Contents

1	Outline of talk	3
2	Anaphora in TLG2.1Jacobson's proposal2.2Adaptation to TLG2.3Binding2.4Percolation	${f 4}{4}{5}{6}{7}$
3	Covering indefinites3.1Basic idea3.2Type Logical implementation3.3Descriptive content	
4	Variable free existential closure	13
5	Linguistic consequences 5.1 Indefinites and scope	17 17
6	Sluicing 6.1 Predictions	21 23
7	Conclusion	26

Indefinites and Sluicing A Type-Logical Approach

Amsterdam Colloquium December 18, 2001

Gerhard Jäger

Zentrum für Allgemeine Sprachwissenschaft Berlin jaeger@zas.gwz-berlin.de www.let.uu.nl/~Gerhard.Jaeger/personal

- Anaphora in Type Logical Grammar
- Extrapolation to indefinites
- Linguistic consequences:
 - Indefinites and scopeSluicing

- Anaphora in Type Logical Grammar
- Extrapolation to indefinites
- Linguistic consequences:
 - Indefinites and scopeSluicing

- Anaphora in Type Logical Grammar
- Extrapolation to indefinites
- Linguistic consequences:

 Indefinites and scope
 Sluicing

- Anaphora in Type Logical Grammar
- Extrapolation to indefinites
- Linguistic consequences:
 - Indefinites and scopeSluicing

- Anaphora in Type Logical Grammar
- Extrapolation to indefinites
- Linguistic consequences:
 - Indefinites and scope Sluicing

2.1. Jacobson's proposal

• Semantics: pronouns denote identity functions

- Syntax: third slash: "A|B" is category of anaphoric expression
- Pronouns: category np|np

2.1. Jacobson's proposal

• Semantics: pronouns denote identity functions

- Syntax: third slash: "A|B" is category of anaphoric expression
- Pronouns: category np|np

2.1. Jacobson's proposal

- Semantics: pronouns denote identity functions
- Syntax: third slash: "A|B" is category of anaphoric expression
- Pronouns: category np|np

2.1. Jacobson's proposal

- Semantics: pronouns denote identity functions
- Syntax: third slash: "A|B" is category of anaphoric expression
- Pronouns: category np|np

Natural Deduction rules for anaphora slash

$$[M:A]_i \quad \cdots \quad \frac{N:B|A}{[NM:B]_i} | E, i \qquad \begin{array}{c} \vdots \quad \frac{M:A|B}{Mx:A} i \\ \vdots \quad \vdots \quad \vdots \\ \frac{N:C}{\lambda x N:C|B} | I, i \end{array}$$

• Only constraint on anaphora resolution: The antecedent must precede the pronoun

Natural Deduction rules for anaphora slash

$$[M:A]_i \quad \cdots \quad \frac{N:B|A}{[NM:B]_i} | E, i \qquad \begin{array}{c} \vdots \quad \frac{M:A|B}{Mx:A} i \\ \vdots \quad \vdots \quad \vdots \\ \frac{N:C}{\lambda x N:C|B} | I, i \end{array}$$

• Only constraint on anaphora resolution: The antecedent must precede the pronoun

Natural Deduction rules for anaphora slash

• Only constraint on anaphora resolution: The antecedent must precede the pronoun

Natural Deduction rules for anaphora slash

$$[M:A]_i \quad \cdots \quad \frac{N:B|A}{[NM:B]_i} | E, i \qquad \begin{array}{c} \vdots \quad \frac{M:A|B}{Mx:A}i \\ \vdots \quad \vdots \quad \vdots \\ \frac{N:C}{\lambda x N:C|B} | I, i \end{array}$$

• Only constraint on anaphora resolution: The antecedent must precede the pronoun

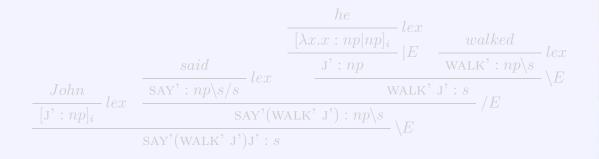
Natural Deduction rules for anaphora slash

$$[M:A]_i \quad \cdots \quad \frac{N:B|A}{[NM:B]_i} | E, i \qquad \begin{array}{c} \vdots \quad \frac{M:A|B}{Mx:A} i \\ \vdots \quad \vdots \quad \vdots \\ \frac{N:C}{\lambda x N:C|B} | I, i \end{array}$$

