Semantics & Discourse

— Noun Phrases and Quantified Noun Phrases —

Philippe de Groote

Outline

Direct naive interpretation

Abstract language

$$\begin{aligned} \text{ERIC} &: \text{NP} \\ \text{REBECCA} &: \text{NP} \\ \text{LOVES} &: \text{NP} \to \text{NP} \to \text{S} \end{aligned}$$

Direct naive interpretation

Abstract language

$$\begin{aligned} \textsc{eric} &: \text{NP} \\ \textsc{rebecca} &: \text{NP} \\ \textsc{loves} &: \text{NP} \to \text{NP} \to \text{S} \end{aligned}$$

Semantic interpretation

$$\begin{aligned} \text{NP} &:= \text{e} \\ \text{S} &:= \text{t} \end{aligned}$$

$$\begin{aligned} \textsc{eric} &:= \mathbf{e} \\ \textsc{rebecca} &:= \mathbf{r} \\ \textsc{loves} &:= \lambda y.\, \lambda x.\, \mathbf{love}\, x\, y \end{aligned}$$

where:

$\mathbf{e}, \mathbf{r} : \text{e}$

$\mathbf{love} : \text{e} \to \text{e} \to \text{t}$

Direct naive interpretation

- *Rebecca loves Eric*
- *Somebody loves Eric*
- *Everybody loves Eric*
- *Nobody loves Eric*

Type raising

Abstract language

$$\text{ERIC} : \text{NP}$$
$$\text{REBECCA} : \text{NP}$$
$$\text{EVERYBODY} : \text{NP}$$
$$\text{SOMEBODY} : \text{NP}$$
$$\text{LOVES} : \text{NP} \rightarrow \text{NP} \rightarrow \text{S}$$

Type raising

Abstract language

$$\begin{aligned} \text{ERIC} &: \text{NP} \\ \text{REBECCA} &: \text{NP} \\ \text{EVERYBODY} &: \text{NP} \\ \text{SOMEBODY} &: \text{NP} \\ \text{LOVES} &: \text{NP} \to \text{NP} \to \text{S} \end{aligned}$$

Semantic interpretation

$$\begin{aligned} \text{NP} &:= (\mathrm{e} \to \mathrm{t}) \to \mathrm{t} \\ \text{S} &:= \mathrm{t} \end{aligned}$$

$$\begin{aligned} \text{ERIC} &:= \lambda k.\, k\, \mathbf{e} \\ \text{REBECCA} &:= \lambda k.\, k\, \mathbf{r} \\ \text{EVERYBODY} &:= \lambda k.\, \forall x.\, (\mathbf{human}\, x) \to (k\, x) \\ \text{SOMEBODY} &:= \lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x) \\ \text{LOVES} &:= \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{love}\, x\, y)) \end{aligned}$$

Noun and determiners

Abstract language

$$
\begin{aligned}
\text{ERIC} &: \text{NP} \\
\text{REBECCA} &: \text{NP} \\
\text{EVERYBODY} &: \text{NP} \\
\text{SOMEBODY} &: \text{NP} \\
\text{MAN} &: \text{N} \\
\text{WOMAN} &: \text{N} \\
\text{EVERY} &: \text{N} \rightarrow \text{NP} \\
\text{A} &: \text{N} \rightarrow \text{NP} \\
\text{LOVES} &: \text{NP} \rightarrow \text{NP} \rightarrow \text{S}
\end{aligned}
$$

Semantic interpretation

$$\begin{aligned}
\mathrm{N} &:= \mathrm{e} \to \mathrm{t} \\
\mathrm{NP} &:= (\mathrm{e} \to \mathrm{t}) \to \mathrm{t} \\
\mathrm{S} &:= \mathrm{t}
\end{aligned}$$

$$\begin{aligned}
\mathrm{ERIC} &:= \lambda k.\, k\, \mathbf{e} \\
\mathrm{REBECCA} &:= \lambda k.\, k\, \mathbf{r} \\
\mathrm{EVERYBODY} &:= \lambda k.\, \forall x.\, (\mathbf{human}\, x) \to (k\, x) \\
\mathrm{SOMEBODY} &:= \lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x) \\
\mathrm{MAN} &:= \lambda x.\, \mathbf{man}\, x \\
\mathrm{WOMAN} &:= \lambda x.\, \mathbf{woman}\, x \\
\mathrm{EVERY} &:= \lambda n.\, \lambda m.\, \forall x.\, (n\, x) \to (m\, x) \\
\mathrm{A} &:= \lambda n.\, \lambda m.\, \exists x.\, (n\, x) \wedge (m\, x) \\
\mathrm{LOVES} &:= \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{love}\, x\, y))
\end{aligned}$$

where:

$$\mathbf{woman}, \mathbf{man} : \mathrm{e} \to \mathrm{t}$$

Generalized quantifiers

Generalized quantifiers

- $(e \to t) \to t$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).

