

Formale Semantik

o6. Quantifikation und Modelltheorie

Roland Schäfer

Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena

stets aktuelle Fassungen: <https://github.com/rsling/VL-Deutsche-Syntax>

- 1 From PC to F1
 - Taking stock
 - Pronouns and context
 - Phrase structure version of PC
 - Trees
 - C-command
- 2 Model theory
 - Models and valuations
 - Assignment functions
- Modified assignment functions
- 3 Problems with natural language
 - Restricted quantification
 - Variable binding and scope
 - Pre-spellout movement
 - LF movement
- 4 Quantification in English: F2
 - Movement rules
 - Fragment F2

From PC to F1

- before we turn to quantification in F1/F2 English:

- before we turn to quantification in F1/F2 English:
- names refer to individuals

- before we turn to quantification in F1/F2 English:
- names refer to individuals
- itr. verbs refer to sets of individuals

- before we turn to quantification in F1/F2 English:
- names refer to individuals
- itr. verbs refer to sets of individuals
- tr. verbs refer to sets of ordered pairs of individuals

- before we turn to quantification in F1/F2 English:
- names refer to individuals
- itr. verbs refer to sets of individuals
- tr. verbs refer to sets of ordered pairs of individuals
- sentences refer to truth values

- ***This*** drives a Golf.

- ***This*** *drives a Golf.*
- *this* = a pronominal NP

Reference of pronouns

- ***This*** *drives a Golf.*
- *this* = a pronominal NP
- denotes an individual

- ***This*** *drives a Golf.*
- *this* = a pronominal NP
- denotes an individual
- but not rigidly

- ***This*** drives a Golf.
- *this* = a pronominal NP
- denotes an individual
- but not rigidly
- fixed only within a specific context (SOA)

- quantified expression: $(\forall x)Px$

- quantified expression: $(\forall x)Px$
- *for all assignments of 'this', 'this' has property P*

- quantified expression: $(\forall x)Px$
- *for all assignments of 'this', 'this' has property P*
- Q evaluation in PC is algorithmic

- quantified expression: $(\forall x)Px$
- *for all assignments of 'this', 'this' has property P*
- Q evaluation in PC is algorithmic
- variables interpreted like definite pronominal NPs (within a fixed context)

- $a \rightarrow \text{const}, \text{var}$

- $a \rightarrow \text{const, var}$
- $\text{conn} \rightarrow \wedge, \vee, \rightarrow, \leftrightarrow$

- $a \rightarrow \text{const, var}$
- $\text{conn} \rightarrow \wedge, \vee, \rightarrow, \leftrightarrow$
- $\text{neg} \rightarrow \neg$

Categories and lexicon

- $a \rightarrow \text{const, var}$
- $\text{conn} \rightarrow \wedge, \vee, \rightarrow, \leftrightarrow$
- $\text{neg} \rightarrow \neg$
- $Q \rightarrow \exists, \forall$

- $\text{pred}_1 \rightarrow P, Q$

- $\text{pred}_1 \rightarrow P, Q$
- $\text{pred}_2 \rightarrow R$

- $\text{pred}_1 \rightarrow P, Q$
- $\text{pred}_2 \rightarrow R$
- $\text{pred}_3 \rightarrow S$

- $\text{pred}_1 \rightarrow P, Q$
- $\text{pred}_2 \rightarrow R$
- $\text{pred}_3 \rightarrow S$
- $\text{const} \rightarrow b, c$

- $\text{pred}_1 \rightarrow P, Q$
- $\text{pred}_2 \rightarrow R$
- $\text{pred}_3 \rightarrow S$
- $\text{const} \rightarrow b, c$
- $\text{var} \rightarrow x_1, x_2, \dots, x_n$

