Contents

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

1. Outline of talk

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

1. Outline of talk

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

1. Outline of talk

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

1. Outline of talk

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

1. Outline of talk

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

2. Anaphora in TLG

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. Anaphora in TLG

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. Anaphora in TLG

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. Anaphora in TLG

2.1. Jacobson's proposal

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

2.2. Adaptation to TLG

- Natural Deduction rules for anaphora slash

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

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

2.2. Adaptation to TLG

- Natural Deduction rules for anaphora slash

$$[M : A]_i \quad \cdots \quad \frac{N : B|A}{[NM : B]_i} |E, i$$

$$\frac{\begin{array}{ccc} \vdots & \dfrac{M : A|B}{Mx : A}\, i & \vdots \\ \vdots & \vdots & \vdots \end{array}}{\dfrac{N : C}{\lambda x N : C|B} |I, i}$$

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

2.2. Adaptation to TLG

- Natural Deduction rules for anaphora slash

$$[M : A]_i \quad \cdots \quad \frac{N : B|A}{[NM : B]_i}\, |E, i$$

$$\begin{array}{ccc} \vdots & \dfrac{M : A|B}{Mx : A}\, i & \vdots \\ \vdots & \vdots & \vdots \\ \hline & \dfrac{N : C}{\lambda x N : C|B}\, |I, i & \end{array}$$

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

2.2. Adaptation to TLG

- Natural Deduction rules for anaphora slash

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

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

2.2. Adaptation to TLG

- Natural Deduction rules for anaphora slash

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

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

2.3. Binding

(1) John said he walked

$$\dfrac{\dfrac{John}{[\text{J'} : np]_i}\,lex \qquad \dfrac{\dfrac{said}{\text{SAY'} : np\backslash s/s}\,lex \qquad \dfrac{\dfrac{\dfrac{he}{[\lambda x.x : np|np]_i}\,lex}{\text{J'} : np}\,|E \qquad \dfrac{walked}{\text{WALK'} : np\backslash s}\,lex}{\text{WALK' J'} : s}\,\backslash E}{\text{SAY'(WALK' J')} : np\backslash s}\,/E}{\text{SAY'(WALK' J')J'} : s}\,\backslash E$$

2.3. Binding

(1) John said he walked

$$\dfrac{\dfrac{John}{[\text{J'} : np]_i}\,lex \qquad \dfrac{\dfrac{said}{\text{SAY'} : np\backslash s/s}\,lex \qquad \dfrac{\dfrac{\dfrac{he}{[\lambda x.x : np|np]_i}\,lex}{\text{J'} : np}\,|E \qquad \dfrac{walked}{\text{WALK'} : np\backslash s}\,lex}{\text{WALK' J'} : s}\,\backslash E}{\text{SAY'(WALK' J')} : np\backslash s}\,/E}{\text{SAY'(WALK' J')J'} : s}\,\backslash E$$

2.3. Binding

(1) John said he walked

$$\dfrac{\dfrac{\textit{John}}{[\text{J'} : np]_i}\,lex \qquad \dfrac{\dfrac{\textit{said}}{\text{SAY'} : np\backslash s/s}\,lex \qquad \dfrac{\dfrac{\dfrac{\textit{he}}{[\lambda x.x : np|np]_i}\,lex}{\text{J'} : np}\,|E \qquad \dfrac{\textit{walked}}{\text{WALK'} : np\backslash s}\,lex}{\text{WALK' J'} : s}\,\backslash E}{\text{SAY'(WALK' J')} : np\backslash s}\,/E}{\text{SAY'(WALK' J')J'} : s}\,\backslash E$$

2.4. Percolation

$$\dfrac{\dfrac{he}{\lambda x.x : np|np}\,lex}{y : np}\,1 \qquad \dfrac{walked}{\text{WALK'} : np\backslash s}\,lex$$

$$\dfrac{y : np \qquad \text{WALK'} : np\backslash s}{\text{WALK'}y : s}\,\backslash E$$

$$\dfrac{said}{\text{SAY'} : np\backslash s/s}\,lex \qquad \dfrac{\text{SAY'} : np\backslash s/s \qquad \text{WALK'}y : s}{\text{SAY'}(\text{WALK'}y) : np\backslash s}\,/E$$

$$\dfrac{John}{\text{J'} : np}\,lex \qquad \dfrac{\text{J'} : np \qquad \text{SAY'}(\text{WALK'}y) : np\backslash s}{\text{SAY'}(\text{WALK'}y)\text{J'} : s}\,\backslash E$$

$$\dfrac{\text{SAY'}(\text{WALK'}y)\text{J'} : s}{\lambda y.\text{SAY'}(\text{WALK'}y)\text{J'} : s|np}\,|I, 1$$

