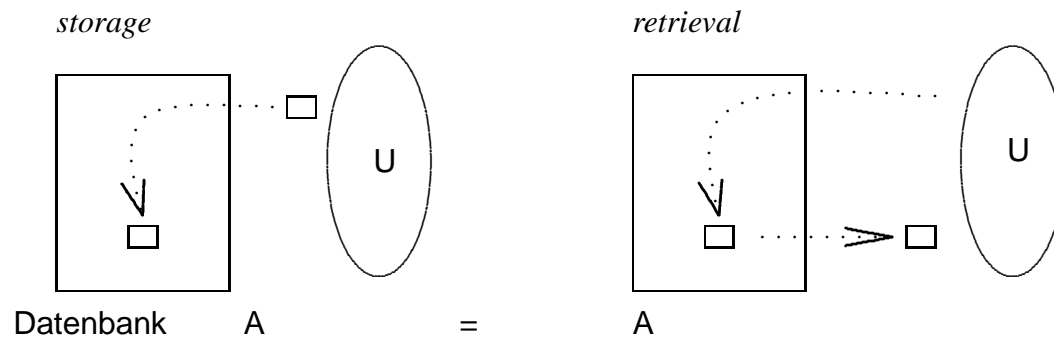


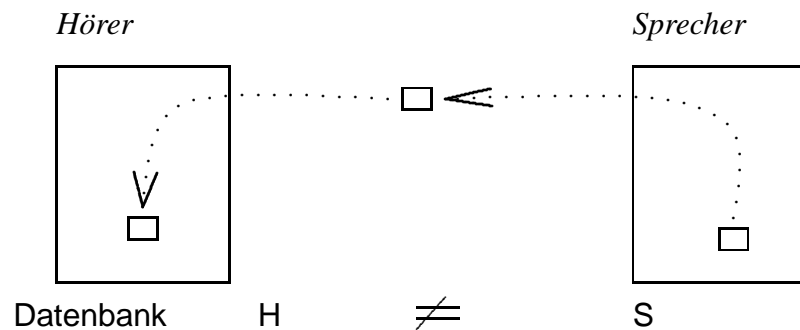
22. Datenbank Semantik

22.1 Datenbank-Metapher natürlicher Kommunikation

22.1.1 Interaktion mit einer konventionellen Datenbank



22.1.2 Natürliche Kommunikation



22.1.3 DB-Interaktion und NL-Kommunikation

- BETEILIGTE EINHEITEN

DB-Interaktion:

Die Interaktion findet zwischen verschiedenen Entitäten statt, nämlich einem Benutzer und einer Datenbank.

NL-Kommunikation:

Die Interaktion findet zwischen gleichartigen und gleichberechtigten Entitäten statt, Sprecher und Hörer.

- URSPRUNG DER STEUERUNG

DB-Interaktion:

Die Steuerung der Datenbankoperationen bei der Ein- und Ausgabe liegt beim Benutzer.

NL-Kommunikation:

Es gibt keinen Benutzer. Die Agenten steuern ihren Informationsfluß gegenseitig, indem sie abwechselnd den Sprecher- und den Hörermodus übernehmen (*turn taking*).

- FORM DER STEUERUNG

DB-Interaktion:

Der Benutzer steuert die Datenbankoperationen mit einer künstlichen Anfragesprache (SQL), deren Befehle in elektronische Prozeduren umgesetzt werden.

NL-Kommunikation:

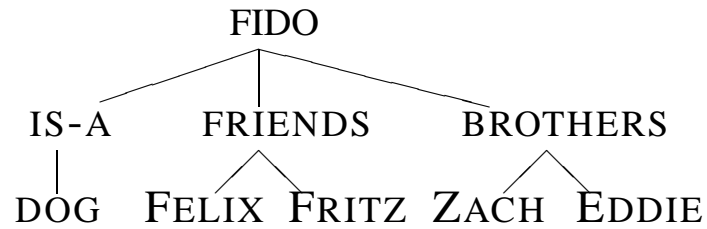
Der Sprecher steuert die Produktion als autonomer kognitiver Agent, wobei die Parameter der Äußerungssituation in das sprachliche Zeichen kodiert werden. Die Interpretation beim Hörer wird über Eigenschaften (der kognitiven Repräsentation) des Sprachzeichens gesteuert.