- $wff \rightarrow pred_n a_1 a_2 \dots a_n$

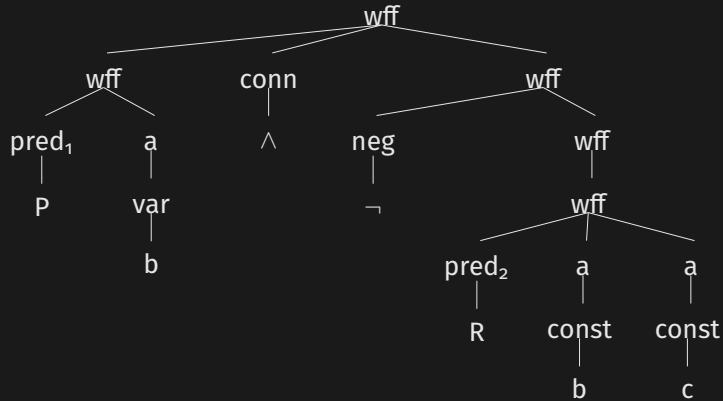
Phrase structure

- $wff \rightarrow pred_n a_1 a_2 \dots a_n$
- $wff \rightarrow neg\ wff$

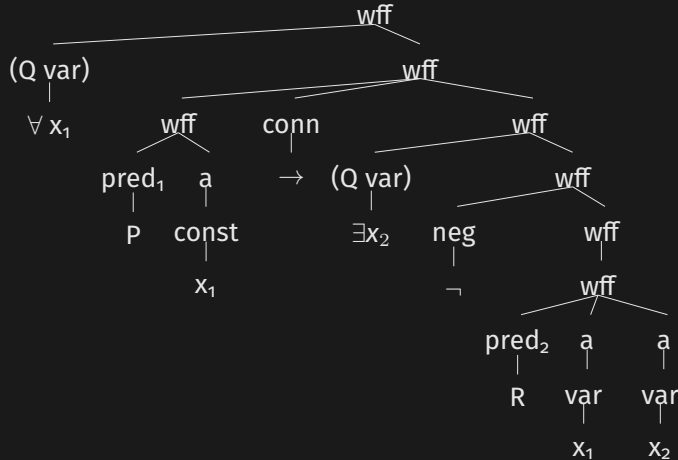
- $wff \rightarrow pred_n a_1 a_2 \dots a_n$
- $wff \rightarrow neg\ wff$
- $wff \rightarrow wff\ con\ wff$

- $wff \rightarrow \text{pred}_n a_1 a_2 \dots a_n$
- $wff \rightarrow \text{neg } wff$
- $wff \rightarrow wff \text{ con } wff$
- $wff \rightarrow (Q \text{ var}) wff$

A wff without Q



A wff with Q's



Definition of c-command

- Node A **c-commands** (constituent-commands) node B iff

Definition of c-command

- Node A **c-commands** (constituent-commands) node B iff
 - ▶ **A does not dominate B** and

Definition of c-command

- Node A **c-commands** (constituent-commands) node B iff
 - ▶ A does not dominate B and
 - ▶ and the first branching node dominating A also dominates B.

Definition of c-command

- Node A **c-commands** (constituent-commands) node B iff
 - ▶ A does not dominate B and
 - ▶ and the first branching node dominating A also dominates B.
- The definition in CM allows a node to dominate itself.

- in configurational tree-structures:

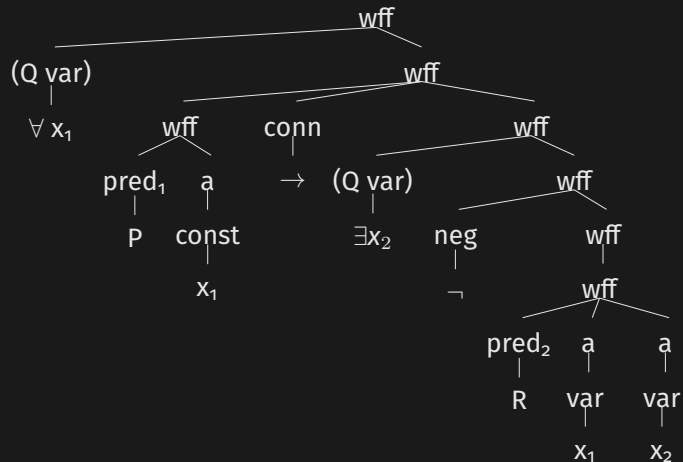
Configurational binding

- in configurational tree-structures:
- A variable is bound by the closest c-commanding coindexed quantifier.