• Only constraint on anaphora resolution: The antecedent must precede the pronoun

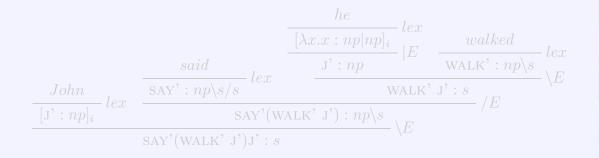
2.3. Binding

(1) John said he walked



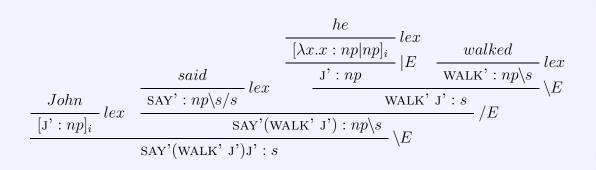
2.3. Binding

(1) John said he walked

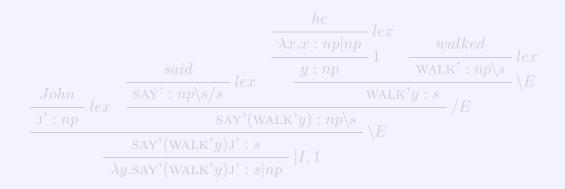


2.3. Binding

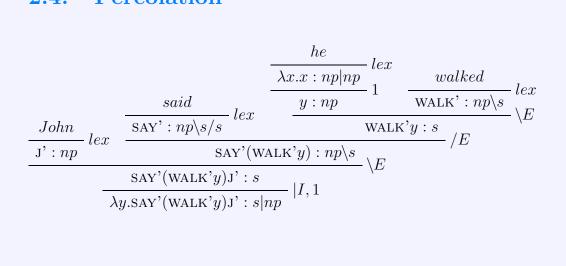
(1) John said he walked



2.4. Percolation



2.4. Percolation



3.1. Basic idea

(2) a. It movedb. Something moved

Proposal: (a) and (b) have
 the same denotation: λx.MOVE'x
 different syntactic categories

3.1. Basic idea

(2) a. It movedb. Something moved

Proposal: (a) and (b) have
 the same denotation: λx.MOVE'x
 different syntactic categories

3.1. Basic idea

(2) a. It movedb. Something moved

- Proposal: (a) and (b) have
 - the same denotation: $\lambda x.MOVE'x$ • different syntactic categories

3.1. Basic idea

(2) a. It movedb. Something moved

Proposal: (a) and (b) have
 the same denotation: λx.MOVE'x
 different syntactic categories

- yet another substructural implication, "→"
- Intuition: $A \rightsquigarrow B$: category of B-sign containing an indefinite A
- category of indefinite NPs: $np \rightsquigarrow np$
- *it* and *something* both denote the identity function on individuals

- yet another substructural implication, " \rightsquigarrow "
- Intuition: $A \rightsquigarrow B$: category of B-sign containing an indefinite A
- category of indefinite NPs: $np \rightsquigarrow np$
- *it* and *something* both denote the identity function on individuals

- yet another substructural implication, " \rightsquigarrow "
- Intuition: $A \rightsquigarrow B$: category of B-sign containing an indefinite A
- category of indefinite NPs: $np \rightsquigarrow np$
- *it* and *something* both denote the identity function on individuals

- yet another substructural implication, " \rightsquigarrow "
- Intuition: $A \rightsquigarrow B$: category of B-sign containing an indefinite A
- category of indefinite NPs: $np \rightsquigarrow np$
- *it* and *something* both denote the identity function on individuals

