Formal Semantics of Natural Language

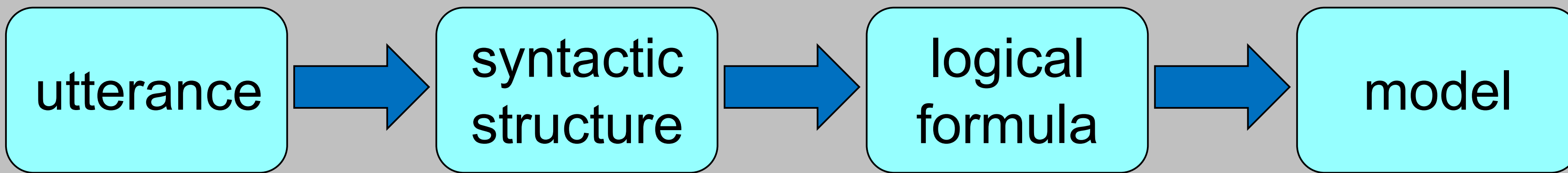
Philippe de Groote

Quantification

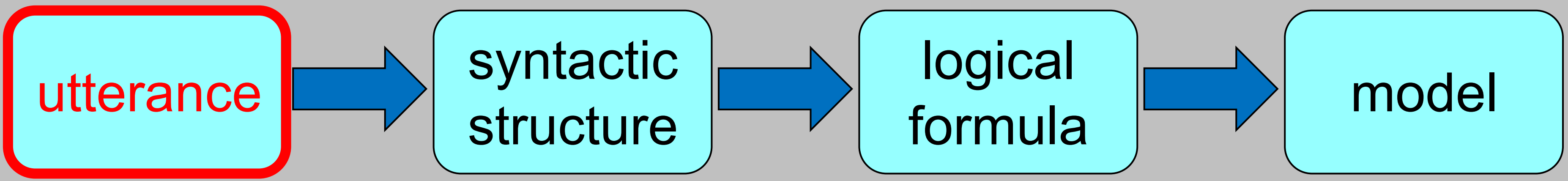
Quantificational expressions

- *Everybody* praised Mary.
- *Everybody but Tina* praised Mary.
- One can find it *everywhere*.
- John *rarely* wears a cap.
- John *most often* wears a cap.
- We are *far from* Beijing.
- There is *a lot of* work to do today.
- *Everybody* needs *some* help *sometimes*.
- *Some* representatives of *every* department in *most* companies saw *a few* samples of *every* product