22.1.4 Erfolgreiche Kommunikation zwischen Sprecher und Hörer

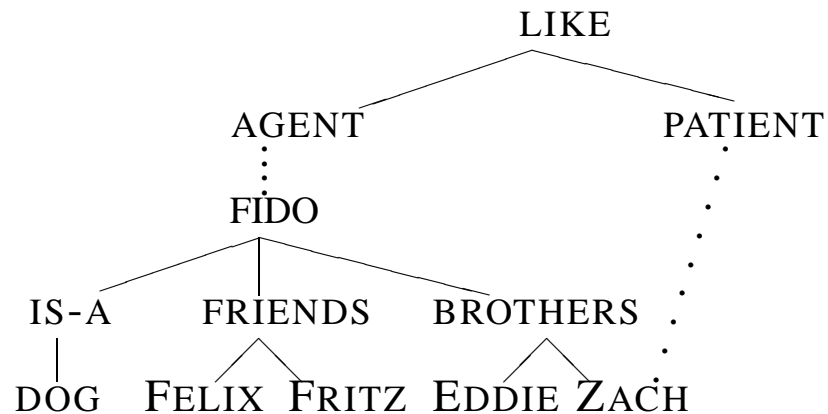
Der Datenbankinhalt, den der Sprecher in Sprache abgebildet, wird vom Hörer analog rekonstruiert.

- **Aussage:** der Hörer speichert den vom Sprecher formulierten Datenbankinhalt an einer analogen Stelle.
- **Frage:** der Hörer lokalisiert, welche Information vom Sprecher erfragt wird.
- **Befehl:** der Hörer lokalisiert, welche Handlung vom Sprecher erbeten wird.

22.1.5 Skizze eines einfachen Teilkontexts



22.1.6 Einbetten des Inhalts von *Fido likes Zach* in 21.1.5



22.2 Deskriptive Aporie und Qual der Wahl

22.2.1 Modelltheoretische Definition des Teilkontexts 22.1.5

\mathcal{MS} sei die Modellstruktur (A, I, J, \leq, F) , wobei

$$A =_{def} \{a_0, a_1, a_2, a_3, a_4\}$$

$$I =_{def} \{i_1\}$$

$$J =_{def} \{j_1\}$$

$$F(\text{fido}')(i_1, j_1) = a_0$$

$$F(\text{felix}')(i_1, j_1) = a_1$$

$$F(\text{fritz}')(i_1, j_1) = a_2$$

$$F(\text{zach}')(i_1, j_1) = a_3$$

$$F(\text{eddie}')(i_1, j_1) = a_4$$

$$F(\text{dog}')(i_1, j_1) = \{a_0\}$$

$$F(\text{fido-friends}')(i_1, j_1) = \{a_1, a_2\}$$

$$F(\text{fido-brothers}')(i_1, j_1) = \{a_3, a_4\}$$

22.2.2 Eine Erweiterung des Hörerkontexts um den Inhalt eines neuen Satzes wie z. B. Fido likes Zach

Erfordert das automatische Hinzufügen von 'F(like)(i₁, j₁) = {(a₀, a₃)}' zu 22.2.1

22.2.3 Kreieren eines *frame*

```
(make-frame
  fido
  (is-a (value dog))
  (friends (value felix fritz))
  (brothers (value zach eddie))
)
```

22.2.4 Frametheoretische Definition des Teilkontexts 22.1.5

```
(fido
  (is-a (value dog))
  (friends (value felix fritz))
  (brothers (value zach eddie))
)
```

22.2.5 Abruf von Information

```
(get-values 'FIDO 'FRIENDS)
(FELIX FRITZ)
```

22.2.6 Erweiterung des Kontexts auf Fido likes Zach

Erfordert Ableitung von

```
(fido  
  (like (value Zach)  
)
```

und automatische Einfügen von

```
(like (value Zach)
```

als neues Slot von 22.2.4.

