

# Semantische Sprachverarbeitung

Carola Eschenbach

Universität Hamburg, FB Informatik

AB Wissens- und Sprachverarbeitung (WSV)

Sommersemester 2004

# Semantische Sprachverarbeitung

## Sitzung 3

- Produktivität von Bedeutung
- Techniken der Bedeutungserfassung
  - Logik : Semantik

## Formale Darstellung von Bedeutungen

### Prädikatenlogik

- Formeln repräsentieren Wahrheitsbedingungen
- Prädikate, Funktions- und Relationssymbole repräsentieren Eigenschaften, Abbildungen und Relationen (die Inhalte)
- Konstanten repräsentieren Objekte
- Logische Symbole (Junktoren, Quantoren, Variablen) bieten Inventar für die Verknüpfung der Inhalte

### Lambda-Kalkül

- Lambda-Abstraktion erlaubt die Bildung von komplexen Prädikaten, Funktions- und Relationsausdrücken.
- Mächtiges Werkzeug für die Darstellung der Kombinationsregeln

## Diskussion

## Systematische Variationen (1)

NP-Variation	VP-Konjunktion
[ <sub>NP</sub> Peter] [ <sub>VP</sub> lacht].	[ <sub>NP</sub> Peter] [ <sub>VP</sub> lacht und winkt].
[ <sub>NP</sub> Ein Junge] [ <sub>VP</sub> lacht].	[ <sub>NP</sub> Ein Junge] [ <sub>VP</sub> lacht und winkt].
[ <sub>NP</sub> Jeder Junge] [ <sub>VP</sub> lacht].	[ <sub>NP</sub> Jeder Junge] [ <sub>VP</sub> lacht und winkt].
[ <sub>NP</sub> Kein Junge] [ <sub>VP</sub> lacht].	[ <sub>NP</sub> Kein Junge] [ <sub>VP</sub> lacht und winkt].
<b>Satz-Konjunktion</b>	
[ <sub>NP</sub> Peter] [ <sub>VP</sub> lacht] und [ <sub>NP</sub> Peter] [ <sub>VP</sub> winkt].	
[ <sub>NP</sub> Ein Junge] [ <sub>VP</sub> lacht] und [ <sub>NP</sub> ein Junge] [ <sub>VP</sub> winkt].	
[ <sub>NP</sub> Jeder Junge] [ <sub>VP</sub> lacht] und [ <sub>NP</sub> jeder Junge] [ <sub>VP</sub> winkt].	
[ <sub>NP</sub> Kein Junge] [ <sub>VP</sub> lacht] und [ <sub>NP</sub> kein Junge] [ <sub>VP</sub> winkt].	

## Systematische Variationen (2)

NP-Variation	VP-Konjunktion
[ <sub>NP</sub> Maus] [ <sub>VP</sub> meselt].	[ <sub>NP</sub> Maus] [ <sub>VP</sub> meselt und diest].
[ <sub>NP</sub> Ein Kosi] [ <sub>VP</sub> meselt].	[ <sub>NP</sub> Ein Kosi] [ <sub>VP</sub> meselt und diest].
[ <sub>NP</sub> Jeder Kosi] [ <sub>VP</sub> meselt].	[ <sub>NP</sub> Jeder Kosi] [ <sub>VP</sub> meselt und diest].
[ <sub>NP</sub> Kein Kosi] [ <sub>VP</sub> meselt].	[ <sub>NP</sub> Kein Kosi] [ <sub>VP</sub> meselt und diest].
Satz-Konjunktion	
[ <sub>NP</sub> Maus] [ <sub>VP</sub> meselt] und [ <sub>NP</sub> Maus] [ <sub>VP</sub> diest].	
[ <sub>NP</sub> Ein Kosi] [ <sub>VP</sub> meselt] und [ <sub>NP</sub> ein Kosi] [ <sub>VP</sub> diest].	
[ <sub>NP</sub> Jeder Kosi] [ <sub>VP</sub> meselt] und [ <sub>NP</sub> jeder Kosi] [ <sub>VP</sub> diest].	
[ <sub>NP</sub> Kein Kosi] [ <sub>VP</sub> meselt] und [ <sub>NP</sub> kein Kosi] [ <sub>VP</sub> diest].	

Einige Aspekte der Bedeutung sind von der Bedeutung der Inhaltswörtern unabhängig

## AN IMPORTANT JOB

This is a story about four people named **Everybody**, **Somebody**, **Anybody** and **Nobody**.

There was an important job to be done and Everybody was sure that Somebody would do it.

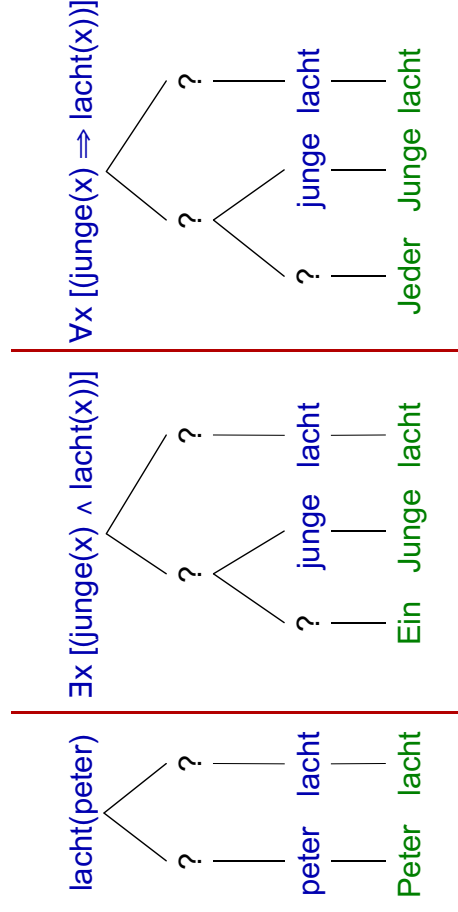
Anybody could have done it, but Nobody did it.

Somebody got angry about that, because it was Everybody's job.

Everybody thought Anybody could do it, but Nobody realised that Everybody wouldn't do it.

It ended up that Everybody blamed Somebody when Nobody did what Anybody could have.