Taking stock

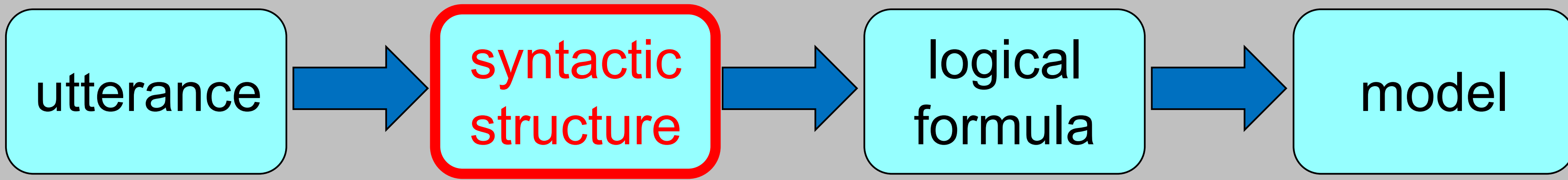


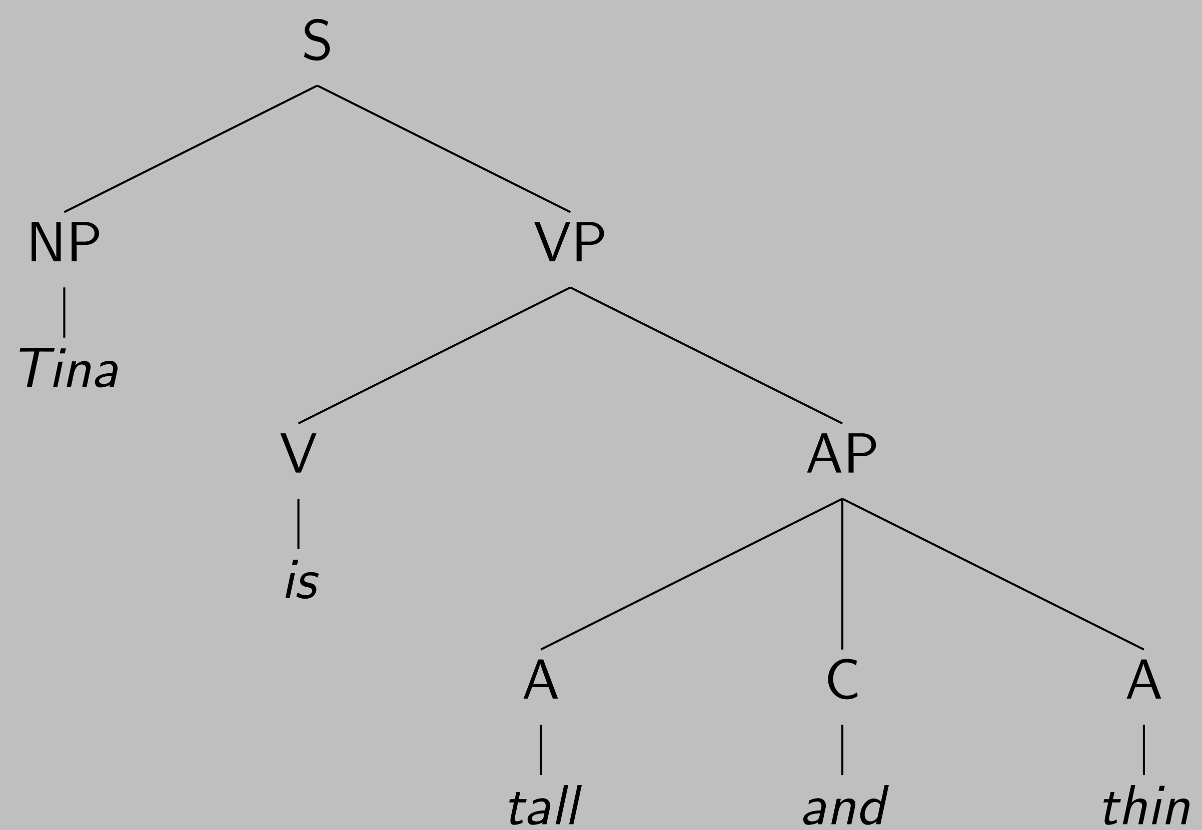
Taking stock



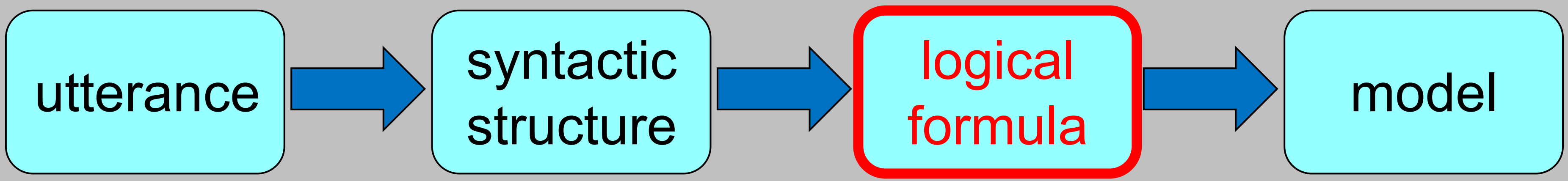
Tina is tall and thin

Taking stock



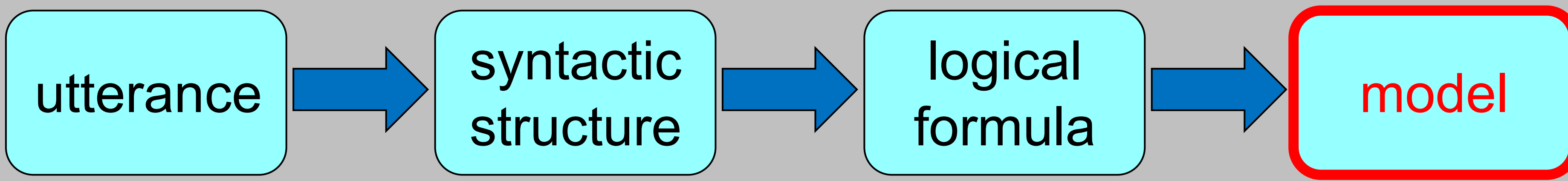


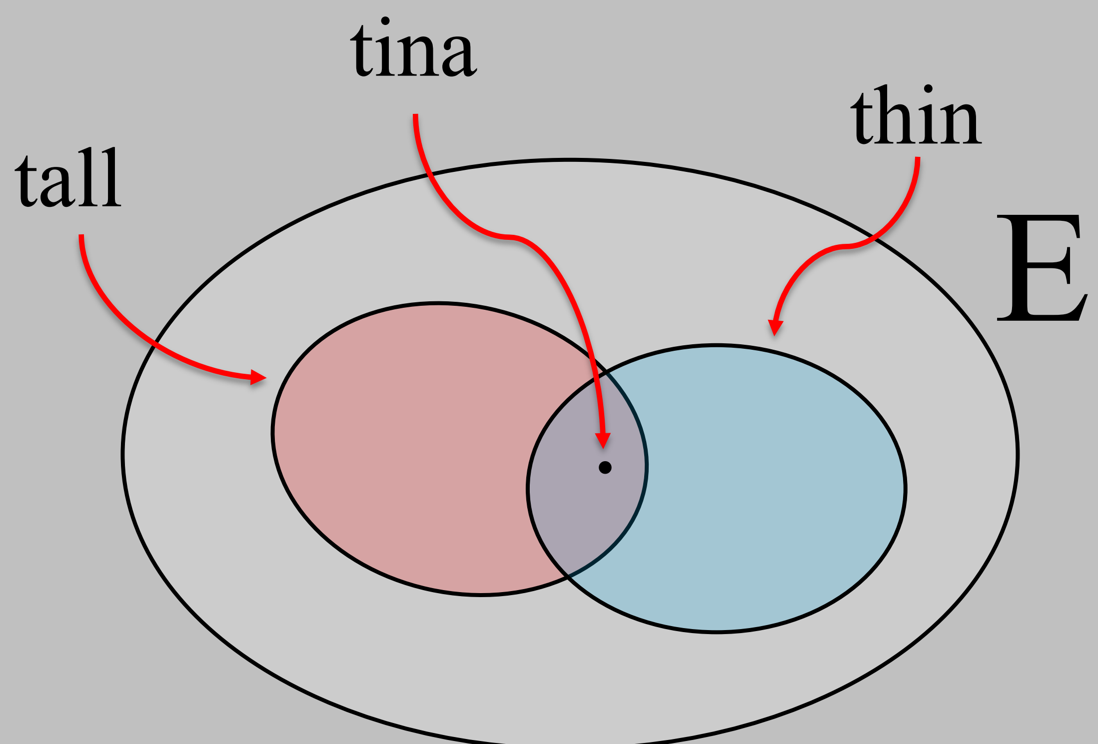
Taking stock



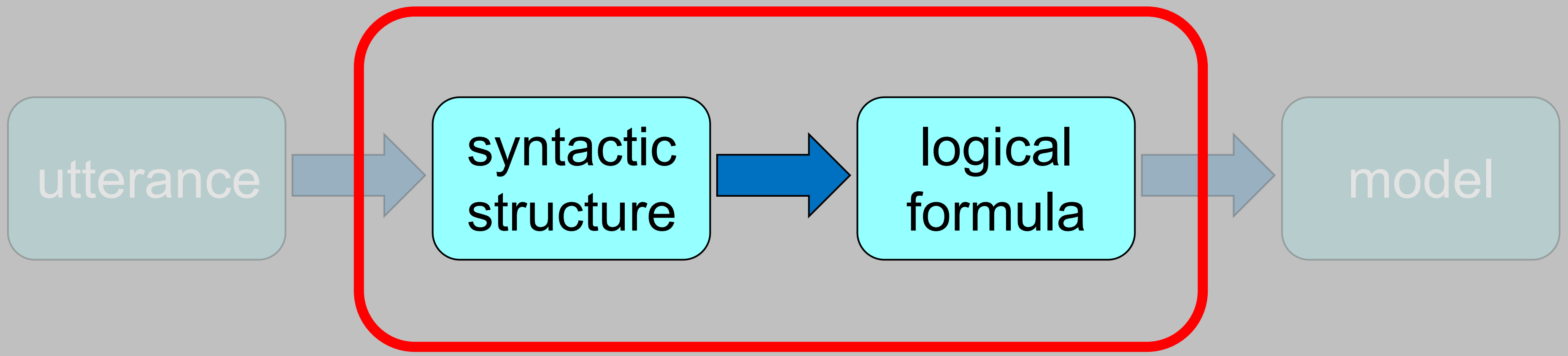
$$
\begin{aligned}
\mathrm{IS}\ (\mathrm{AND\ TALL\ THIN})\ \mathrm{TINA} &= (\lambda px.\, p\, x)\ (\mathrm{AND\ TALL\ THIN})\ \mathrm{TINA} \\