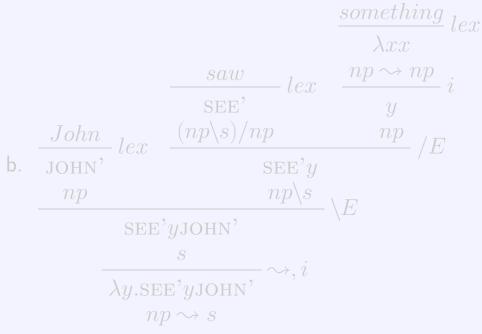
- yet another substructural implication, "→"
- Intuition: $A \rightsquigarrow B$: category of B-sign containing an indefinite A
- category of indefinite NPs: $np \rightsquigarrow np$
- *it* and *something* both denote the identity function on individuals

- indefinites function compose with their semantic environment
- Natural deduction rule:

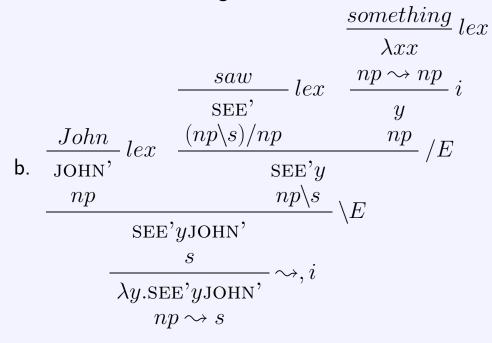
$$\frac{M:A \rightsquigarrow B}{Mx:B}i \\
\vdots \\
\frac{B}{i} \\
\frac{N:C}{\lambda x N:A \rightsquigarrow C} \sim, i$$

- indefinites function compose with their semantic environment
- Natural deduction rule:

(3) a. John saw something



(3) a. John saw something



3.3. Descriptive content

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- ||a cup|| = identity function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:
 - f(x) = 1 iff x is a cup that moved
 - $\circ f(x) = 0$ iff x is a cup that did not move
 - $\circ f(x)$ is undefined iff x is not a cup

3.3. Descriptive content

• Idea: descriptive content expresses domain restriction

- ||a|| = function that maps a property to the identity function over its extension
- ||a cup|| = identity function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:

 $\circ f(x) = 1$ iff x is a cup that moved

- $\circ f(x) = 0$ iff x is a cup that did not move
- $\circ f(x)$ is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- ||a cup|| = identity function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:
 - $\circ f(x) = 1$ iff x is a cup that moved
 - $\circ f(x) = 0$ iff x is a cup that did not move
 - $\circ f(x)$ is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- $\|a \ cup\| = identity$ function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:
 - $\circ f(x) = 1$ iff x is a cup that moved
 - $\circ f(x) = 0$ iff x is a cup that did not move
 - $\circ f(x)$ is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- $\|a \ cup\| = identity$ function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:

• f(x) = 1 iff x is a cup that moved • f(x) = 0 iff x is a cup that did not move • f(x) is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- $\|a \ cup\| = identity$ function on the set of cups
- ||a cup moved|| = partial function f from individuals to truth values:

f(x) = 1 iff x is a cup that moved
f(x) = 0 iff x is a cup that did not move
f(x) is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- $\|a \ cup\| = identity \ function \ on \ the \ set \ of \ cups$
- ||a cup moved|| = partial function f from individuals to truth values:

f(x) = 1 iff x is a cup that moved
f(x) = 0 iff x is a cup that did not move
f(x) is undefined iff x is not a cup

- Idea: descriptive content expresses domain restriction
- ||a|| = function that maps a property to the identity function over its extension
- $\|a \ cup\| = identity \ function \ on \ the \ set \ of \ cups$
- ||a cup moved|| = partial function f from individuals to truth values:

f(x) = 1 iff x is a cup that moved
f(x) = 0 iff x is a cup that did not move
f(x) is undefined iff x is not a cup

- Existential closure of a partial function: "big union" over its domain
- built in into the truth definition and the semantics of propositional operators (as in DRT)
- Relativization to syntactic categories to confine existential closure to indefinites

- Existential closure of a partial function: "big union" over its domain
- built in into the truth definition and the semantics of propositional operators (as in DRT)
- Relativization to syntactic categories to confine existential closure to indefinites

- Existential closure of a partial function: "big union" over its domain
- built in into the truth definition and the semantics of propositional operators (as in DRT)
- Relativization to syntactic categories to confine existential closure to indefinites