## Beispiel



## Prädikatenlogische Repräsentation: Inventar

### Nicht-logisches Inventar

- (freie, verfügbare Symbole)
- Konstanten: **peter**, ...
- Prädikate (einstellig): **junge**, **lacht**, **winkt**, ...

### Logisches Inventar

- (feste, logische Symbole)
- Junktoren, einstellig:  $\neg$  (Negation)
- zweistellig:  $\wedge$  (Konjunktion),  $\vee$  (Disjunktion),  $\Rightarrow$  (Implikation)
- Quantoren:  $\forall$  (All-Quantor),  $\exists$  (Existenzquantor)
- Variablen: **x**, **y**, **z**, ...

### Hilfssymbole

- Klammern, Komma

## Prädikatenlogische Repräsentation: formale Sprache

### Terme

- Konstanten
- Variablen
- (Kombinationen aus Funktionssymbolen und Termen)

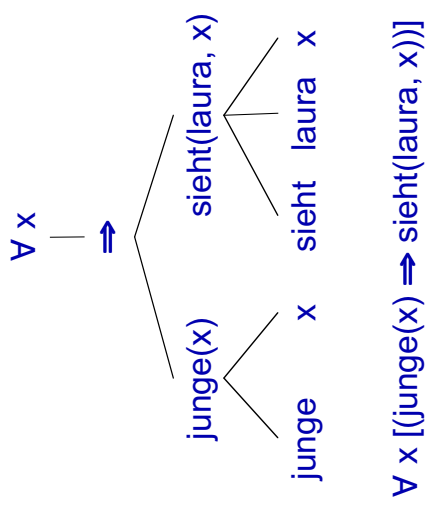
### Formeln

- Kombinationen aus einem Prädikat und einem Term:  $\text{lacht}(\text{peter}), \text{winkt}(x), \text{junge}(y)$
- (Kombinationen aus n-stelligen Relationssymbolen und n Termen)
- Negation einer Formel:  $\neg \text{lacht}(\text{peter})$
- Kombination von zwei Formeln mit einem zweistelligen Junktor:  $(\text{winkt}(\text{peter}) \wedge \neg \text{lacht}(\text{peter}))$
- Kombination aus Quantor, Variable und Formel:  $\exists x [\text{lacht}(x)]$

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-9

## Darstellung der hierarchischen Struktur von Formeln



C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-10

## Prädikatenlogische Repräsentation: Interpretation

### Eine Interpretation einer prädikatenlogischen Sprache

- ist gegeben durch eine Abbildung der **frei verfügbaren Symbole** auf passende Entitäten
  - Konstanten  $\rightarrow$  Objekte
  - Prädikate  $\rightarrow$  Mengen von Objekten
  - Relationssymbole  $\rightarrow$  Mengen von Objekt-Tupeln
  - Funktionssymbole  $\rightarrow$  Funktionen
- determiniert **eindeutig**, welchen Wert ein komplexer geschlossener Ausdruck erhält.
  - prädikatenlogische Grammatik ist eindeutig
  - Interpretation der logischen Symbole ist fest vorgegeben

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-11

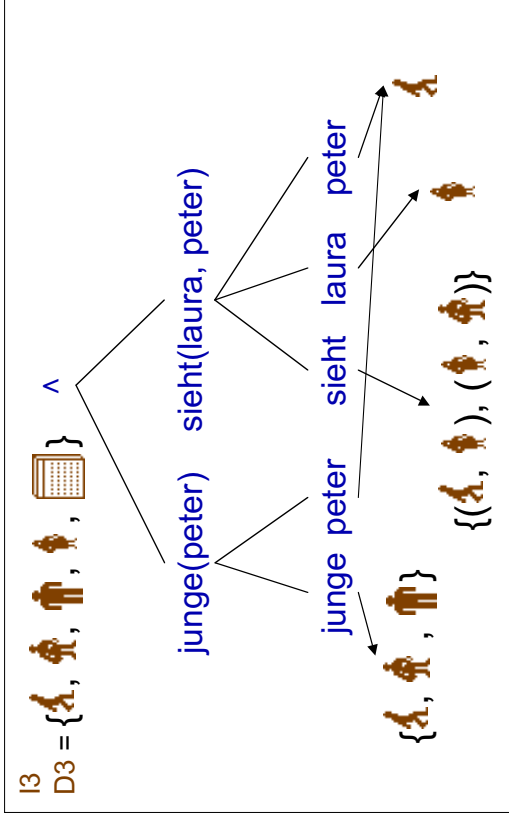
## Zwei Interpretationen

Domäne						
junge	✓			✓		✓
lacht	✓				✓	✓
winkt			✓		✓	✓
peter	✓					
Domäne						
junge				✓		
lacht					✓	
winkt					✓	✓
peter						✓

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-12

### Interpretation 3: freie Symbole



### Interpretation zusammengesetzter Ausdrücke

Sei  $D$  eine Menge und  $I$  eine Interpretation in  $D$  für die

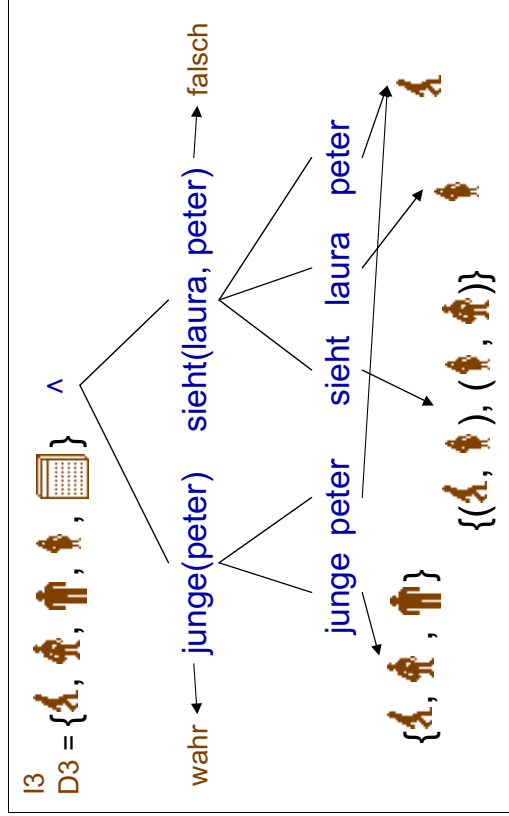
Symbole  $peter$ ,  $laura$ ,  $lacht$ ,  $winkt$ ,  $junge$ ,  $haus$ ,  $sieht$