Generalized quantifiers

Generalized quantifiers

- $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is called a generalized quantifier.

Generalized quantifiers

Generalized quantifiers

- $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.

Generalized quantifiers

Generalized quantifiers

- $(\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.
- Interpretation of *everybody*:

Generalized quantifiers

Generalized quantifiers

- $(\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.
- Interpretation of *everybody*:
  - $\lambda p. \forall x. (\mathbf{human}\, x) \to (p\, x)$

Generalized quantifiers

Generalized quantifiers

- $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.
- Interpretation of *everybody*:
  - $\lambda p. \forall x. (\mathbf{human}\, x) \rightarrow (p\, x)$
  - $\{P \in 2^{\mathrm{E}} \mid \mathbf{human} \subset P\}$

Generalized quantifiers

Determiners as binary generalized quantifiers

- The type of the interpretations of EVERY and SOME is $(\mathrm{e} \to \mathrm{t}) \to (\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$.

Generalized quantifiers

Determiners as binary generalized quantifiers

- The type of the interpretations of EVERY and SOME is $(\mathrm{e} \to \mathrm{t}) \to (\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$.
- Every term of type $(\mathrm{e} \to \mathrm{t}) \to (\mathrm{e} \to \mathrm{t}) \to \mathrm{t}$ is called a binary generalized quantifier.

Generalized quantifiers

Determiners as binary generalized quantifiers

- The type of the interpretations of EVERY and SOME is $(e \to t) \to (e \to t) \to t$.
- Every term of type $(e \to t) \to (e \to t) \to t$ is called a binary generalized quantifier.
- Semantically, a binary generalized quantifier corresponds to a relation between two sets of entities.

Generalized quantifiers

Determiners as binary generalized quantifiers

SOME $A\,B$	$A \cap B \neq \varnothing$	$\uparrow\uparrow$
EVERY $A\,B$	$A \subset B$	$\downarrow\uparrow$
NO $A\,B$	$A \cap B = \varnothing$	$\downarrow\downarrow$
(AT-LEAST n) $A\,B$	$\lvert A \cap B\rvert \geq n$	$\uparrow\uparrow$
(AT-MOST n) $A\,B$	$\lvert A \cap B\rvert \leq n$	$\downarrow\downarrow$
(EXACTLY n) $A\,B$	$\lvert A \cap B\rvert = n$	$-\,-$
MOST $A\,B$	$\lvert A\rvert \leq 2\,\lvert A \cap B\rvert$	$-\uparrow$

Scope ambiguities

Subject vs object wide scope

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Two possible interpretations:

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Two possible interpretations:

- $\forall x.\mathbf{man}\,x \rightarrow (\exists y.\mathbf{woman}\,y \wedge \mathbf{love}\,x\,y)$

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Two possible interpretations:

- $\forall x.\mathbf{man}\,x \rightarrow (\exists y.\mathbf{woman}\,y \wedge \mathbf{love}\,x\,y)$
- $\exists y.\mathbf{woman}\,y \wedge (\forall x.\mathbf{man}\,x \wedge \mathbf{love}\,x\,y)$

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Two possible interpretations:

- $\forall x.\mathbf{man}\, x \to (\exists y.\mathbf{woman}\, y \wedge \mathbf{love}\, x\, y)$
- $\exists y.\mathbf{woman}\, y \wedge (\forall x.\mathbf{man}\, x \wedge \mathbf{love}\, x\, y)$

Subject wide scope:

- $\text{LOVES} := \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{love}\, x\, y))$

Scope ambiguities

Subject vs object wide scope

Every man loves a woman.