&\to_\beta (\lambda x.\, \mathrm{AND\ TALL\ THIN}\ x)\ \mathrm{TINA} \\
&\to_\beta \mathrm{AND\ TALL\ THIN\ TINA} \\
&= (\lambda pqx.\, (p\, x) \wedge (q\, x))\ \mathrm{TALL\ THIN\ TINA} \\
&\to_\beta (\lambda qx.\, (\mathrm{TALL}\ x) \wedge (q\, x))\ \mathrm{THIN\ TINA} \\
&= (\lambda qx.\, ((\lambda x.\, \mathbf{tall}\, x)\ x) \wedge (q\, x))\ \mathrm{THIN\ TINA} \\
&\to_\beta (\lambda qx.\, (\mathbf{tall}\, x) \wedge (q\, x))\ \mathrm{THIN\ TINA} \\
&\to_\beta (\lambda x.\, (\mathbf{tall}\, x) \wedge (\mathrm{THIN}\ x))\ \mathrm{TINA} \\
&= (\lambda x.\, (\mathbf{tall}\, x) \wedge ((\lambda x.\, \mathbf{thin}\, x)\ x))\ \mathrm{TINA} \\
&\to_\beta (\lambda x.\, (\mathbf{tall}\, x) \wedge (\mathbf{thin}\, x))\ \mathrm{TINA} \\
&\to_\beta (\mathbf{tall}\ \mathrm{TINA}) \wedge (\mathbf{thin}\ \mathrm{TINA}) \\
&= (\mathbf{tall}\ \mathbf{tina}) \wedge (\mathbf{thin}\ \mathbf{tina})
\end{aligned}
$$