- Existential closure of a partial function: "big union" over its domain
- built in into the truth definition and the semantics of propositional operators (as in DRT)
- Relativization to syntactic categories to confine existential closure to indefinites

• Truth is relativized to sequence of referents and syntactic category

Definition 1 (Truth)

 $\vec{e} \models \alpha : s \quad \text{iff} \quad \alpha = 1$ $c\vec{e} \models \alpha : S | np \quad \text{iff} \quad \vec{e} \models (\alpha c) : S$ $\vec{e} \models \alpha : np \rightsquigarrow S \quad \text{iff} \quad \vec{e} \models (\bigcup_{\alpha c \text{ is defined}} (\alpha c)) : S$

• Truth is relativized to sequence of referents and syntactic category

Definition 1 (Truth)

$$\begin{array}{c} \vec{e} \models \alpha : s \quad \text{iff} \quad \alpha = 1 \\ c\vec{e} \models \alpha : S | np \quad \text{iff} \quad \vec{e} \models (\alpha c) : S \\ \vec{e} \models \alpha : np \rightsquigarrow S \quad \text{iff} \quad \vec{e} \models (\bigcup_{\substack{\alpha c \text{ is defined}}} (\alpha c)) : S \end{array}$$

(4) A cup moved

$$\vec{e} \models \|\lambda x_{\text{CUP}'x}.\text{MOVE}'x\|_g : np \rightsquigarrow s \iff \vec{e} \models \bigcup_{a \in \|\text{CUP}'\|_g} \|\text{MOVE}'\|_g(a) : s \iff \bigcup_{a \in \|\text{CUP}'\|_g} \|\text{MOVE}'\|_g(a) = 1 \iff \exists a.a \in \|\text{CUP}'\|_g \cap \|\text{MOVE}'\|_g$$

(4) A cup moved

$$\begin{split} \vec{e} &\models \|\lambda x_{\text{CUP}'x}.\text{MOVE}'x\|_g : np \rightsquigarrow s \iff \\ \vec{e} &\models \bigcup_{a \in \|\text{CUP}'\|_g} \|\text{MOVE}'\|_g(a) : s \iff \\ \bigcup_{a \in \|\text{CUP}'\|_g} \|\text{MOVE}'\|_g(a) = 1 \iff \\ \exists a.a \in \|\text{CUP}'\|_g \cap \|\text{MOVE}'\|_g \end{split}$$

- Negation is polymorphic
- indefinites in its scope are (optionally) existentially closed

anaphora slots are passed through unchanged
 Definition 2 (Negation)

$$\sim \alpha : s = 1 - \alpha$$

$$\sim \alpha : S | A = \lambda c. \sim (\alpha c)$$

$$\sim \alpha : A \rightsquigarrow S = \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)$$

• Negation is polymorphic

indefinites in its scope are (optionally) existentially closed

anaphora slots are passed through unchanged
 Definition 2 (Negation)

$$\sim \alpha : s = 1 - \alpha$$

$$\sim \alpha : S | A = \lambda c. \sim (\alpha c)$$

$$\sim \alpha : A \rightsquigarrow S = \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)$$

- Negation is polymorphic
- indefinites in its scope are (optionally) existentially closed

• anaphora slots are passed through unchanged **Definition 2 (Negation)**

$$\sim \alpha : s = 1 - \alpha$$

$$\sim \alpha : S | A = \lambda c. \sim (\alpha c)$$

$$\sim \alpha : A \rightsquigarrow S = \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)$$

- Negation is polymorphic
- indefinites in its scope are (optionally) existentially closed
- anaphora slots are passed through unchanged

Definition 2 (Negation)

$$\sim \alpha : s = 1 - \alpha$$

$$\sim \alpha : S | A = \lambda c. \sim (\alpha c)$$

$$\sim \alpha : A \rightsquigarrow S = \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)$$

- Negation is polymorphic
- indefinites in its scope are (optionally) existentially closed
- anaphora slots are passed through unchanged

Definition 2 (Negation)

$$\sim \alpha : s = 1 - \alpha$$

$$\sim \alpha : S | A = \lambda c. \sim (\alpha c)$$

$$\sim \alpha : A \rightsquigarrow S = \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)$$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