Configurational binding

- in configurational tree-structures:
- A variable is bound by the closest c-commanding coindexed quantifier.
- scope = binding domain

A wff with Q's



Model theory

- remember T-sentences: **S of L is true in v iff p.**

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns
 - ★ individuals to names

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns
 - ★ individuals to names
 - ★ sets of n-tuples of individuals to pred_n

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns
 - ★ individuals to names
 - ★ sets of n-tuples of individuals to pred_n
- g is function from variables to individuals in \mathcal{M}

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns
 - ★ individuals to names
 - ★ sets of n-tuples of individuals to pred_n
- g is function from variables to individuals in \mathcal{M}
- we evaluate: $\llbracket \alpha \rrbracket^{\mathcal{M}_n, g_n}$

- remember T-sentences: **S of L is true in v iff p.**
- \mathcal{M} is a model of the accessible universe of discourse
 - ▶ $\mathcal{M} = \langle U_n, V_n \rangle$
 - ▶ U_n = the set of accessible individuals (domain)
 - ▶ V_n = a valuation function which assigns
 - ★ individuals to names
 - ★ sets of n-tuples of individuals to pred_n
- g is function from variables to individuals in \mathcal{M}
- we evaluate: $\llbracket \alpha \rrbracket^{\mathcal{M}_n, g_n}$
- *the extension of α relative to \mathcal{M}_n and g_n*

- V_n evaluates *statically*

Fixed and context-bound denotation

- V_n evaluates *statically*
- Q's require flexible valuation of pronominal matrices

Fixed and context-bound denotation

- V_n evaluates *statically*
- Q's require flexible valuation of pronominal matrices
- g_n is like V_n for constants, only flexible

Fixed and context-bound denotation

- V_n evaluates *statically*
- Q's require flexible valuation of pronominal matrices
- g_n is like V_n for constants, only flexible
- it can *iterate through* U_n

- V_n evaluates *statically*
- Q's require flexible valuation of pronominal matrices
- g_n is like V_n for constants, only flexible
- it can *iterate through* U_n
- initial assignment can be anything:

$$g_1 = \left[\begin{array}{l} x_1 \rightarrow \textit{Herr Webelhuth} \\ x_2 \rightarrow \textit{Frau Eckardt} \\ x_3 \rightarrow \textit{Turm – Mensa} \end{array} \right]$$

- for each Q loop, one modification

Iterating through U_n

- for each Q loop, one modification
- read $g_n [d/x_m]$ as
‘...relative to g_n where x_m is reassigned to d ’

Iterating through U_n

- for each Q loop, one modification
- read $g_n[d/x_m]$ as
‘...relative to g_n where x_m is reassigned to d ’
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[Eckardt/x_1]} = \textit{Frau Eckardt}$

Iterating through U_n

- for each Q loop, one modification
- read $g_n [d/x_m]$ as
‘...relative to g_n where x_m is reassigned to d ’
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [Eckardt/x_1]} = \textit{Frau Eckardt}$
- $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[Eckardt/x_1]Mensa/x_2]} = \textit{Mensa}$

- $\llbracket (\forall x_1) P x_1 \rrbracket^{\mathcal{M}_1, g_1}$

- $\llbracket (\forall x_1) Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- start with initial assignment: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$

- $\llbracket (\forall x_1) Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- start with initial assignment: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- modify: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[Eckardt/x_1]} = \textit{Eckardt}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$

- $\llbracket (\forall x_1) Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- start with initial assignment: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- modify: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[Eckardt/x_1]} = \textit{Eckardt}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- modify: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[Mensa/x_1]} = \textit{Mensa}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$

- $\llbracket (\forall x_1) Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- start with initial assignment: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- modify: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[\textit{Eckardt}/x_1]} = \textit{Eckardt}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- modify: $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1[\textit{Mensa}/x_1]} = \textit{Mensa}$
check: $\llbracket Px_1 \rrbracket^{\mathcal{M}_1, g_1}$
- iff the answer was never 0, then $\llbracket (\forall x_1) Px_1 \rrbracket^{\mathcal{M}_1, g_1} = 1$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2) p_{x_1 x_2} \rrbracket^{\mathcal{M}_1, g_1}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2) p_{x_1 x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2) p_{x_1 x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Eckardt}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2) p_{x_1 x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\textit{Webelhuth}/x_2]} = \textit{Webelhuth}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \textit{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\textit{Webelhuth}/x_2]} = \textit{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\textit{Mensa}/x_2]} = \textit{Mensa}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [Webelhuth/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [Mensa/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [Eckardt/x_1]} = \text{Eckardt}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [Webelhuth/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [Mensa/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [Eckardt/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [Eckardt/x_1]} = \text{Eckardt}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1]\text{Webelhuth}/x_2]} = \text{Webelhuth}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1]\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1]\text{Mensa}/x_2]} = \text{Mensa}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1]\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1]\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_1]} = \text{Mensa}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2) P_{x_1 x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth} / x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa} / x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt} / x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt} / x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt} / x_1] \text{Webelhuth} / x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt} / x_1] \text{Mensa} / x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa} / x_1]} = \text{Mensa}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa} / x_1]} = \text{Eckardt}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1] \text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1] \text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_1]} = \text{Mensa}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Mensa}/x_1] \text{Webelhuth}/x_2]} = \text{Webelhuth}$