Taking stock

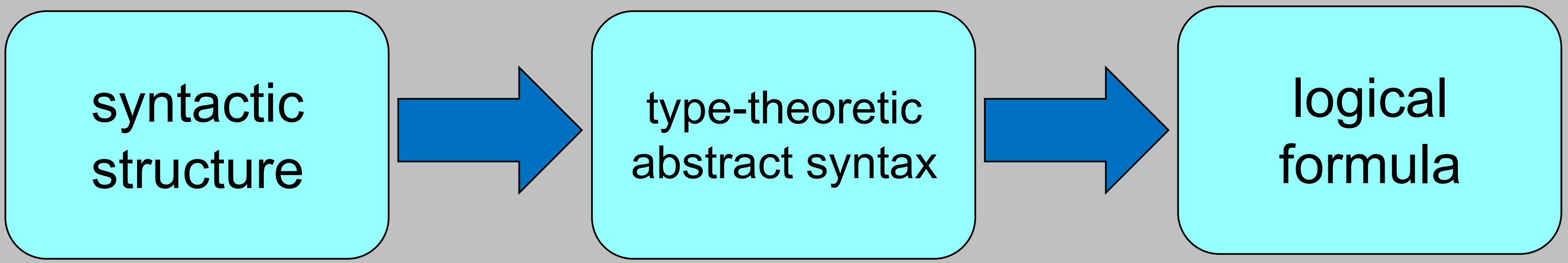




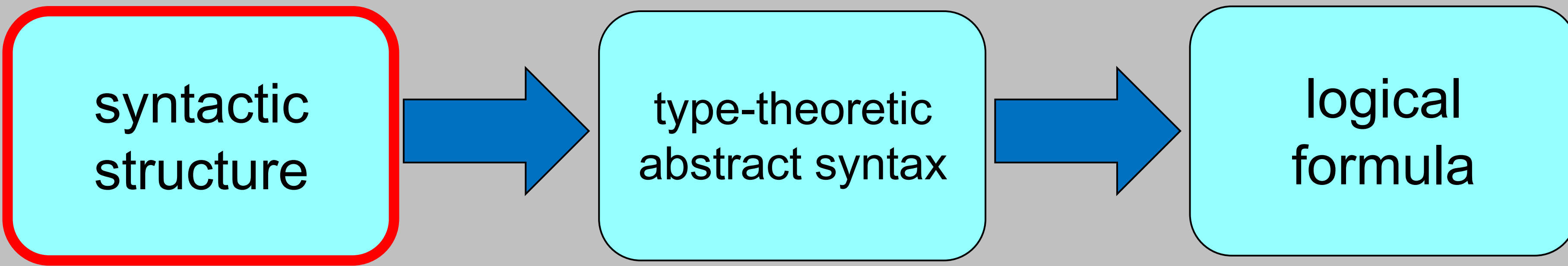
Syntax-semantics interface

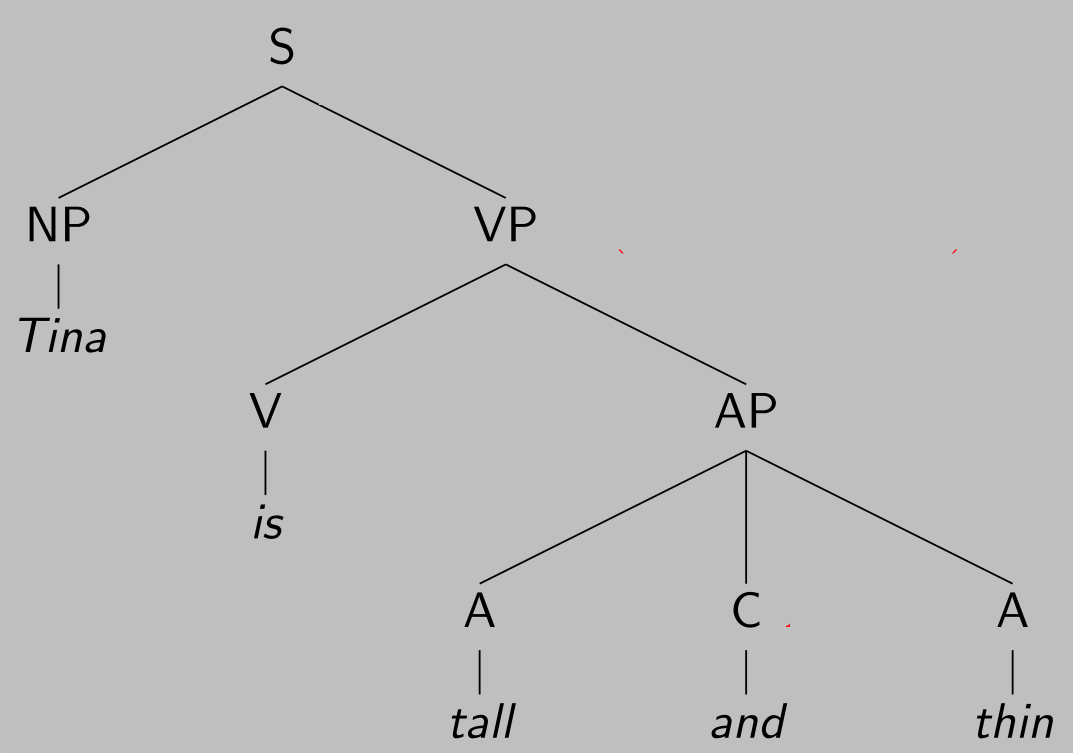


Syntax-semantics interface

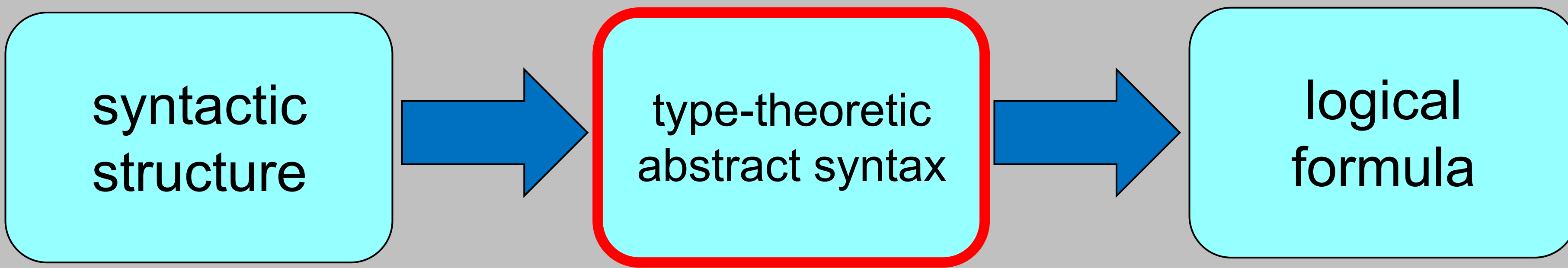


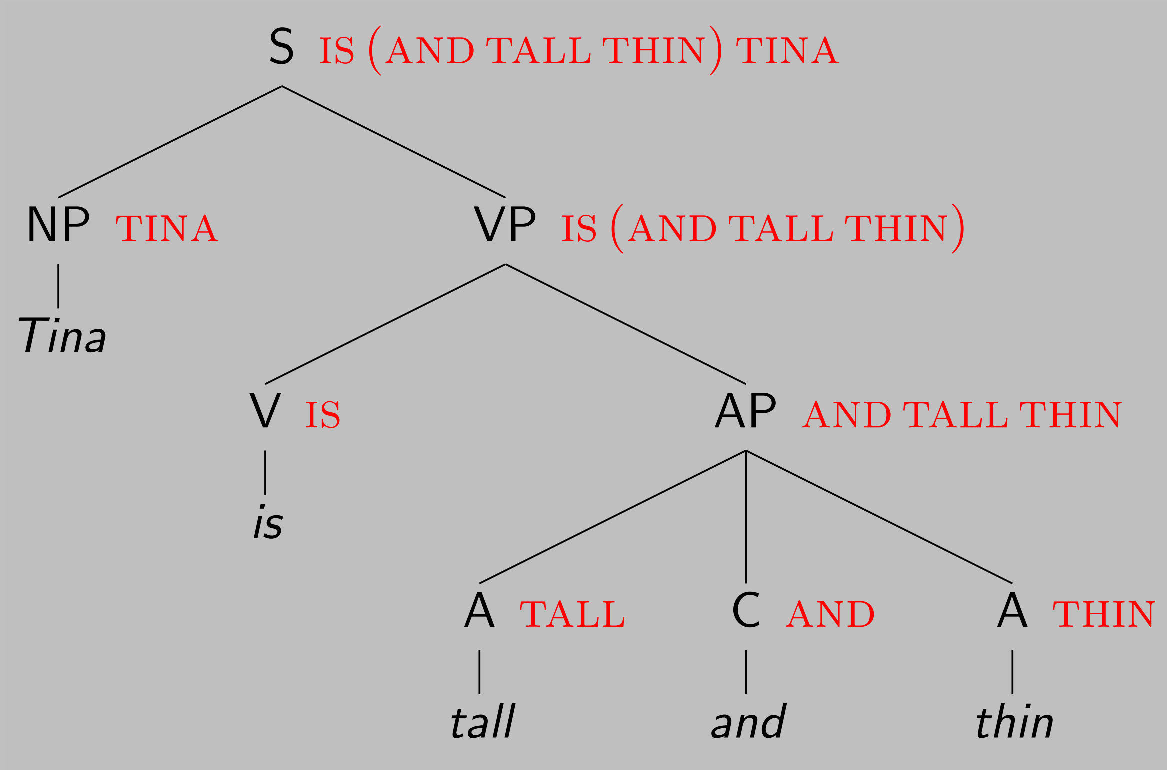
Syntax-semantics interface



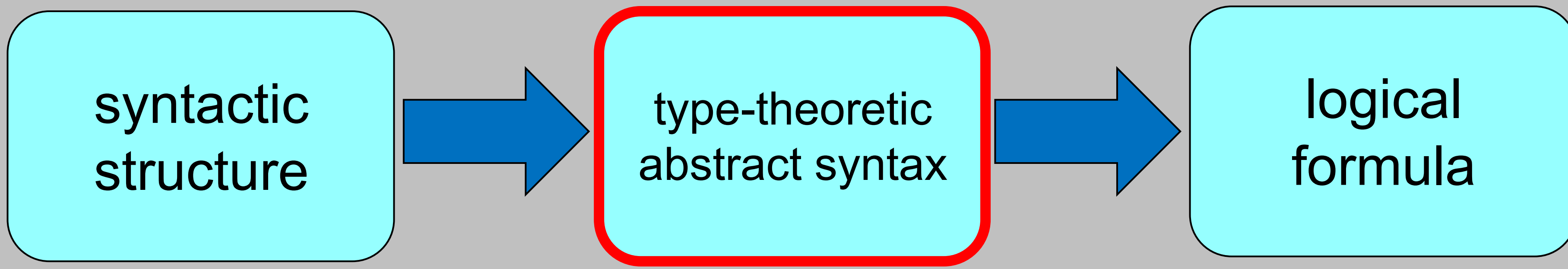


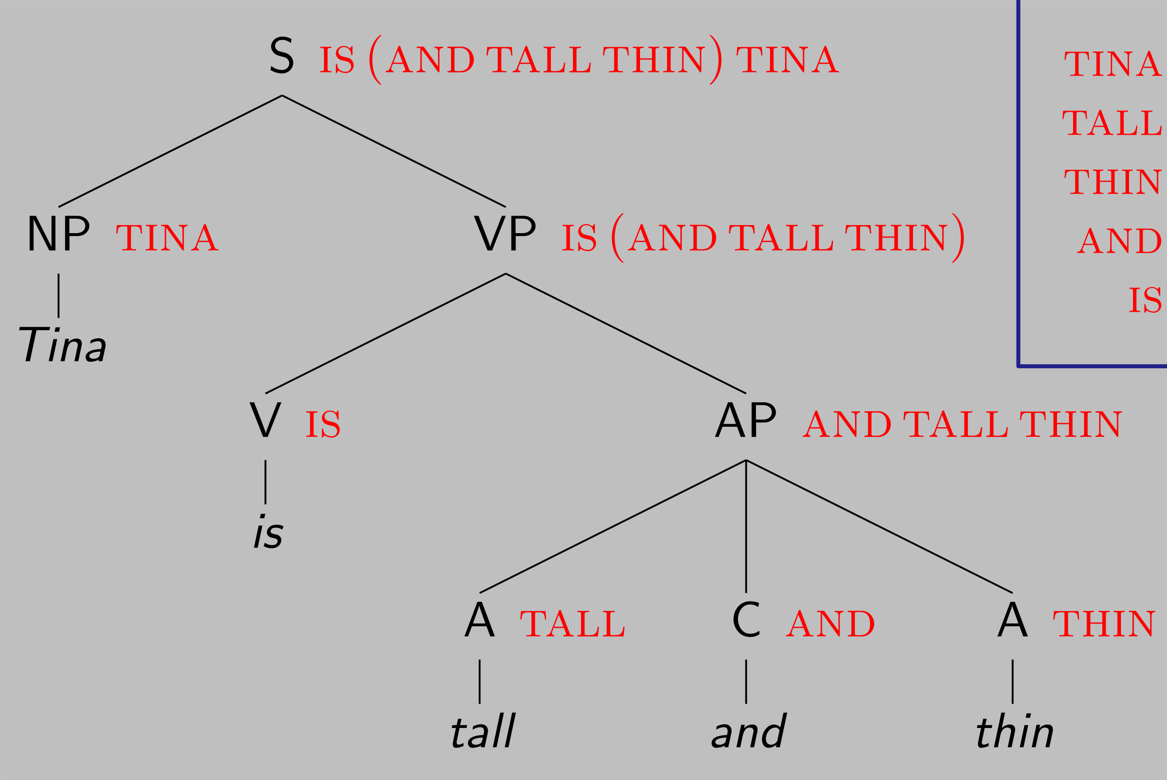
Syntax-semantics interface





Syntax-semantics interface





$$\text{TINA} : \mathbf{e}$$
$$\text{TALL} : \mathbf{e} \rightarrow \mathbf{t}$$
$$\text{THIN} : \mathbf{e} \rightarrow \mathbf{t}$$
$$\text{AND} : (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{e} \rightarrow \mathbf{t}$$