 $\neg \lambda x_{\text{CUP}'x}$.SEE'(MOVE'x)JOHN'

 $\neg \exists x (\text{CUP'}x \land \text{SEE'}(\text{MOVE'}x) \text{JOHN'})$

• Second option: existential closure at matrix level:

 $\lambda x_{\mathrm{CUP}'x}$. $\neg \mathrm{SEE'}(\mathrm{MOVE'}x)$ John'

 $\exists x (\text{CUP}'x \land \neg \text{SEE}'(\text{MOVE}'x) \text{JOHN}')$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

 $\neg \lambda x_{\text{CUP}'x}$.SEE'(MOVE'x)JOHN'

 $\neg \exists x (\text{CUP'}x \land \text{SEE'}(\text{MOVE'}x) \text{JOHN'})$

• Second option: existential closure at matrix level:

 $\lambda x_{\mathrm{CUP}'x}$. \neg SEE' (MOVE'x) JOHN'

 $\exists x (\text{CUP}'x \land \neg \text{SEE}'(\text{MOVE}'x) \text{JOHN}')$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

 $\neg \lambda x_{\text{CUP}}'_x$.SEE'(MOVE'x)JOHN'

 $\neg \exists x (\text{CUP'}x \land \text{SEE'}(\text{MOVE'}x) \text{JOHN'})$

• Second option: existential closure at matrix level:

 $\lambda x_{\mathrm{CUP}'x}$. $\neg \mathrm{SEE'}(\mathrm{MOVE'}x)$ John'

 $\exists x (\text{CUP}'x \land \neg \text{SEE}'(\text{MOVE}'x) \text{JOHN}')$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

 $\neg \lambda x_{\text{CUP}'x}$.SEE'(MOVE'x)JOHN'

 $\neg \exists x (\texttt{CUP'}x \land \texttt{SEE'}(\texttt{MOVE'}x) \texttt{JOHN'})$

• Second option: existential closure at matrix level:

 $\lambda x_{\text{CUP}'x}$. $\neg \text{SEE'}(\text{MOVE'}x)$ John'

 $\exists x (\text{CUP}'x \land \neg \text{SEE}'(\text{MOVE}'x) \text{JOHN}')$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

 $\neg \lambda x_{\text{CUP}'x}$.SEE'(MOVE'x)JOHN'

 $\neg \exists x (\texttt{CUP'}x \land \texttt{SEE'}(\texttt{MOVE'}x) \texttt{JOHN'})$

• Second option: existential closure at matrix level:

 $\lambda x_{\text{CUP}'x}.\neg\text{See'}(\text{move}'x)\text{John'}$

 $\exists x(\text{cup'}x \land \neg \text{see'}(\text{move'}x)\text{john'})$

- 5.1. Indefinites and scope
- (5) John didn't see a cup move
 - First option: existential closure by negation:

$$\neg \lambda x_{\text{CUP}'x}$$
.see'(move'x)john'

 $\neg \exists x (\texttt{CUP'}x \land \texttt{SEE'}(\texttt{MOVE'}x) \texttt{JOHN'})$

=

• Second option: existential closure at matrix level:

$$\lambda x_{\text{CUP}'x}$$
. $\neg \text{SEE'}(\text{MOVE'}x)$ JOHN'

 $\exists x (\texttt{CUP'}x \land \neg \texttt{SEE'}(\texttt{MOVE'}x) \texttt{JOHN'})$

=

- An indefinite can take scope over each clause it is contained in
- Indefinites scopally interact with operators like negation, but:
 - \circ No movement involved \leadsto not constrained by constraints on movement
 - o scoping mechanism is independent from quantifier scoping → not constrained by constraints on quantifier scope

- An indefinite can take scope over each clause it is contained in
- Indefinites scopally interact with operators like negation, but:
 - \circ No movement involved \leadsto not constrained by constraints on movement
 - o scoping mechanism is independent from quantifier scoping → not constrained by constraints on quantifier scope

