ihre Beiträge schreiben sie für alle lesbar an die Tafel; diese können, aber müssen nicht „Adressaten“ haben
- „**Blackboard-Architektur**“



- Blackboards mit oder ohne Speicher
- Heutige Implementierung: Tupelraum
- Adäquatheit für Modellierung von Kognition wird zuweilen behauptet
- Berühmtes, klassisches Beispiel:
Hearsay II
(kombinierte online-Sensordatenverarbeitung mit semantischer Interpretation!)

Erman & al.: The Hearsay II Speech-Understanding System.
Integrating Knowledge to Resolve Uncertainty.
ACM Comp. Surveys 12:213-253 (1980)

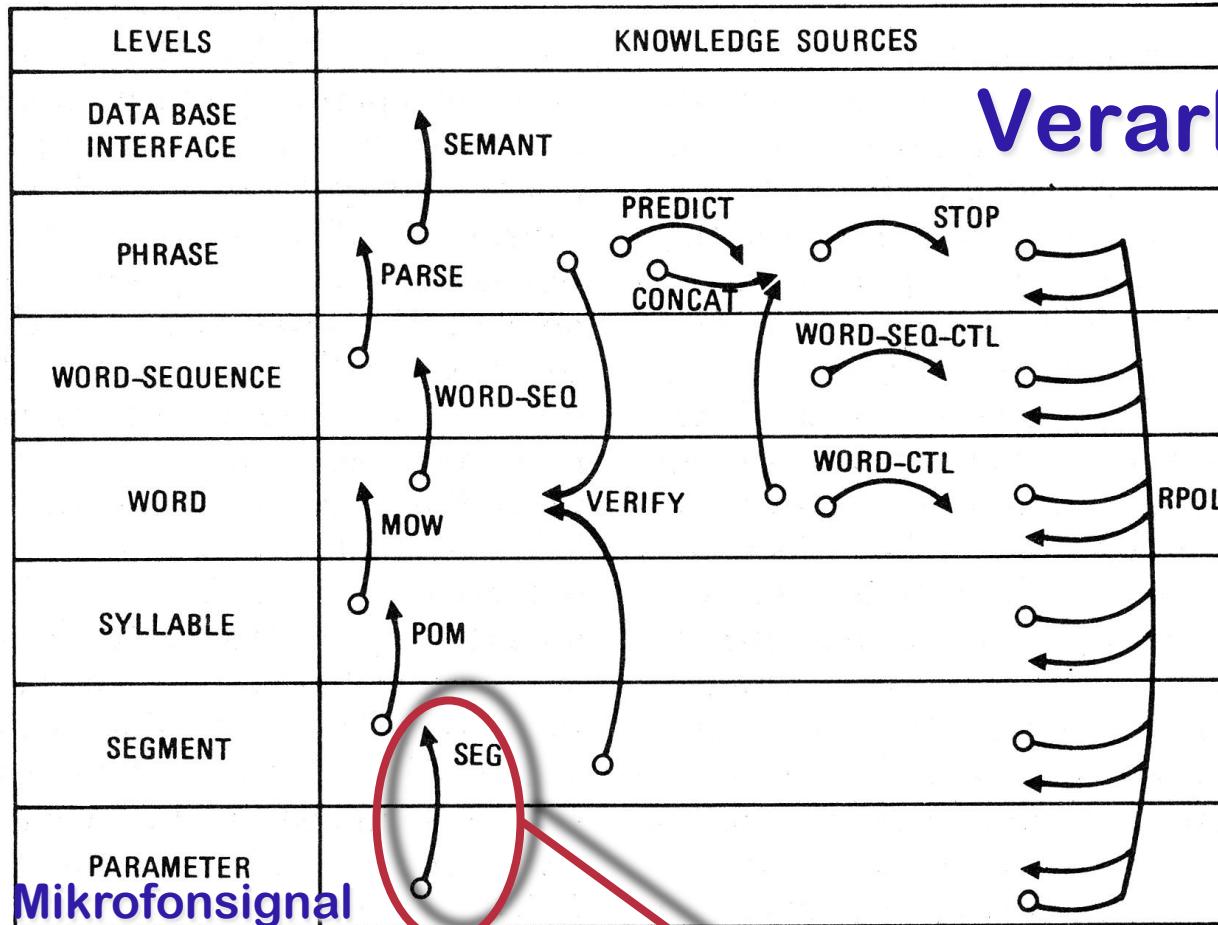
Exkurs: Hearsay II

- Erfolgreich entwickelt für einen (D)ARPA-Wettbewerb zum Verstehen natürlicher gesprochener Sprache 1971-6:
 - Vokabular 1000 Worte, „quasi-online“, <10% Fehlerrate
- Blackboard-Architektur zur Vermittlung zwischen Daten- und Wissensebenen der Sprachverarbeitung
- Stark eingeschränkter Diskursbereich (Schach, frühe KI-Literatur)

Hearsay II repräsentiert nicht den Stand der Technik
in Verarbeitung gesprochener Sprache!
Heute setzt man z.B. Hidden-Markov-Modelle ein