2.4. Percolation

$$
\dfrac{\dfrac{\dfrac{\mathit{John}}{\text{J'} : np}\,lex \qquad \dfrac{\dfrac{\mathit{said}}{\text{SAY'} : np\backslash s/s}\,lex \qquad \dfrac{\dfrac{\dfrac{\mathit{he}}{\lambda x.x : np|np}\,lex}{y : np}\,1 \qquad \dfrac{\mathit{walked}}{\text{WALK'} : np\backslash s}\,lex}{\text{WALK'}y : s}\,\backslash E}{\text{SAY'}(\text{WALK'}y) : np\backslash s}\,/E}{\text{SAY'}(\text{WALK'}y)\text{J'} : s}\,\backslash E}{\lambda y.\text{SAY'}(\text{WALK'}y)\text{J'} : s|np}\,|I, 1
$$

3. Covering indefinites

3.1. Basic idea

(2) a. It moved

b. Something moved

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

3. Covering indefinites

3.1. Basic idea

(2) a. It moved
    b. Something moved

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

3. Covering indefinites

3.1. Basic idea

(2) a. It moved
    b. Something moved

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

3. Covering indefinites

3.1. Basic idea

(2) a. It moved

b. Something moved

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

3.2. Type Logical implementation

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

3.2. Type Logical implementation

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

3.2. Type Logical implementation

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

3.2. Type Logical implementation

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

3.2. Type Logical implementation

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

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

$$\dfrac{\dfrac{\begin{matrix} \vdots & \dfrac{M : A \rightsquigarrow B}{Mx : B}\,i & \vdots \\ \vdots & \vdots & \vdots \end{matrix}}{N : C}}{\lambda x N : A \rightsquigarrow C}\rightsquigarrow, i$$

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

$$\dfrac{\dfrac{\begin{array}{ccc} \vdots & \dfrac{M : A \leadsto B}{Mx : B}\,i & \vdots \\ \vdots & \vdots & \vdots \end{array}}{N : C}}{\lambda x N : A \leadsto C}\,\leadsto, i$$

(3) a. John saw something

b.
$$
\dfrac{
  \dfrac{\textit{John}}{\begin{array}{c}\text{JOHN'}\\ np\end{array}}\,lex
  \qquad
  \dfrac{
    \dfrac{\textit{saw}}{\begin{array}{c}\text{SEE'}\\ (np\backslash s)/np\end{array}}\,lex
    \quad
    \dfrac{\dfrac{\textit{something}}{\begin{array}{c}\lambda x x\\ np \leadsto np\end{array}}\,lex}{\begin{array}{c}y\\ np\end{array}}\,i
  }{\begin{array}{c}\text{SEE'}y\\ np\backslash s\end{array}}\,/E
}{
  \dfrac{\begin{array}{c}\text{SEE'}y\text{JOHN'}\\ s\end{array}}{\begin{array}{c}\lambda y.\text{SEE'}y\text{JOHN'}\\ np \leadsto s\end{array}}\,\leadsto, i
}\,\backslash E
$$

(3) a. John saw something

b.

$$\dfrac{\dfrac{John}{\begin{matrix}\text{JOHN'}\\ np\end{matrix}}\,lex \quad \dfrac{\dfrac{saw}{\begin{matrix}\text{SEE'}\\ (np\backslash s)/np\end{matrix}}\,lex \quad \dfrac{\dfrac{something}{\begin{matrix}\lambda x x\\ np \leadsto np\end{matrix}}\,lex}{\begin{matrix}y\\ np\end{matrix}}\,i}{\begin{matrix}\text{SEE'}y\\ np\backslash s\end{matrix}}\,/E}{\dfrac{\begin{matrix}\text{SEE'}y\text{JOHN'}\\ s\end{matrix}}{\begin{matrix}\lambda y.\text{SEE'}y\text{JOHN'}\\ np \leadsto s\end{matrix}}\,\leadsto, i}\,\backslash E$$

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
  - $f(x) = 0$ iff x is a cup that did not move
  - $f(x)$ 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:
  - $f(x) = 1$ 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

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
  - $f(x) = 0$ iff x is a cup that did not move
  - $f(x)$ 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:
  - $f(x) = 1$ 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

3.3. Descriptive content

- Idea: descriptive content expresses domain restriction
- $\|\text{a}\|$ = function that maps a property to the identity function over its extension
- $\|\text{a cup}\|$ = identity function on the set of cups
- $\|\text{a cup moved}\|$ = 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

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
  - $f(x) = 0$ iff x is a cup that did not move
  - $f(x)$ 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:
  - $f(x) = 1$ 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

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

4. Variable free existential closure

- 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

4. Variable free existential closure

- 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

4. Variable free existential closure

- 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

4. Variable free existential closure

- 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)