- An indefinite can take scope over each clause it is contained in
- Indefinites scopally interact with operators like negation, but:
 - \circ No movement involved \leadsto not constrained by constraints on movement
 - o scoping mechanism is independent from quantifier scoping → not constrained by constraints on quantifier scope

- An indefinite can take scope over each clause it is contained in
- Indefinites scopally interact with operators like negation, but:
 - \circ No movement involved \leadsto not constrained by constraints on movement
 - o scoping mechanism is independent from quantifier scoping → not constrained by constraints on quantifier scope

- An indefinite can take scope over each clause it is contained in
- Indefinites scopally interact with operators like negation, but:
 - \circ No movement involved \leadsto not constrained by constraints on movement
 - o scoping mechanism is independent from quantifier scoping → not constrained by constraints on quantifier scope

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

$$Dom(f) \subseteq Dom(f \circ g)$$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

 $Dom(f)\subseteq Dom(f\circ g)$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

 $Dom(f) \subseteq Dom(f \circ g)$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

$$Dom(f)\subseteq Dom(f\circ g)$$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

$$Dom(f)\subseteq Dom(f\circ g)$$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- descriptive part is interpreted as domain restriction of partial function
- is inherited by superconstituents in semantic composition:

$$Dom(f)\subseteq Dom(f\circ g)$$

- Existential closure entails non-emptiness of domain
- Thus existential and descriptive scope are always identical

- "Donald Duck Problem" of naive long-distance existential closure analysis:
- (6) a. John will be offended if we invite a certain philosopher
 - b. $\simeq \exists x(\text{PHILO'}x \land (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$
 - c. $\neq \exists x (PHILO'x \land INVITE'xWE' \rightarrow OFFENDED'M')$
- "Bound Pronoun Problem" of choice function analysis
- (7) a. Every girl visited a boy she fancied
 - b. = $\forall x (\text{GIRL}'x \rightarrow \exists y (\text{BOY}'y \land \text{FANCY}'yx \land \text{VISIT}'yx))$
 - c. $\neq \exists f(ChF(f) \land \forall x(\text{GIRL'}x \rightarrow \text{(VISIT'}f(\lambda y.\text{BOY'}y \land \text{FANCY'}yx)x))$

●First ●Prev ●Next ●Last ●Go Back ●Full Screen ●Close ●Quit

- "Donald Duck Problem" of naive long-distance existential closure analysis:
- (6) a. John will be offended if we invite a certain philosopher
 - b. $\simeq \exists x(\text{PHILO'}x \land (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$
 - c. $\neq \exists x (PHILO'x \land INVITE'xWE' \rightarrow OFFENDED'M')$
- "Bound Pronoun Problem" of choice function analysis
- (7) a. Every girl visited a boy she fancied
 - b. = $\forall x (\text{GIRL}'x \rightarrow \exists y (\text{BOY}'y \land \text{FANCY}'yx \land \text{VISIT}'yx))$
 - c. $\neq \exists f(ChF(f) \land \forall x(\text{GIRL}'x \rightarrow \land \text{VISIT}'f(\lambda y.\text{BOY}'y \land \text{FANCY}'yx)x))$

● First ● Prev ● Next ● Last ● Go Back ● Full Screen ● Close ● Quit

- "Donald Duck Problem" of naive long-distance existential closure analysis:
- (6) a. John will be offended if we invite a certain philosopher
 - b. $\simeq \exists x(\text{PHILO'}x \land (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$
 - c. $\neq \exists x (PHILO'x \land INVITE'xWE' \rightarrow OFFENDED'M')$

• "Bound Pronoun Problem" of choice function analysis

- (7) a. Every girl visited a boy she fancied
 - **b**. = $\forall x (\text{GIRL}'x \rightarrow \exists y (\text{BOY}'y \land \text{FANCY}'yx \land \text{VISIT}'yx))$

c. $\neq \exists f(ChF(f) \land \forall x(\text{GIRL}'x \rightarrow \land \text{VISIT}'f(\lambda y.\text{BOY}'y \land \text{FANCY}'yx)x))$