- da  $peter$ ,  $laura$  Konstanten sind:  $I(peter)$ ,  $I(laura) \in D$
- da  $lacht$ ,  $winkt$ ,  $junge$  Prädikate sind:  $I(lacht)$ ,  $I(winkt)$ ,  $I(junge)$ ,  $I(haus) \subseteq D$
- da  $sieht$  ein zweistelliges Relationssymbol ist:  $I(sieht) \subseteq D^2$

Interpretation von atomare Formeln

- $I(junge(peter)) = \text{wahr}$ , wenn  $I(peter) \in I(junge)$ , sonst falsch
- $I(lacht(peter)) = \text{wahr}$ , wenn  $I(peter) \in I(lacht)$ , sonst falsch
- $I(winkt(laura)) = \text{wahr}$ , wenn  $I(laura) \in I(winkt)$ , sonst falsch
- $I(sieht(laura, peter)) = \text{wahr}$ , wenn  $(I(laura), I(peter)) \in I(sieht)$ , sonst falsch

### Interpretation 3: Atomare Formeln

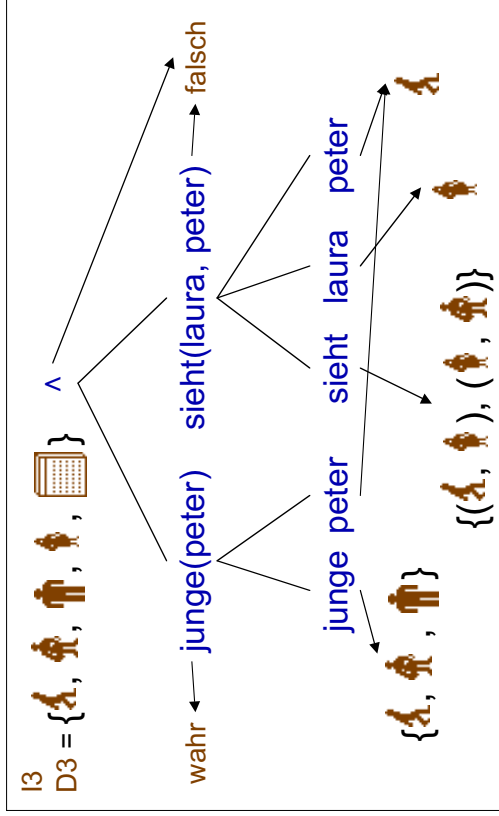


### Interpretation von Junktoren

Sei  $I$  eine Interpretation

- $I(\neg F) = \text{wahr}$ , wenn  $I(F) = \text{falsch}$ , sonst falsch
- $I((F \wedge G)) = \text{wahr}$ , wenn  $I(F) = I(G) = \text{wahr}$ , sonst falsch
- $I((F \vee G)) = \text{wahr}$ , wenn  $I(F) = \text{wahr}$  oder  $I(G) = \text{wahr}$ , sonst falsch
- $I((F \vee G)) = \text{falsch}$ , wenn  $I(F) = I(G) = \text{falsch}$ , sonst wahr
- $I((F \Rightarrow G)) = \text{falsch}$ , wenn  $I(F) = \text{wahr}$  und  $I(G) = \text{falsch}$ , sonst wahr
- $I((F \Rightarrow G)) = \text{wahr}$ , wenn  $I(F) = \text{falsch}$  oder  $I(G) = \text{wahr}$ , sonst falsch

### Interpretation 3: Formeln mit Junktoren



### Interpretation von Quantoren

#### Sei $D$ eine Menge und $I$ eine Interpretation in $D$

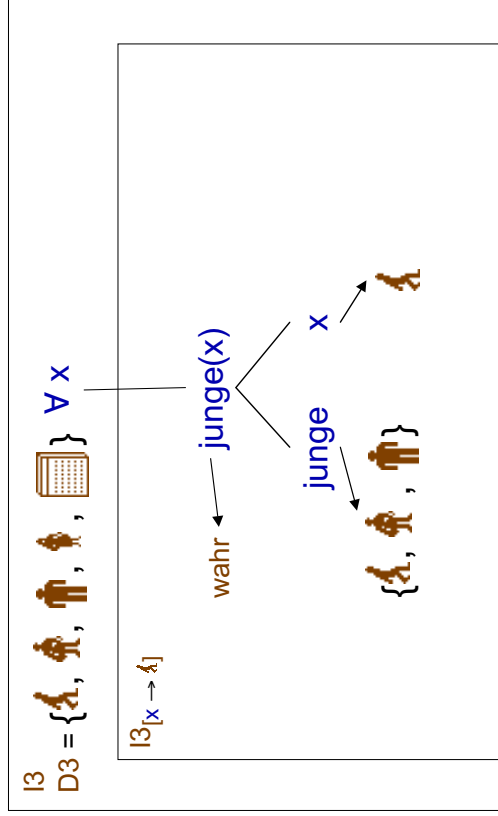
- bzgl. jeder Variable ( $x$ ) und jedes  $d \in D$  lässt sich die Interpretation  $I$  ergänzen / variieren

- $I_{[x \rightarrow d]}(X) = d$ , für alle anderen Symbole  $\pi$ :  $I_{[x \rightarrow d]}(\pi) = I(\pi)$