Multiple Q's: subloops

- $\llbracket (\forall x_1)(\exists x_2)P_{x_1x_2} \rrbracket^{\mathcal{M}_1, g_1}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Eckardt}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1] \text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Eckardt}/x_1] \text{Mensa}/x_2]} = \text{Mensa}$
- $\llbracket x_1 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_1]} = \text{Mensa}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [\text{Mensa}/x_1]} = \text{Eckardt}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Mensa}/x_1] \text{Webelhuth}/x_2]} = \text{Webelhuth}$
 - ▶ $\llbracket x_2 \rrbracket^{\mathcal{M}_1, g_1 [[\text{Mensa}/x_1] \text{Mensa}/x_2]} = \text{Mensa}$

Problems with natural language

- quantifying expressions in NL beyond \forall and \exists

- quantifying expressions in NL beyond \forall and \exists
- some seem to work differently:

- quantifying expressions in NL beyond \forall and \exists
- some seem to work differently:
- *All patients adore Dr. Rick Dagless M.D.*
 $(\forall x_1)Px_1 \rightarrow Ax_1d$ (ok)

- quantifying expressions in NL beyond \forall and \exists
- some seem to work differently:
- *All patients* adore Dr. Rick Dagless M.D.
 $(\forall x_1)Px_1 \rightarrow Ax_1d$ (ok)
- but: *Most patients* adore Dr. Rick Dagless M.D.
 $(MOST x_1)Px_1 \rightarrow Ax_1d$ (wrong interpretation)

- quantifying expressions in NL beyond \forall and \exists
- some seem to work differently:
- *All patients* adore Dr. Rick Dagless M.D.
 $(\forall x_1)Px_1 \rightarrow Ax_1d$ (ok)
- but: *Most patients* adore Dr. Rick Dagless M.D.
 $(MOST x_1)Px_1 \rightarrow Ax_1d$ (wrong interpretation)
- domain should be the set of patients, not individuals

- quantifying expressions in NL beyond \forall and \exists
- some seem to work differently:
- *All patients* adore Dr. Rick Dagless M.D.
 $(\forall x_1)Px_1 \rightarrow Ax_1d$ (ok)
- but: *Most patients* adore Dr. Rick Dagless M.D.
 $(MOST x_1)Px_1 \rightarrow Ax_1d$ (wrong interpretation)
- domain should be the set of patients, not individuals
- For NL: Assume that the checking domain for Q is the set denoted by CN.

- c-command condition on binding/scope fails in NL

Scope ambiguities

- c-command condition on binding/scope fails in NL
- no PNF's in NL

Scope ambiguities

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)

Scope ambiguities

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)
- ambiguities independent of Q position

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)
- *ambiguities independent of Q position*
 - ▶ *Everybody loves somebody. (ELS)*

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)
- *ambiguities independent of Q position*
 - ▶ *Everybody loves somebody. (ELS)*
 - ▶ $(\forall x_1)(\exists x_2)Lx_1x_2$

Scope ambiguities

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)
- *ambiguities independent of Q position*
 - ▶ *Everybody loves somebody. (ELS)*
 - ▶ $(\forall x_1)(\exists x_2)Lx_1x_2$
 - ▶ $(\exists x_2)(\forall x_1)Lx_1x_2$

- c-command condition on binding/scope fails in NL
- no PNF's in NL
- Q and common noun (CN) usually *in-situ* (e.g., argument position)
- *ambiguities independent of Q position*
 - ▶ *Everybody loves somebody. (ELS)*
 - ▶ $(\forall x_1)(\exists x_2)Lx_1x_2$
 - ▶ $(\exists x_2)(\forall x_1)Lx_1x_2$
- *Q ambiguity cannot be structural* (e.g., \exists will never c-command \forall)