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

- Truth is relativized to sequence of referents and syntactic category

Definition 1 (Truth)

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

(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

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

Negation

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

Definition 2 (Negation)

$$\begin{aligned} \sim \alpha : s &= 1 - \alpha \\ \sim \alpha : S|A &= \lambda c. \sim (\alpha c) \\ \sim \alpha : A \rightsquigarrow S &= \sim (\bigcup_{c \in Dom(\alpha)} \alpha c) \end{aligned}$$

Negation

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

Definition 2 (Negation)

$$\begin{aligned} \sim \alpha : s &= 1 - \alpha \\ \sim \alpha : S|A &= \lambda c. \sim (\alpha c) \\ \sim \alpha : A \rightsquigarrow S &= \sim (\bigcup_{c \in Dom(\alpha)} \alpha c) \end{aligned}$$

Negation

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

Definition 2 (Negation)

$$
\begin{aligned}
\sim \alpha : s &= 1 - \alpha \\
\sim \alpha : S|A &= \lambda c. \sim (\alpha c) \\
\sim \alpha : A \rightsquigarrow S &= \sim \Big(\bigcup_{c \in Dom(\alpha)} \alpha c \Big)
\end{aligned}
$$

Negation

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

Definition 2 (Negation)

$$\begin{aligned} \sim\alpha : s &= 1 - \alpha \\ \sim\alpha : S|A &= \lambda c. \sim(\alpha c) \\ \sim\alpha : A \rightsquigarrow S &= \sim\Big(\bigcup_{c \in Dom(\alpha)} \alpha c\Big) \end{aligned}$$

Negation

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

Definition 2 (Negation)

$$
\begin{aligned}
\sim \alpha : s &= 1 - \alpha \\
\sim \alpha : S|A &= \lambda c. \sim (\alpha c) \\
\sim \alpha : A \rightsquigarrow S &= \sim (\bigcup_{c \in Dom(\alpha)} \alpha c)
\end{aligned}
$$

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

- Second option: existential closure at matrix level:

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

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

- Second option: existential closure at matrix level:

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

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

$$\equiv$$

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

- Second option: existential closure at matrix level:

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

$$\equiv$$

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

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

- Second option: existential closure at matrix level:

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

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

$$\equiv$$

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

- Second option: existential closure at matrix level:

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

$$\equiv$$

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

5. Linguistic consequences

5.1. Indefinites and scope

(5) John didn't see a cup move

- First option: existential closure by negation:

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

- Second option: existential closure at matrix level:

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

Properties of the analysis

No island effects

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

Properties of the analysis

No island effects

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

Properties of the analysis

No island effects

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

Properties of the analysis

No island effects

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

Properties of the analysis

No island effects

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

No split between existential force and descriptive content

- 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

No split between existential force and descriptive content

- 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

No split between existential force and descriptive content

- 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

No split between existential force and descriptive content

- 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

No split between existential force and descriptive content

- 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

No split between existential force and descriptive content

- 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

Avoids

- "Donald Duck Problem" of naive long-distance existential closure analysis:

(6) a. John will be offended if we invite a certain philosopher

b. $\simeq \exists x(\text{PHILO'}x \wedge (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$

c. $\neq \exists x(\text{PHILO'}x \wedge \text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'})$

- "Bound Pronoun Problem" of choice function analysis

(7) a. Every girl visited a boy she fancied

b. $= \forall x(\text{GIRL'}x \rightarrow \exists y(\text{BOY'}y \wedge \text{FANCY'}yx \wedge \text{VISIT'}yx))$

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

Avoids

- "Donald Duck Problem" of naive long-distance existential closure analysis:

(6) a. John will be offended if we invite a certain philosopher

b. $\simeq \exists x(\text{PHILO'}x \wedge (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$

c. $\neq \exists x(\text{PHILO'}x \wedge \text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'})$

- "Bound Pronoun Problem" of choice function analysis

(7) a. Every girl visited a boy she fancied

b. $= \forall x(\text{GIRL'}x \rightarrow \exists y(\text{BOY'}y \wedge \text{FANCY'}yx \wedge \text{VISIT'}yx))$

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

Avoids

- "Donald Duck Problem" of naive long-distance existential closure analysis:

(6) a. John will be offended if we invite a certain philosopher

b. $\simeq \exists x(\text{PHILO'}x \wedge (\text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'}))$

c. $\neq \exists x(\text{PHILO'}x \wedge \text{INVITE'}x\text{WE'} \rightarrow \text{OFFENDED'M'})$

- "Bound Pronoun Problem" of choice function analysis

(7) a. Every girl visited a boy she fancied

b. $= \forall x(\text{GIRL'}x \rightarrow \exists y(\text{BOY'}y \wedge \text{FANCY'}yx \wedge \text{VISIT'}yx))$