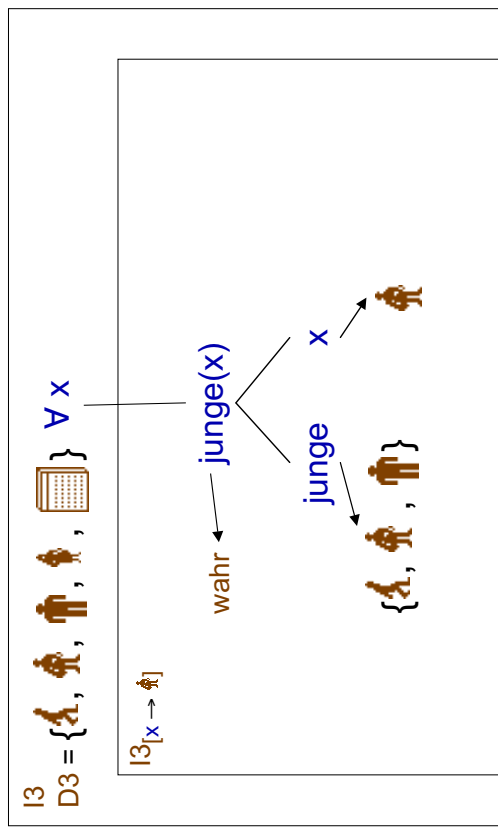
#### Quantoren-Interpretation erfordert Variation

- $I(\exists x [F]) = \text{wahr}$ , wenn es ein  $d \in D$  gibt, so dass  $I_{[x \rightarrow d]}(F) = \text{wahr}$ , sonst falsch
- $I(\forall x [F]) = \text{falsch}$ , wenn es ein  $d \in D$  gibt, so dass  $I_{[x \rightarrow d]}(F) = \text{falsch}$ , sonst wahr
- $I(\forall x [F]) = \text{wahr}$ , wenn für jede Wahl  $d \in D$   $I_{[x \rightarrow d]}(F) = \text{wahr}$ , sonst falsch

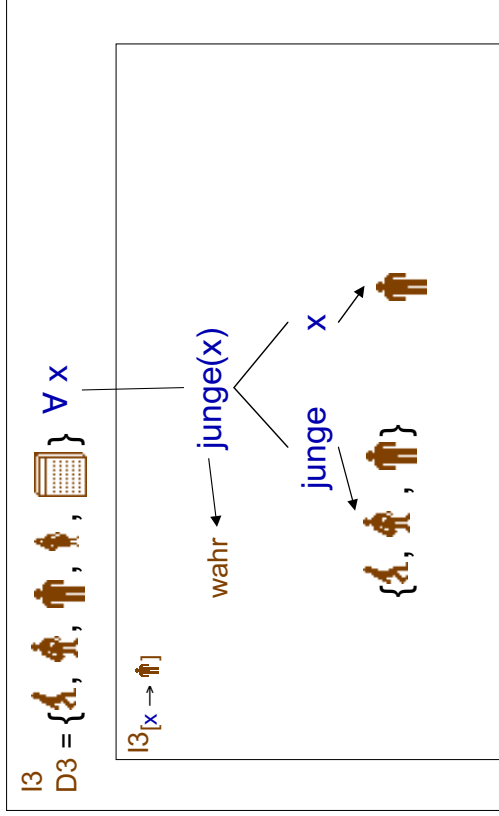
### Interpretation 3: Quantoren und Variationen (1)



### Interpretation 3: Quantoren und Variationen (2)



### Interpretation 3: Quantoren und Variationen (3)



C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 21

### Diskussion

#### Aufgabe

Welchen Wahrheitswert erhalten die Formeln

(sieht(peter, laura)  $\vee$  lacht(laura))

$\forall x$  [junge(x)  $\Rightarrow$  lacht(x)]

unter den Interpretationen

$D_1 = \{a1, a2, a3, a4, a5\}$

$I_1$ : peter  $\rightarrow a1$ , laura  $\rightarrow a3$ , junge  $\rightarrow \{a1, a2, a4\}$ ,

lacht  $\rightarrow \{a1, a3\}$ , sieht  $\rightarrow \{(a1, a3), (a1, a5)\}$

$D_2 = \{a1, a2, a3\}$

$I_2$ : peter  $\rightarrow a1$ , laura  $\rightarrow a2$ , junge  $\rightarrow \{a1, a3\}$ ,

lacht  $\rightarrow \{a1, a3\}$ , sieht  $\rightarrow \{(a1, a2), (a3, a1), (a1, a1)\}$

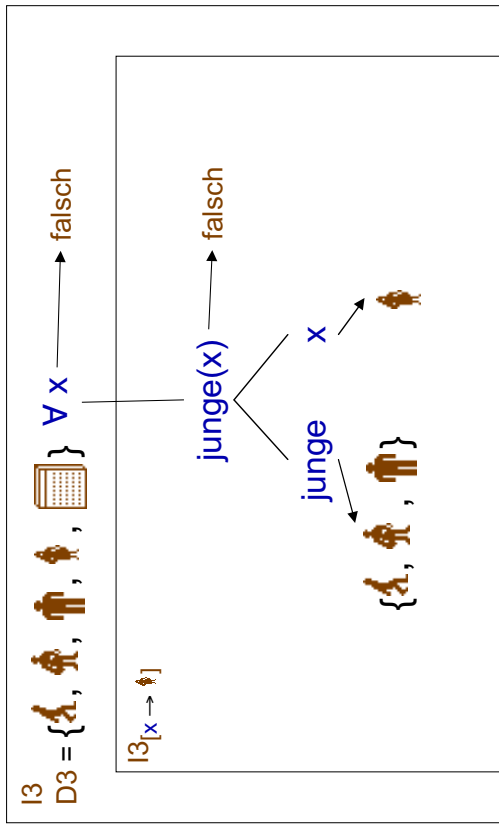
was ist mit

(sieht(peter, x)  $\vee$  lacht(x))

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 23

### Interpretation 3: Quantoren und Variationen (4)



C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 22

### Darstellung in PROLOG

Prädikatenlogische Formeln

- Prüfung auf Wohlgeformtheit

Interpretationen

Modell-Prüfer (Model Checker)

- Berechnung des Wahrheitswertes einer wohlgeformten Formel

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 24

## Prädikatenlogik in PROLOG: Beispiel

```
example_F(e1, lacht(peter)).
example_F(e2, sieht(peter, laura) v lacht(laura)).
example_F(e3, sieht(peter, A) v lacht(A)).
example_F(e4, forall(X, junge(X) -> lacht(X))).
example_F(e5, exists(X, junge(X) & winkt(X))).
example_F(e6, ~ forall(X, junge(X) -> lacht(X))).
example_F(e7, ~ exists(X, junge(X) & winkt(X))).
example_F(e8, ~ exists(X, junge(X) & ~ lacht(X))).
example_F(e9, forall(X, sieht(X, A) v sieht(A, X))).
```