$$\text{IS} : (\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{e} \rightarrow \mathbf{t}$$

Syntax-semantics interface

$$\textsc{is} : \text{AP} \rightarrow (\text{NP} \rightarrow \text{S})$$

$$\textsc{and} : \text{AP} \rightarrow (\text{AP} \rightarrow \text{AP})$$

$$\textsc{tall} : \text{AP}$$

$$\textsc{thin} : \text{AP}$$

$$\textsc{tina} : \text{NP}$$

Syntax-semantics interface

$$\textsc{is} : AP \rightarrow (NP \rightarrow S)$$
$$\textsc{and} : AP \rightarrow (AP \rightarrow AP)$$
$$\textsc{tall} : AP$$
$$\textsc{thin} : AP$$
$$\textsc{tina} : NP$$

$$S := \mathbf{t}$$
$$NP := \mathbf{e}$$
$$AP := \mathbf{e} \rightarrow \mathbf{t}$$

Syntax-semantics interface

$$\begin{aligned} \textsc{is} &: \mathrm{AP} \rightarrow (\mathrm{NP} \rightarrow \mathrm{S}) \\ \textsc{and} &: \mathrm{AP} \rightarrow (\mathrm{AP} \rightarrow \mathrm{AP}) \\ \textsc{tall} &: \mathrm{AP} \\ \textsc{thin} &: \mathrm{AP} \\ \textsc{tina} &: \mathrm{NP} \end{aligned}$$

$$\begin{aligned} \mathrm{S} &:= \mathbf{t} \\ \mathrm{NP} &:= \mathbf{e} \\ \mathrm{AP} &:= \mathbf{e} \rightarrow \mathbf{t} \end{aligned}$$

$$\begin{aligned} \textsc{tina} &:= \mathbf{tina} \\ \textsc{tall} &:= \lambda x.\, \mathbf{tall}\, x \\ \textsc{thin} &:= \lambda x.\, \mathbf{thin}\, x \\ \textsc{and} &:= \lambda pqx.\, (p\, x) \wedge (q\, x) \\ \textsc{is} &:= \lambda px.\, p\, x \end{aligned}$$