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

Avoids

- "Donald Duck Problem" of naive long-distance existential closure analysis:

(6) a. John will be offended if we invite a certain philosopher

b. $\simeq \quad \exists x(\text{PHILO'}x \ \wedge \ (\text{INVITE'}x\text{WE'} \ \rightarrow \ \text{OFFENDED'M'}))$

c. $\neq \quad \exists x(\text{PHILO'}x \ \wedge \ \text{INVITE'}x\text{WE'} \ \rightarrow \ \text{OFFENDED'M'})$

- "Bound Pronoun Problem" of choice function analysis

(7) a. Every girl visited a boy she fancied

b. $= \forall x(\text{GIRL'}x \rightarrow \exists y(\text{BOY'}y \wedge \text{FANCY'}yx \wedge \text{VISIT'}yx))$

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

Avoids

- "Donald Duck Problem" of naive long-distance existential closure analysis:

(6) a. John will be offended if we invite a certain philosopher
b. $\simeq \exists x(\textsc{philo}'x \wedge (\textsc{invite}'x\textsc{we}' \rightarrow \textsc{offended}'\textsc{m}'))$
c. $\neq \exists x(\textsc{philo}'x \wedge \textsc{invite}'x\textsc{we}' \rightarrow \textsc{offended}'\textsc{m}')$

- "Bound Pronoun Problem" of choice function analysis

(7) a. Every girl visited a boy she fancied
b. $= \forall x(\textsc{girl}'x \rightarrow \exists y(\textsc{boy}'y \wedge \textsc{fancy}'yx \wedge \textsc{visit}'yx))$
c. $\neq \exists f(ChF(f) \wedge \forall x(\textsc{girl}'x \rightarrow \wedge\textsc{visit}'f(\lambda y.\textsc{boy}'y \wedge \textsc{fancy}'yx)x))$

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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

6. Sluicing

(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. $q/(s \uparrow np) : \lambda P.?x\textsc{cup}'x \wedge Px$

b. $q|(np \rightsquigarrow s) : \lambda P.?x\textsc{cup}'x \wedge Px$

- 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. $q/(s \uparrow np) : \lambda P.?x\textsc{cup}'x \wedge Px$
b. $q|(np \rightsquigarrow s) : \lambda P.?x\textsc{cup}'x \wedge Px$

- 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. $q/(s \uparrow np) : \lambda P.?x\textsc{cup}'x \wedge Px$
b. $q|(np \rightsquigarrow s) : \lambda P.?x\textsc{cup}'x \wedge Px$

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

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

6.1. Predictions

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

6.1. Predictions

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

6.1. Predictions

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

6.1. Predictions

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

6.1. Predictions

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

Sluicing is island insensitive

- 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

Sluicing is island insensitive

- 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

Sluicing is island insensitive

- 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

Sluicing is island insensitive

- 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

Sluicing is island insensitive

- 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

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\text{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\text{DAT}}$ / *WHO$_{\text{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
- clause containing dative indefinite: $np(dat) \rightsquigarrow s$
- Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)$

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\text{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\text{DAT}}$ / *WHO$_{\text{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
- clause containing dative indefinite: $np(dat) \rightsquigarrow s$
- Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)$

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\mathrm{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\mathrm{DAT}}$ / *WHO$_{\mathrm{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \leadsto np(dat)$
- clause containing dative indefinite: $np(dat) \leadsto s$
- Sluicing remnant in dative: $q|(np(dat) \leadsto s)$

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\text{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\text{DAT}}$ / *WHO$_{\text{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
- clause containing dative indefinite: $np(dat) \rightsquigarrow s$
- Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)$

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\mathrm{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\mathrm{DAT}}$ / *WHO$_{\mathrm{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \leadsto np(dat)$
- clause containing dative indefinite: $np(dat) \leadsto s$
- Sluicing remnant in dative: $q|(np(dat) \leadsto s)$

Morphological sensitivity

(12) Er will jemandem schmeicheln, aber sie wissen nicht {wem / *wen}
HE WANTS SOMEONE$_{\text{DAT}}$ FLATTER BUT THEY KNOW NOT {WHO$_{\text{DAT}}$ / *WHO$_{\text{ACC}}$}
'He wants to flatter someone, but they don't know whom'

- morphological information coded in syntactic category
- indefinite NP in dative has category $np(dat) \rightsquigarrow np(dat)$
- clause containing dative indefinite: $np(dat) \rightsquigarrow s$
- Sluicing remnant in dative: $q|(np(dat) \rightsquigarrow s)$

7. Conclusion

- 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

7. Conclusion

- 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

7. Conclusion

- 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

7. Conclusion

- 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

7. Conclusion

- 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

7. Conclusion

- 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