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 25

exampleModels.pl

## Prädikatenlogik in PROLOG: Vokabular

```
% constant ?Atom
constant(peter).
constant(laura).

%relation ?Atom ?Stelligkeit
relation(junge,1).
relation(haus,1).
relation(winkt,1).
relation(lacht,1).
relation(rot,1).

relation(sieht,2).
```

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 27

exampleModels.pl

## Prädikatenlogik in PROLOG

- Formeln und Terme werden durch PROLOG-Terme repräsentiert
- Variablen
  - PROLOG-Variablen
- Kein Symbol-Zeichensatz in PROLOG**
- Junktoren
  - $\&$ ,  $\vee$ ,  $\rightarrow$ ,  $\sim$  anstelle von  $\wedge$ ,  $\vee$ ,  $\Rightarrow$ ,  $\neg$
  - definiert als (Infix-)Operatoren
  - Quantoren
    - exists(X, F), forall(Y, F) anstelle von  $\exists x [F]$ ,  $\forall y [F]$
- freie Symbole: Deklaration von Konstanten und Relationen
  - constant(laura). relation(junge, 1).

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 26

## Wohlgeformtheitstest (Variante 3)

**wff/1: ist das Argument ein wohlgeformter Term ?**  
wff(A) :- var(A).  
wff(A) :- \+var(A), constant(A).

**wffs/1: ist das Argument eine Liste von wohlgeformten Termen ?**  
wffs([]).  
wffs([A | RestList]) :- wff(A), wffs(RestList).

**wff/1: ist das Argument eine wohlgeformte Formel ? z.B.**  
wff(Formula1 v Formula2) :- wff(Formula1), wff(Formula2).  
wff(exists(X, Formula)) :- wff(Formula), var(X).  
wff(Formula) :-  
    Formula =.. [Pred | ArgList],  
    relation(Pred, Arity),  
    length(ArgList, Arity), wffs(ArgList).

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 28

ModelChecker3.pl

## Interpretationen in PROLOG

### Domäne

- Mengen werden durch Listen dargestellt  
`domain(i1, [a1, a2, a3, a4, a5]).`

### Interpretation

- Für die Darstellung von Abbildungen wird auch  $\rightarrow$  verwendet  
`interpretation(i1,  
  [peter  $\rightarrow$  a1,           laura  $\rightarrow$  a3,  
  junge  $\rightarrow$  [a1, a2, a4],   haus  $\rightarrow$  [a5],  
  winkt  $\rightarrow$  [a1, a2],      lacht  $\rightarrow$  [a1, a3],  
  rot  $\rightarrow$  [a5],  
  sieht  $\rightarrow$  [(a1, a3), (a1, a5)])].`

### Variablenbelegungen

`[X  $\rightarrow$  a1, Y  $\rightarrow$  a4]`

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 29

exampleModels.pl

## Modell-Prüfer 3 (Ausschnitt, vereinfacht)

```
satisfy(Formula1 & Formula2, Model, VarBelegung):-  
  satisfy(Formula1, Model, VarBelegung),  
  satisfy(Formula2, Model, VarBelegung).  
satisfy(~ Formula, Model, VarBelegung):-  
  dissatisfy(Formula, Model, VarBelegung).  
satisfy(exists(X, Formula), Model, VarBelegung):-  
  domain(Model, Domain), member(D, Domain),  
  satisfy(Formula, Model, [X  $\rightarrow$  D | VarBelegung]).  
satisfy(Formula, Model, VarBelegung):-  
  Formula =..[Pred | ArgList ],  
  evaluateArgs(ArgList, Args, Model, VarBelegung),  
  interpretation(Model, Interpretation),  
  member(Pred  $\rightarrow$  Value, Interpretation),  
  member(Args, Value).
```

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 31

ModelChecker3.pl

## Einbindung des Modell-Prüfers (Variante 3)

```
evaluate_F(Formula, Model):-  
  wff(Formula), % evaluiere nur wohlgeformte Formeln  
  domain(Model, Domain), % die Domäne  
  interpretation(Model, Interpretation), % die Interpretation  
  freieVariablenBelegung_F(Formula, Domain, []-VarBelegung),  
  % generiere eine Variablenbelegung fuer die freien Variablen  
  .. % Textausgabe  
  ((satisfy(Formula, Model, VarBelegung), % Puefe, ob erfuehlt  
  writeln('erfuehlt.')); % und gib das Ergebnis bekannt  
  (dissatisfy(Formula, Model, VarBelegung),%oder nicht erfuehlt  
  writeln('nicht erfuehlt.'))) . % gib das Ergebnis bekannt
```

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3– 30

ModelChecker3.pl

## Ausprobieren?

**exampleModels.pl:** Beispiel Operatoren und Modelle  
**modelChecker1.pl:** ohne Variablen und Quantoren  
**modelChecker2.pl:** gebundene Variablen und Quantoren, keine freien Variablen  
**modelChecker3.pl:** sogar freie Variablen

## Prädikatenlogik

### Logische Eigenschaften und Relationen

- gelten in allen Interpretationen
- sind unabhängig von der Interpretation der freien Symbole
- Tautologie, gültige Formel
  - **wahr** unter jeder Interpretation  
z.B.  $\text{lacht}(\text{peter}) \vee \neg \text{lacht}(\text{peter})$
- Kontradiktion, widersprüchliche Formel
  - **falsch** unter jeder Interpretation  
z.B.  $\text{lacht}(\text{peter}) \wedge \neg \text{lacht}(\text{peter})$
- Folgerung: aus **F** folgt **G**, wenn unter jeder Interpretation, in der **F wahr** ist, auch **G wahr** ist
  - z.B.  $\text{lacht}(\text{peter}) \wedge \text{winkt}(\text{peter}) \models \text{lacht}(\text{peter})$

## Logische Eigenschaften und Relationen

### gelten unter allen Interpretationen

- in der Prädikatenlogik gibt es unendlich viele Interpretationen

### können durch Beweisverfahren festgestellt werden

- z.B. Resolution ( $\rightarrow$  F1-Vorlesung)
- z.B. Tableau-Beweiser ( $\rightarrow$  LOS-Vorlesung)