● First ● Prev ● Next ● Last ● Go Back ● Full Screen ● Close ● Quit

- "Donald Duck Problem" of naive long-distance existential closure analysis:
- (6) a. John will be offended if we invite a certain philosopher
 - b. $\simeq \exists x(\text{PHILO'}x \land (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$
 - c. $\neq \exists x (PHILO'x \land INVITE'xWE' \rightarrow OFFENDED'M')$
- "Bound Pronoun Problem" of choice function analysis
- (7) a. Every girl visited a boy she fancied b. = $\forall x (\text{GIRL}'x \rightarrow \exists y (\text{BOY}'y \land \text{FANCY}'yx \land \text{VISIT}'yx))$

● First ● Prev ● Next ● Last ● Go Back ● Full Screen ● Close ● Quit

- "Donald Duck Problem" of naive long-distance existential closure analysis:
- (6) a. John will be offended if we invite a certain philosopher
 - b. $\simeq \exists x(\text{PHILO'}x \land (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$
 - c. $\neq \exists x (PHILO'x \land INVITE'xWE' \rightarrow OFFENDED'M')$
- "Bound Pronoun Problem" of choice function analysis
- (7) a. Every girl visited a boy she fancied
 - **b.** = $\forall x (\text{GIRL}'x \rightarrow \exists y (\text{BOY}'y \land \text{FANCY}'yx \land \text{VISIT}'yx))$

●First ●Prev ●Next ●Last ●Go Back ●Full Screen ●Close ●Quit

- (8) a. A cup moved, and Bill wonders which cupb. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
 - Semantics:
 - "missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

(8) a. A cup moved, and Bill wonders which cup

- b. A cup moved, and Bill wonders which cup moved
- Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
- Semantics:

"missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cupb. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
 - Semantics:
 - "missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cup
 - b. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
 - Semantics:
 - "missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cupb. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
 - Semantics:
 - "missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cup
 - b. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent

• Semantics:

"missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cup
 - b. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent

• Semantics:

"missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- (8) a. A cup moved, and Bill wonders which cup
 - b. A cup moved, and Bill wonders which cup moved
 - Syntax:
 - Sluicing involves a bare *wh*-phrase
 - needs a declarative clause containing an indefinite as antecedent
 - Semantics:
 - "missing" material is identical to antecedent except that indefinite is replaced by *wh*-trace

- Proposal: *which cup* has two types (but only one meaning):
- (9) a. $q/(s \uparrow np) : \lambda P.?x \text{CUP}'x \land Px$ b. $q|(np \rightsquigarrow s) : \lambda P.?x \text{CUP}'x \land Px$
- Antecedent clause has exactly the denotation that is needed to complete the elliptical question

- Proposal: which cup has two types (but only one meaning):
- (9) a. $q/(s \uparrow np) : \lambda P.?x \text{CUP}'x \land Px$ b. $q|(np \rightsquigarrow s) : \lambda P.?x \text{CUP}'x \land Px$
- Antecedent clause has exactly the denotation that is needed to complete the elliptical question

- Proposal: which cup has two types (but only one meaning):
- (9) a. $q/(s \uparrow np) : \lambda P.?x \text{CUP}'x \land Px$ b. $q|(np \rightsquigarrow s) : \lambda P.?x \text{CUP}'x \land Px$
- Antecedent clause has exactly the denotation that is needed to complete the elliptical question

Proposal: which cup has two types (but only one meaning):

(9) a.
$$q/(s \uparrow np) : \lambda P.?x \text{CUP}'x \land Px$$

b. $q|(np \rightsquigarrow s) : \lambda P.?x \text{CUP}'x \land Px$

• Antecedent clause has exactly the denotation that is needed to complete the elliptical question

Antecedent must contain an indefinite

(10) *The cup moved, and Bill wonders which cup

- First conjunct has category *s*
- which cup requires antecedent of category $np \rightsquigarrow s$
- -elimination not applicable

Antecedent must contain an indefinite

(10) *The cup moved, and Bill wonders which cup

- First conjunct has category *s*
- which cup requires antecedent of category $np \rightsquigarrow s$
- -elimination not applicable

Antecedent must contain an indefinite

(10) *The cup moved, and Bill wonders which cup

\bullet First conjunct has category s

- which cup requires antecedent of category $np \rightsquigarrow s$
- -elimination not applicable