Wissensebenen der Sprachverarbeitung

- **Pragmatik**: Regeln im Sprachgebrauch (z.B. Sprecherwechsel)
- **Semantik**: Wort/Phrasenbedeutungen, inhaltliche Beziehungen zwischen Teilen einer Äußerung („Ich Tarzan, Du Jane!“)
- **Syntax**: Regeln der „korrekten“ oder „akzeptablen“ Kombination von Wortformen und Reihenfolgen (nicht nur was Deutschlehrer korrekt finden, ist praktisch okay!)
- **Morphologie**: Regeln, wie Wortsegmente korrekt/verständlich mit anderen kombiniert werden (z.B. an-, ab-, -tu-, -n, -t)
- **Phonologie**: Lautliche Eigenschaften von Wörtern, auch z.B. Koartikulation („Abschleifung“), Prosodie („Wort/Satzmelodie“)
- **Phonetik**: Physikalische und physiologische Grundlagen der Spracherzeugung (aus welchen „Grundgeräuschen“ entsteht Sprache, und wie werden sie gehört?)



Verarbeitungsebenen

- Funktionen verarbeiten Daten der jeweiligen Ebenen
- reagieren auf neuen Blackboard-Eintrag („Stimulus“) mit Aktion, die ggf zu neuem Blackboard-Eintrag führt

Symbol Grounding!

A priori gibt es keine Reihenfolge
der Verarbeitung der Daten auf diesen Ebenen!

Z.B. POM: Vom Segment/Phon zur Silbe

... oder besser: Zur Silbenklassenhypothese

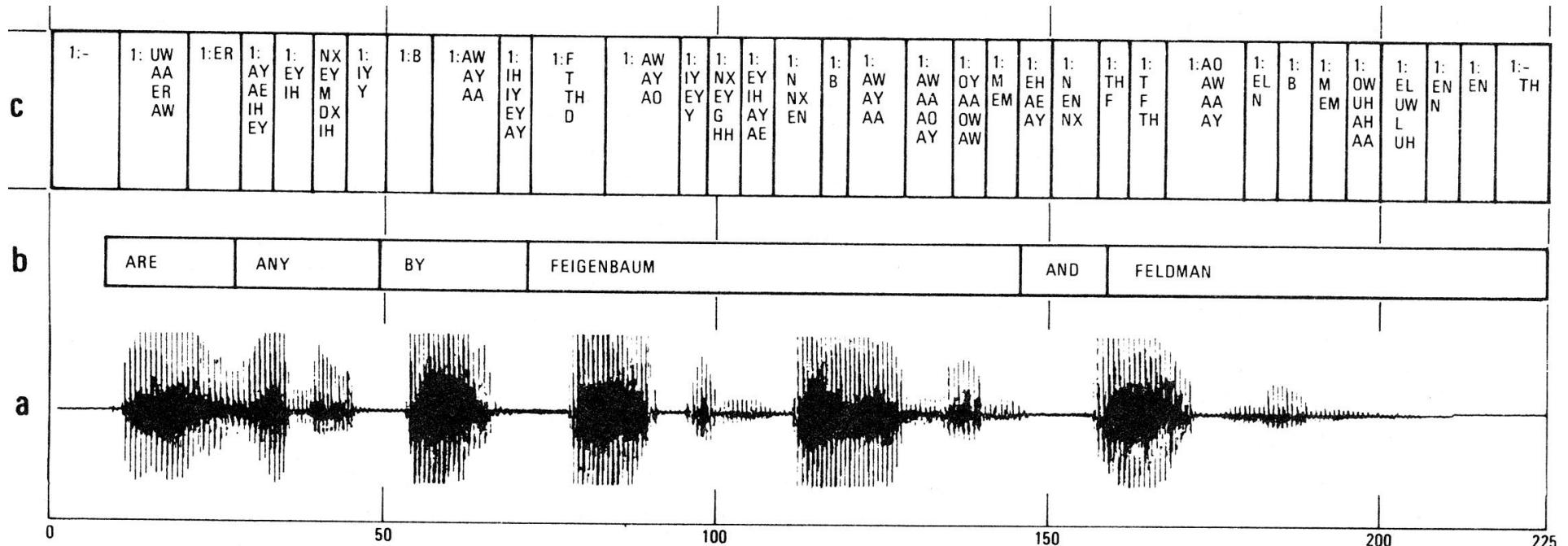
Code	Phon-Klasse	Phone in der Klasse
A	A-like	AE, AA, AH, AO, AX
I	I-like	IY, IH, EY, EH, IX, AY
U	U-like	OW, UH, U, UW, ER, AW, OY, EL, EM, EN
L	Liquid	Y, W, R, L
N	Nasal	M, N, NX
P	Stop	P, T, K, B, D, G, DX
F	Fricative	HH, F, TH, S, SH, V, DH, Z, ZH, CH, JH, WH

- Gegeben Phone (bzw. Phon-Hypothesen) in passender Zeitfolge,
- Finde Silbenklassenhypothesen, die daraus erzeugbar sind!

Beispiel: F – A – N – P – L – P – L

→ HAM – B(u)R – G(e)R

Hearsay-Verarbeitung („unterste“ Ebenen)



c: Phon-Hypothesen

b: als Referenz: wirklich gesprochener Text (*genuine truth*)

a: Mikrofon-Signal