### aber: Prädikatenlogik (PL) ist nur Semi-Entscheidbar

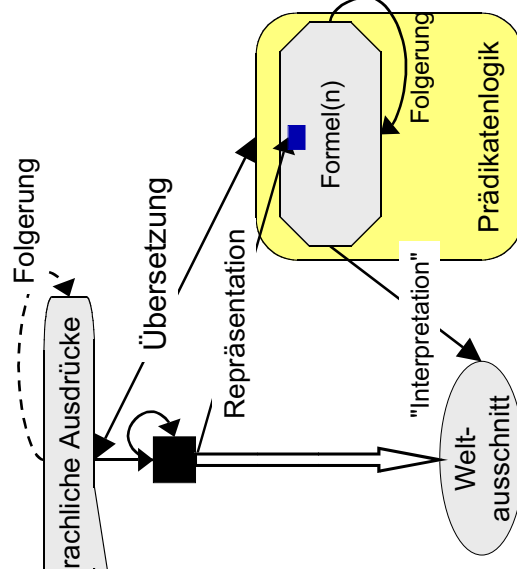
- für jedes Beweisverfahren gibt es eine Aufgabe, die es nicht (in endlicher Zeit) bearbeiten kann
- man muss damit rechnen, kein Ergebnis zu erhalten

### Bestimmte Fragmente der PL sind entscheidbar

## Formale Linguistik

### Modellierung von Bedeutung

- durch systematische Abbildung natürlicher sprachlicher Sätze auf prädikatenlogische Formeln,
- so dass (empirische) Folgerungsbeziehungen der natürlichen Sprache als nachweisbare Folgerungen in der Logik resultieren



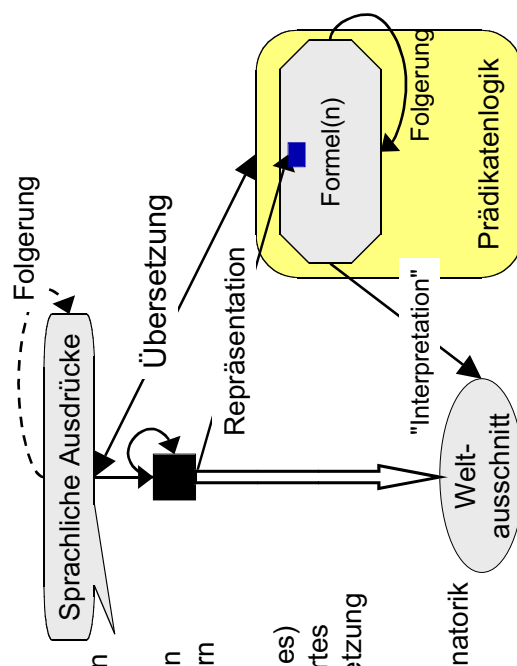
## Formale Linguistik

### Logik dient

- zur Repräsentation der Bedeutungen
- zur Kommunikation zwischen Forschern

### Ziel

- (möglichst einfaches) prinzipiengesteuertes Modell der Übersetzung
- Aufdeckung der Systematik der Bedeutungskombinatorik



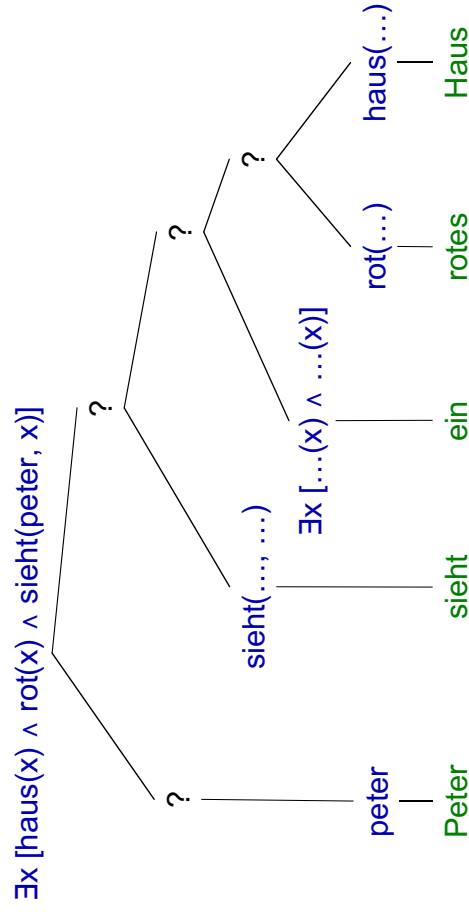
## Systematische Variationen

NP-Variation	VP-Konjunktion
$[_{NP} \text{ Peter}] [_{VP} \text{ lacht}]$ $\text{lacht}(\text{peter})$	$[_{NP} \text{ Peter}] [_{VP} \text{ lacht und winkt}]$ $(\text{lacht}(\text{peter}) \wedge \text{winkt}(\text{peter}))$
$[_{NP} \text{ Ein Junge}] [_{VP} \text{ lacht}]$ $\exists x [(junge(x) \wedge \text{lacht}(x))]$	$[_{NP} \text{ Ein Junge}] [_{VP} \text{ lacht und winkt}]$ $\exists x [(junge(x) \wedge (\text{lacht}(x) \wedge \text{winkt}(x)))]$
$[_{NP} \text{ Jeder Junge}] [_{VP} \text{ lacht}]$ $\forall x [(junge(x) \Rightarrow \text{lacht}(x))]$	$[_{NP} \text{ Jeder Junge}] [_{VP} \text{ lacht und winkt}]$ $\forall x [(junge(x) \Rightarrow (\text{lacht}(x) \wedge \text{winkt}(x)))]$
$[_{NP} \text{ Kein Junge}] [_{VP} \text{ lacht}]$ $\neg \exists x [(junge(x) \wedge \text{lacht}(x))]$	$[_{NP} \text{ Kein Junge}] [_{VP} \text{ lacht und winkt}]$ $\neg \exists x [(junge(x) \wedge (\text{lacht}(x) \wedge \text{winkt}(x)))]$
Satz-Konjunktion	
$[_{NP} \text{ Peter}] [_{VP} \text{ lacht}]$ und $[_{NP} \text{ Peter}] [_{VP} \text{ winkt}]$ $(\text{lacht}(\text{peter}) \wedge \text{winkt}(\text{peter}))$	
$[_{NP} \text{ Ein Junge}] [_{VP} \text{ lacht}]$ und $[_{NP} \text{ ein Junge}] [_{VP} \text{ winkt}]$ $\exists x [(junge(x) \wedge \text{lacht}(x))] \wedge \exists y [(junge(y) \wedge \text{winkt}(y))]$	
$[_{NP} \text{ Jeder Junge}] [_{VP} \text{ lacht}]$ und $[_{NP} \text{ jeder Junge}] [_{VP} \text{ winkt}]$ $\forall x [(junge(x) \Rightarrow \text{lacht}(x))] \wedge \forall y [(junge(y) \Rightarrow \text{winkt}(y))]$	
$[_{NP} \text{ Kein Junge}] [_{VP} \text{ lacht}]$ und $[_{NP} \text{ kein Junge}] [_{VP} \text{ winkt}]$ $\neg \exists x [(junge(x) \wedge \text{lacht}(x))] \wedge \neg \exists y [(junge(y) \wedge \text{winkt}(y))]$	