22.3 Propositionen als Mengen koindizierter Proplets

22.3.1 Proposition 3.4.2 als Propletmenge (vorläufiges Format)

$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: Feld} \\ \text{Rolle: Argument} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x1 \\ \text{Funktorkonzept: enthalten} \\ \text{prn: 23} \\ \text{id: 7} \end{array} \right] \end{array} \right]$	$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: enthalten} \\ \text{Rolle: Funktor} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x2 \\ \text{Argument 1: Feld} \\ \text{Argument 2: Dreieck} \\ \text{prn: 23} \\ \text{cnj: 23 und 24} \end{array} \right] \end{array} \right]$	$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: Dreieck} \\ \text{Rolle: Argument} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x3 \\ \text{Funktorkonzept: enthalten} \\ \text{prn: 23} \\ \text{id: 8} \end{array} \right] \end{array} \right]$
---	---	--

$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: Feld} \\ \text{Rolle: Argument} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x4 \\ \text{Funktorkonzept: enthalten} \\ \text{prn: 24} \\ \text{id: 7} \end{array} \right] \end{array} \right]$	$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: enthalten} \\ \text{Rolle: Funktor} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x5 \\ \text{Argument 1: Feld} \\ \text{Argument 2: Quadrat} \\ \text{prn: 24} \\ \text{cnj: 23 und 24} \end{array} \right] \end{array} \right]$	$\left[\begin{array}{l} \textit{Type:} \\ \left[\begin{array}{l} \text{M-Konzept: Quadrat} \\ \text{Rolle: Argument} \end{array} \right] \\ \textit{Token:} \\ \left[\begin{array}{l} \text{I-Konzept}_{loc}: x6 \\ \text{Funktorkonzept: enthalten} \\ \text{prn: 24} \\ \text{id: 9} \end{array} \right] \end{array} \right]$
---	---	--

22.4 Proplets in einer klassischen Datenbank

22.4.1 Typen von Datenbanken

klassisch: Verbundbasiert

nichtklassisch: basiert auf dem Prinzip von Slot und Filler

22.4.2 Typen klassischer Datenbanken

Relationale, hierarchische und Netzwerk Datenbanken

22.4.3 Relationen zwischen Propletmerkmalen

Type \leftrightarrow Token

Token \leftrightarrow prn

prn \leftrightarrow cnj

Token \leftrightarrow id

Argument \leftrightarrow Funktor

Modifikator \leftrightarrow Modifikandum

22.4.4 Propositionen 3.4.2 als Wordbank (entgültiges Format)

TYPES

VEREINFACHTE PROPLETS

[M-Konzept: Dreieck] Rolle: Argument]	[I-Konzept _{loc} : x3 Funktorkonzept: enthalten prn: 23 id: 8]	
[M-Konzept: enthalten] Rolle: Funktor]	[I-Konzept _{loc} : x2 Argument 1:Feld Argument 2:Dreieck prn: 23 cnj: 23 und 24]	[I-Konzept _{loc} : x5 Argument 1:Feld Argument 2:Quadrat prn: 24 cnj: 23 und 24]
[M-Konzept: Feld] Rolle: Argument]	[I-Konzept _{loc} : x1 Funktorkonzept: enthalten prn: 23 id: 7]	[I-Konzept _{loc} : x4 Funktorkonzept: enthalten prn: 24 id:7]
[M-Konzept: Quadrat] Rolle: Argument]	[I-Konzept _{loc} : x6 Funktorkonzept: enthalten prn: 24 id: 9]	

22.4.5 Beispiel einer Netzwerkdatenbank

<i>owner records</i>	<i>member records</i>			
Informatik	Riedle	Schmidt	Stoll	...
Mathematik	Müller	Barth	Jacobs	...
Physik	Weber	Meier	Miele	...

22.4.6 Fortsetzungstypen

intrapositional:

vom Argument zum Funktor, vom Funktor zum Argument, vom Modifikator zum Modifikandum und umgekehrt.

extrapositional:

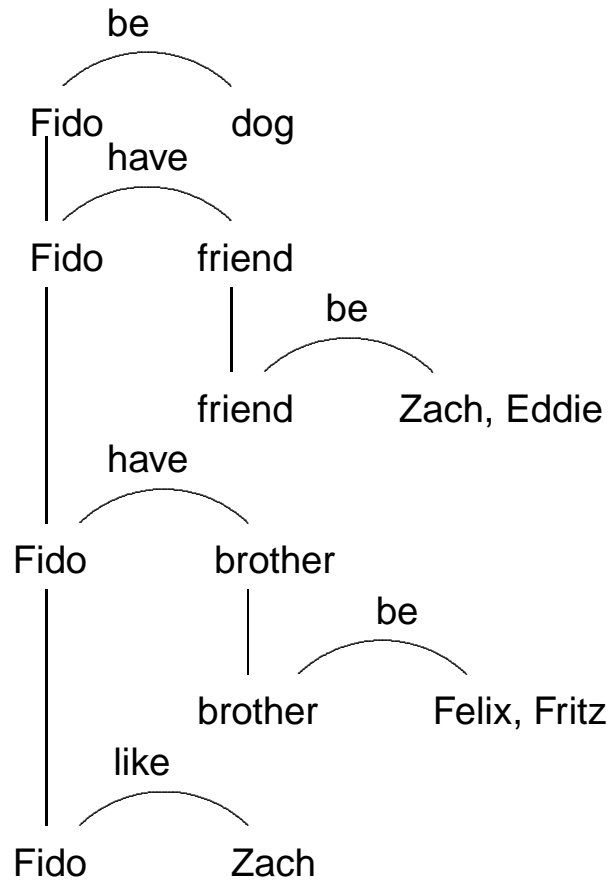
cnj von Verb zu Verb, id von Nomen zu Nomen

22.5 Beispiel einer Wortbank

22.5.1 Verkettete Propositionen in einer Wortbank

1. Fido is a dog.
2. Fido has friends.
3. The friends are Zach and Eddie.
4. Fido has brothers.
5. The brothers are Felix and Fritz.
6. Fido likes Zach.

22.5.2 Graphische Darstellung der Propositionen in 22.5.1



22.5.3 Subkontext 22.1.5 als eine Wordbank

TYPES

PROPLETS

[M-concept: be]
role: functor]

[I-concept_{loc}: x1]
arg1: Fido
arg2: dog
prn: 1
cnj: 1 and 2]

[I-concept_{loc}: x2]
arg1: friend
arg2: Zach, Eddie
prn: 3
cnj: 2 and 3
3 and 4]

[I-concept_{loc}: x3]
arg1: brother
arg2: Felix, Fritz
prn: 5
cnj: 4 and 5
5 and 6]

[M-concept: brother]
role: argument]

[I-concept_{loc}: x4]
functor: have
prn: 4
id:]

[I-concept_{loc}: x5]
functor: be
prn: 5
id:]

[M-concept: dog]
role: argument]

[I-concept_{loc}: x6]
functor: be
prn: 4
id:]

[M-concept: Eddie]
role: argument]

[I-concept_{loc}: x7]
functor: be
prn: 3
id: 3]

[M-concept: Felix role: argument]	[I-concept _{loc} : x8 functor: be prn: 5 id: 4]				
[M-concept: Fritz role: argument]	[I-concept _{loc} : x9 functor: be prn: 5 id: 5]				
[M-concept: Fido role: argument]	[I-con. _{loc} : x10 functor: be prn: 1 id: 1]	[I-con. _{loc} : x11 functor: have prn: 2 id: 1]	[I-con. _{loc} : x12 functor: have prn: 4 id: 1]	[I-con. _{loc} : x13 functor: like prn: 6 id: 1]	&
[M-concept: friend role: argument]	[I-concept _{loc} : x14 functor: have prn: 2 id:]	[I-concept _{loc} : x15 functor: be prn: 3 id:]			
[M-concept: have role: functor]	[I-concept _{loc} : x16 arg1: Fido arg2: friend prn: 2 cnj: 1 and 2 2 and 3]	[I-concept _{loc} : x17 arg1: Fido arg2: brother prn: 4 cnj: 3 and 4 4 and 5]			

[M-concept: like] role: functor	I-concept _{loc} : x18 arg1: Fido arg2: Zach prn: 6 cnj: 5 and 6	&	[M-concept: Zach] role: argument	I-concept _{loc} : x19 functor: be prn: 3 id: 2	I-concept _{loc} : x20 functor: like prn: 6 id: 2	&
------------------------------------	--	---	-------------------------------------	--	--	---

22.5.4 Semantische Repräsentation von Proposition 6

TYPES

PROPLETS

[M-concept: Fido]
[role: argument]

[I-concept_{loc}: x13]
functor: like
prn: 6
id: ?]

[M-concept: like]
[role: functor]

[I-concept_{loc}: x18]
arg1: Fido
arg2: Zach
prn: 6
cnj: ?]

[M-concept: Zach]
[role: argument]

[I-concept_{loc}: x20]
functor: like
prn: 6
id: ?]