Two possible interpretations:

- $\forall x.\mathbf{man}\, x \rightarrow (\exists y.\mathbf{woman}\, y \wedge \mathbf{love}\, x\, y)$
- $\exists y.\mathbf{woman}\, y \wedge (\forall x.\mathbf{man}\, x \wedge \mathbf{love}\, x\, y)$

Subject wide scope:

- $\text{LOVES} := \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{love}\, x\, y))$

Object wide scope:

- $\text{LOVES}_{\text{ows}} := \lambda o.\, \lambda s.\, o\, (\lambda y.\, s\, (\lambda x.\, \mathbf{love}\, x\, y))$

Scope ambiguities

Quantifier raising

Scope ambiguities

Quantifier raising

Another solution:

Scope ambiguities

Quantifier raising

Another solution:

- QR : $\mathrm{NP} \rightarrow (\mathrm{NP} \rightarrow \mathrm{S}) \rightarrow \mathrm{S}$

Scope ambiguities

Quantifier raising

Another solution:

- $\mathrm{QR} \;:\; \mathrm{NP} \rightarrow (\mathrm{NP} \rightarrow \mathrm{S}) \rightarrow \mathrm{S}$
- $\mathrm{QR} \;:=\; \lambda n.\,\lambda p.\,n\,(\lambda x.\,p\,(\lambda k.\,k\,x))$

Scope ambiguities

Quantifier raising

Another solution:

- $\textsc{qr} \; : \; \mathrm{NP} \rightarrow (\mathrm{NP} \rightarrow \mathrm{S}) \rightarrow \mathrm{S}$
- $\textsc{qr} \; := \; \lambda n.\, \lambda p.\, n\, (\lambda x.\, p\, (\lambda k.\, k\, x))$

Wide scope reading:

- $\textsc{qr}\, (\textsc{a woman})\, (\lambda o.\, \textsc{love}\, o\, (\textsc{every man}))$

De re and de dicto readings as scope ambiguities

De re and de dicto readings

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

Two possible interpretations:

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

Two possible interpretations:

- $\exists x.\mathbf{unicorn}\, x \wedge \mathbf{try}\, \mathbf{j}\, (\lambda z.\, \mathbf{find}\, z\, x)$

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

Two possible interpretations:

- $\exists x.\mathbf{unicorn}\, x \wedge \mathbf{try}\,\mathbf{j}\,(\lambda z.\,\mathbf{find}\, z\, x)$
- $\mathbf{try}\,\mathbf{j}\,(\lambda z.\,\exists x.\mathbf{unicorn}\, x \wedge \mathbf{find}\, z\, x)$

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

Two possible interpretations:

- $\exists x.\mathbf{unicorn}\,x \wedge \mathbf{try}\,\mathbf{j}\,(\lambda z.\,\mathbf{find}\,z\,x)$
- $\mathbf{try}\,\mathbf{j}\,(\lambda z.\,\exists x.\mathbf{unicorn}\,x \wedge \mathbf{find}\,z\,x)$

De re reading:

- $[\![\mathrm{SEEK}_{\mathrm{re}}]\!] \;=\; \lambda o.\,\lambda s.\,s\,(\lambda x.\,o\,(\lambda y.\,\mathbf{try}\,x\,(\lambda z.\,\mathbf{find}\,z\,y)))$

De re and de dicto readings as scope ambiguities

De re and de dicto readings

John seeks a unicorn.

Two possible interpretations:

- $\exists x.\mathbf{unicorn}\,x \land \mathbf{try}\,\mathbf{j}\,(\lambda z.\,\mathbf{find}\,z\,x)$
- $\mathbf{try}\,\mathbf{j}\,(\lambda z.\,\exists x.\mathbf{unicorn}\,x \land \mathbf{find}\,z\,x)$

De re reading:

- $[\![\mathrm{SEEK}_{\mathrm{re}}]\!] \;=\; \lambda o.\,\lambda s.\,s\,(\lambda x.\,o\,(\lambda y.\,\mathbf{try}\,x\,(\lambda z.\,\mathbf{find}\,z\,y)))$

De dicto reading:

- $[\![\mathrm{SEEK}_{\mathrm{dicto}}]\!] \;=\; \lambda o.\,\lambda s.\,s\,(\lambda x.\,\mathbf{try}\,x\,(\lambda z.\,o\,(\lambda y.\,\mathbf{find}\,z\,y)))$