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-37

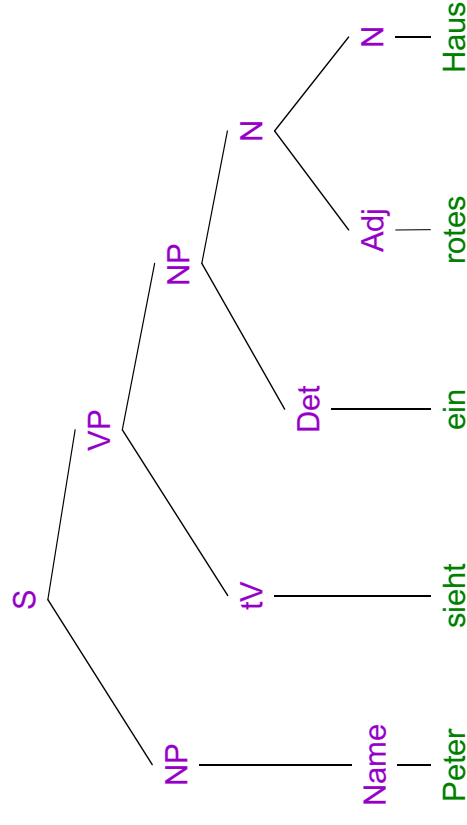
## Wahrheitsbedingungen: Peter sieht ein rotes Haus.



C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-39

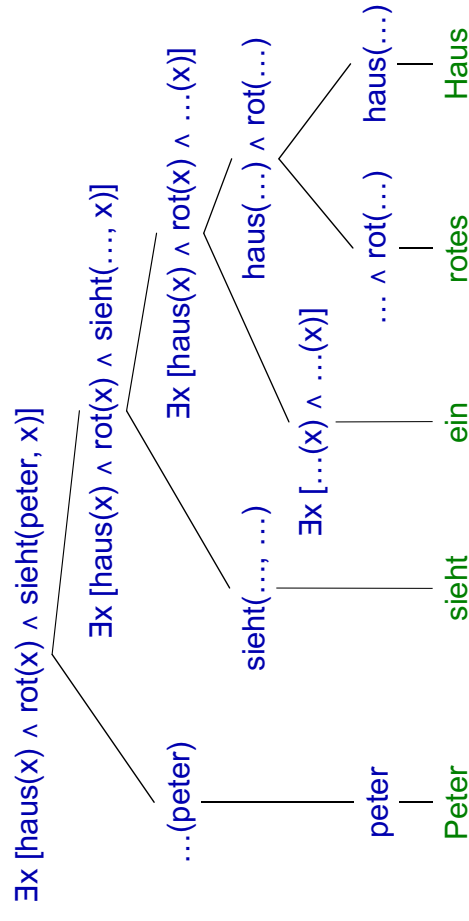
## Peter sieht ein rotes Haus.



C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-38

## Wahrheitsbedingungen: Peter sieht ein rotes Haus.

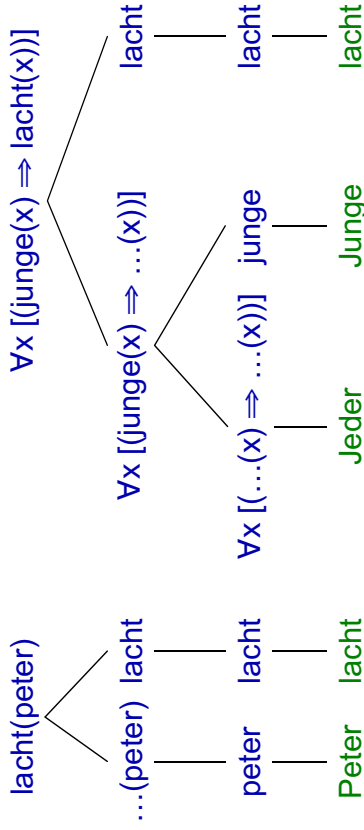


C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3-40



## Beispiel



## Grundidee

### die Bedeutung eines Ausdrucks

- beinhaltet eine eigentlichen, inhaltlichen Beitrag
- weist (wohldefinierte) Lücken auf, die der Kontext spezifiziert
- Beitrag und Lücken werden zu prädikatenlogischen Fragmenten zusammengesetzt

### der Typ des Ausdrucks

- determiniert, welche Lücken vom Kontext zu füllen sind

### Regeln

- determinieren, wie die Lücken gefüllt werden

### PROLOG-Umsetzung

- Lücken und Prädikatenlogik-Variablen werden durch Prolog-Variablen repräsentiert

## Konjunktionen (F2)

### verknüpfen zwei Sätze zu einem Satz

#### Lexikon

conj(conj/ (-und), [F, G]/(F & G)) ----> lexem(und).  
 conj(conj/ (-oder), [F, G]/(F v G)) ----> lexem(oder).

#### Regeln

s(s/[S1,C,S2], For) ----> s(S1, F), conj(C, [F, G]/For), s(S2, G).