Antecedent must contain an indefinite

(10) *The cup moved, and Bill wonders which cup

- First conjunct has category s
- \bullet which cup requires antecedent of category $np \rightsquigarrow s$
- -elimination not applicable

Antecedent must contain an indefinite

(10) *The cup moved, and Bill wonders which cup

- First conjunct has category *s*
- \bullet which cup requires antecedent of category $np \rightsquigarrow s$
- |-elimination not applicable

- No transformational connection to non-elliptical counterpart
- No restrictions on scope of indefinites ⇒ no restrictions on embedding depth of antecedent indefinites in Sluicing
- (11) a. The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one
 - b. *The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one the administration has issued a statement that it is willing to meet with from Chung, Ladusaw and McCloskey 1995

- No transformational connection to non-elliptical counterpart
- No restrictions on scope of indefinites ⇒ no restrictions on embedding depth of antecedent indefinites in Sluicing
- (11) a. The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one
 - b. *The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one the administration has issued a statement that it is willing to meet with from Chung, Ladusaw and McCloskey 1995

- No transformational connection to non-elliptical counterpart
- No restrictions on scope of indefinites \Rightarrow no restrictions on embedding depth of antecedent indefinites in Sluicing
- (11) a. The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one
 - b. *The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one the administration has issued a statement that it is willing to meet with from Chung, Ladusaw and McCloskey 1995

- No transformational connection to non-elliptical counterpart
- No restrictions on scope of indefinites ⇒ no restrictions on embedding depth of antecedent indefinites in Sluicing
- (11) a. The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one
 - b. *The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one the administration has issued a statement that it is willing to meet with from Chung, Ladusaw and McCloskey 1995

- No transformational connection to non-elliptical counterpart
- No restrictions on scope of indefinites ⇒ no restrictions on embedding depth of antecedent indefinites in Sluicing
- (11) a. The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one
 - b. *The administration has issued a statement that it is willing to meet with one of the student groups, but I'm not sure which one the administration has issued a statement that it is willing to meet with from Chung, Ladusaw and McCloskey 1995

- (12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
 HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT {WHO_{DAT} / *WHO_{ACC}}
 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - \bullet clause containing dative indefinite: $np(dat) \rightsquigarrow s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)|$

- (12) Er will jemandem schmeicheln, aber sie wissen nicht $\{wem / *wen\}$ HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT $\{WHO_{DAT} / *WHO_{ACC}\}$ 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - clause containing dative indefinite: $np(dat) \rightsquigarrow s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)|$

- (12) Er will jemandem schmeicheln, aber sie wissen nicht $\{wem / *wen\}$ HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT $\{WHO_{DAT} / *WHO_{ACC}\}$ 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - \bullet clause containing dative indefinite: $np(dat) \rightsquigarrow s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)|$

- (12) Er will jemandem schmeicheln, aber sie wissen nicht $\{wem / *wen\}$ HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT $\{WHO_{DAT} / *WHO_{ACC}\}$ 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - \bullet clause containing dative indefinite: $np(dat) \rightsquigarrow s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)|$

- (12) Er will jemandem schmeicheln, aber sie wissen nicht $\{wem / *wen\}$ HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT $\{WHO_{DAT} / *WHO_{ACC}\}$ 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - \bullet clause containing dative indefinite: $np(dat) \leadsto s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)|$

- (12) Er will jemandem schmeicheln, aber sie wissen nicht $\{wem / *wen\}$ HE WANTS SOMEONE_{DAT} FLATTER BUT THEY KNOW NOT $\{WHO_{DAT} / *WHO_{ACC}\}$ 'He wants to flatter someone, but they don't know whom'
 - morphological information coded in syntactic category
 - \bullet indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
 - \bullet clause containing dative indefinite: $np(dat) \leadsto s$
 - Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)$

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing

- Indefinites and pronouns are interpreted as (partial) identity functions
- Pronoun binding via syntactic deduction
- existential impact of indefinites is buried in truth definition/semantics of negation etc.
- descriptive content of indefinites is interpreted as domain restriction
- empirical coverage: specificity and sluicing