Cases of overt movement and traces

- *wh* movement:
- *What_i will Agent Cooper solve t_i?*
-

Cases of overt movement and traces

- *wh* movement:
- *What_i will Agent Cooper solve t_i?*
-
- *passive* movement:
- *(Laura Palmer)_i was killed t_i.*
-

Cases of overt movement and traces

- *wh* movement:
 - *What_i will Agent Cooper solve t_i?*
 -
- *passive* movement:
 - *(Laura Palmer)_i was killed t_i.*
 -
- *raising* verbs:
 - *(Laura Palmer)_i seems t_i to be dead.*
 -

- construction of an independent representational level LF

Levels of representation

- construction of an independent representational level LF
- could use movement mechanism as used at surface level

Levels of representation

- construction of an independent representational level LF
- could use movement mechanism as used at surface level
- All quantifiers adjoin to the left periphery of S at LF.

Levels of representation

- construction of an independent representational level LF
- could use movement mechanism as used at surface level
- All quantifiers adjoin to the left periphery of S at LF.
- LF is constructed by syntactic rules!

- $[_{S''} \text{everybody}_i [_{S'} \text{somebody}_j [_{}_S t_i \text{ loves } t_j]]]$
-

- $[_{S''} \text{everybody}_i [_{S'} \text{somebody}_j [_S t_i \text{ loves } t_j]]]$
-
- $[_{S''} \text{somebody}_j [_{S'} \text{everybody}_i [_S t_i \text{ loves } t_j]]]$
-

Quantification in English: F2

The Q raising rule

$$[{}_S X NP Y] \Rightarrow [{}_{S'} NP_i [{}_S X t_i Y]]$$

The Q raising rule

$$[_S X NP Y] \Rightarrow [_{S'} NP_i [_S X t_i Y]]$$

- specify a PS as input and output

The Q raising rule

$$[_S X NP Y] \Rightarrow [_{S'} NP_i [_S X t_i Y]]$$

- specify a PS as input and output
- QR rule also introduces coindexing of traces

- copies all definitions from F1

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.
 - ▶ *Det* \rightarrow *every*, *some*

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.
 - ▶ $Det \rightarrow \text{every, some}$
 - ▶ $NP \rightarrow DetN_{\text{common-count}}$

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.
 - ▶ $Det \rightarrow \text{every, some}$
 - ▶ $NP \rightarrow DetN_{\text{common-count}}$
- adds the QR rule

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.
 - ▶ $Det \rightarrow \text{every, some}$
 - ▶ $NP \rightarrow DetN_{\text{common-count}}$
- adds the QR rule
- assume introduction of reasonable syntactic types/rules without specifying

- copies all definitions from F1
- adds appropriate definitions of quantifying determiners etc.
 - ▶ $Det \rightarrow \text{every, some}$
 - ▶ $NP \rightarrow DetN_{\text{common-count}}$
- adds the QR rule
- assume introduction of reasonable syntactic types/rules without specifying
- assume admissible (reasonable, possible) models \mathcal{M}

$$\llbracket \llbracket \textit{every } \beta \rrbracket_i S \rrbracket^{\mathcal{M},g} = 1 \text{ iff for all } u \in U : \\ \text{if } u \in \llbracket \beta \rrbracket^{\mathcal{M},g} \text{ then } \llbracket S \rrbracket^{\mathcal{M},g[u/t_i]}$$

A sentence containing the trace t_i with an adjoined NP_i (which consists of *every* plus the common noun β) extend to 1 iff for each individual u in the universe U which is in the set referred to by the common noun β , S denotes 1 with u assigned to the pronominal trace t_i . g is modified iteratively to check that.

$$\begin{aligned} \llbracket [a \beta]_i S \rrbracket^{\mathcal{M},g} = 1 \text{ iff for some } u \in U : \\ u \in \llbracket \beta \rrbracket^{\mathcal{M},g} \text{ and } \llbracket S \rrbracket^{\mathcal{M},g[u/t_i]} \end{aligned}$$

(similar)

Kontakt

Prof. Dr. Roland Schäfer
Institut für Germanistische Sprachwissenschaft
Friedrich-Schiller-Universität Jena
Fürstengraben 30
07743 Jena

<https://rolandschaefer.net>
roland.schaefer@uni-jena.de

Creative Commons BY-SA-3.0-DE

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ *Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland* zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie

<http://creativecommons.org/licenses/by-sa/3.0/de/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.