Parser-Information  
 Syntaxbaum  
 Semantik  
 Wortform

inhaltlicher Beitrag  
 interne Lücke  
 Lücken-Info nach außen geben  
 Lücken-Info nutzen und füllen

## Konjunktionen (F2)

### verknüpfen zwei Sätze zu einem Satz

#### Lexikon

conj(conj/ (-und), [F, G]/(F & G)) ----> lexem(und).  
 conj(conj/ (-oder), [F, G]/(F v G)) ----> lexem(oder).

#### Regeln

s(s/[S1,C,S2], For) ----> s(S1, F), conj(C, [F, G]/For), s(S2, G).

vp(vp/[VP1,C,VP2], [X]/For) ----> vp(VP1, [X]/F),  
 conj(C, [F, G]/For), vp(VP2,[X]/G).

## Verbalphrasen und Sätze (F1, F2)

### Lexikon

iv(iv/ (-lacht), [X]/lacht(X)) ----> lexem(lacht).  
iv(iv/ (-winkt), [X]/winkt(X)) ----> lexem(winkt).  
tv(tv/ (-sieht), [X, Y]/sieht(X, Y)) ----> lexem(sieht).

### Regeln

vp(vp/[IV], [X]/For) ----> iv(IV, [X]/For).  
vp(vp/[TV,NP], [X]/For) ----> tv(TV, [X, Y]/G),  
np(NP, [Y, G]/For).

s(s/[NP,VP], For) ----> np(NP, [X, G]/For), vp(VP, [X]/G).

## Nominalphrasen (F1, F2)

### Determinatoren verknüpfen Eigenschaften von Nomen und VPs

#### Lexikon

det(det/ (-ein), [X, F, G]/exists(X, F & G)) ----> lexem(ein).  
det(det/ (-jeder), [X, F, G]/forall(X, F -> G)) ----> lexem(jeder).  
det(det/ (-kein), [X, F, G]/(~ exists(X, F & G))) ----> lexem(kein).

pn(pn/ (-peter), peter) ----> lexem(peter).

pn(pn/ (-laura), laura) ----> lexem(laura).

#### Regel

np(np/[Det, N], [X, G]/For) ----> det(Det, [X, F, G]/For), n(N, [X]/F).  
np(np/[PN], [Name, For]/For) ----> pn(PN, Name).

## Nomen und Adjektive (F1, F2)

### Nomen drücken Eigenschaften aus Adjektive modifizieren Eigenschaften

#### Lexikon

n(n/ (-haus), [X]/(haus(X))) ----> lexem(haus).  
n(n/ (-junge), [X]/(junge(X))) ----> lexem(junge).  
adj(adj/ (-rotes), [X, F]/(rot(X & F))) ----> lexem(rotes).

#### Regel

n(n/[Adj, N], [X]/For) ----> adj(Adj, [X, F]/For),  
n(N, [X]/F).

## Grammatik mit Syntaxbaum und Semantik (F2)

s(s/[NP,VP], For) ----> np(NP, [X, G]/For), vp(VP, [X]/G).

np(np/[PN], [Name, For]/For) ----> pn(PN, Name).

np(np/[Det, N], [X, G]/For) ----> det(Det, [X, F, G]/For), n(N, [X]/F).

n(n/[Adj, N], [X]/For) ----> adj(Adj, [X, F]/For), n(N, [X]/F).

vp(vp/[IV], [X]/For) ----> iv(IV, [X]/For).

vp(vp/[TV,NP], [X]/For) ----> tv(TV, [X, Y]/G),  
np(NP, [Y, G]/For).

vp(vp/[VP1,C,VP2], [X]/For) ----> vp(VP1, [X]/F), conj(C, [F, G]/For),  
vp(VP2, [X]/G).

s(s/[S1,C,S2], For) ----> s(S1, F), conj(C, [F, G]/For),  
s(S2, G).

## Lexikon mit Syntax und Semantik (F2)

n(n/ (-haus), [X]/(haus(X))) ----> lexem(haus).  
 n(n/ (-junge), [X]/(junge(X))) ----> lexem(junge).  
 adj(adj/ (-rotes), [X, F]/(rot(X) & F)) ----> lexem(rotes).  
 det(det/ (-ein), [X, F, G]/exists(X, F & G)) ----> lexem(ein).  
 det(det/ (-jeder), [X, F, G]/forall(X, F -> G)) ----> lexem(jeder).  
 det(det/ (-kein), [X, F, G]/(~ exists(X, F & G))) ----> lexem(kein).  
 pn(pn/ (-peter), peter) ----> lexem(peter).  
 pn(pn/ (-laura), laura) ----> lexem(laura).  
 iv(iv/ (-lacht), [X]/lacht(X)) ----> lexem(lacht).  
 iv(iv/ (-winkt), [X]/winkt(X)) ----> lexem(winkt).  
 tv(tv/ (-sieht), [X, Y]/sieht(X, Y)) ----> lexem(sieht).  
 conj(conj/ (-und), [F, G]/(F & G)) ----> lexem(und).  
 conj(conj/ (-oder), [F, G]/(F v G)) ----> lexem(oder).

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 53

## Welche Formeln werden erzeugt? (Auswahl)

NP-Variation	VP-Konjunktion
[NP Peter] [VP lacht]. [NP Ein Junge] [VP lacht]. [NP Jeder Junge] [VP lacht]. [NP Kein Junge] [VP lacht].	[NP Peter] [VP lacht und winkt]. [NP Ein Junge] [VP lacht und winkt]. [NP Jeder Junge] [VP lacht und winkt]. [NP Kein Junge] [VP lacht und winkt].
Satz-Konjunktion	
[NP Peter] [VP lacht] und [NP Peter] [VP winkt]. [NP Ein Junge] [VP lacht] und [NP ein Junge] [VP winkt]. [NP Jeder Junge] [VP lacht] und [NP jeder Junge] [VP winkt]. [NP Kein Junge] [VP lacht] und [NP kein Junge] [VP winkt].	

Welche Aussagen/Formeln sind in welchen Modellen wahr ?

C. Eschenbach: Semantische Sprachverarbeitung, Sommer 2004

3- 54