Noun phrases (naive interpretation)

Noun phrases (naive interpretation)

Abstract syntax:

$$\begin{aligned} \textsc{Tina} &: \mathrm{NP} \\ \textsc{Mary} &: \mathrm{NP} \\ \textsc{praised} &: \mathrm{NP} \rightarrow \mathrm{NP} \rightarrow \mathrm{S} \end{aligned}$$

Noun phrases (naive interpretation)

Abstract syntax:

$$\begin{aligned} \textsc{Tina} &: \mathrm{NP} \\ \textsc{Mary} &: \mathrm{NP} \\ \textsc{praised} &: \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \end{aligned}$$

Semantic interpretation:

Noun phrases (naive interpretation)

Abstract syntax:

$$
\begin{aligned}
\textsc{Tina} &: \mathrm{NP} \\
\textsc{Mary} &: \mathrm{NP} \\
\textsc{praised} &: \mathrm{NP} \rightarrow \mathrm{NP} \rightarrow \mathrm{S}
\end{aligned}
$$

Semantic interpretation:

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

Noun phrases (naive interpretation)

Abstract syntax:

$$\begin{aligned} \textsc{Tina} &: \mathrm{NP} \\ \textsc{Mary} &: \mathrm{NP} \\ \textsc{praised} &: \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \end{aligned}$$

Semantic interpretation:

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

$$\begin{aligned} \textsc{Tina} &:= \mathbf{tina} \\ \textsc{Mary} &:= \mathbf{mary} \\ \textsc{praised} &:= \lambda xy.\, \mathbf{praised}\, y\, x \end{aligned}$$

where:

$$\begin{aligned} &\mathbf{tina}, \mathbf{mary} : \mathbf{e} \\ &\mathbf{praised} : \mathbf{e} \to \mathbf{e} \to \mathbf{t} \end{aligned}$$

Quantified noun phrases

- Tina praised Mary.
- Everybody praised Mary.
- Nobody praised Mary.
- Tina praised somebody.
- Everybody praised somebody.
- Everybody ran.

Generalized quantifiers & type raising

Generalized quantifiers & type raising

- Remember that $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).

Generalized quantifiers & type raising

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

Generalized quantifiers & type raising

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

Generalized quantifiers & type raising

- Remember that $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$ is the type of the logical constants **all** (∀) and **exists** (∃).
- Every term of type $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.
- The expected meaning of "*everybody ran*" might be captured by the following formula:

$$\forall x.\ (\mathbf{human}\ x) \to (\mathbf{ran}\ x)$$

Generalized quantifiers & type raising

- Remember that $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$ is the type of the logical constants **all** ($\forall$) and **exists** ($\exists$).
- Every term of type $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$ is called a generalized quantifier.
- Semantically, a generalized quantifier corresponds to a set of sets of entities.
- The expected meaning of "*everybody ran*" might be captured by the following formula:

$$\forall x.\, (\mathbf{human}\ x) \to (\mathbf{ran}\ x)$$

- Accordingly, the following $\boldsymbol{\lambda}$-term, which is of type $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$, is a good candidate for the interpretation of "*everybody*":

$$\boldsymbol{\lambda} p.\, \forall x.\, (\mathbf{human}\ x) \to (p\ x)$$

Generalized quantifiers & type raising

Syntax/semantics interface:

$$\begin{aligned}
\textsc{tina} &: \mathrm{NP} \\
\textsc{mary} &: \mathrm{NP} \\
\textsc{everybody} &: \mathrm{NP} \\
\textsc{somebody} &: \mathrm{NP} \\
\textsc{ran} &: \mathrm{NP} \to \mathrm{S} \\
\textsc{praised} &: \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}
\end{aligned}$$

Semantic interpretation:

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

$$\begin{aligned}
\textsc{tina} &:= ? \\
\textsc{mary} &:= ? \\
\textsc{everybody} &:= \lambda k.\, \forall x.\, (\mathbf{human}\, x) \to (k\, x) \\
\textsc{somebody} &:= \lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x) \\
\textsc{ran} &:= ? \\
\textsc{praised} &:= ?
\end{aligned}$$

Proper names as generalized quantifiers

Proper names as generalized quantifiers

- The interpretations of TINA and MARY must be of type **$(e \rightarrow t) \rightarrow t$.**

Proper names as generalized quantifiers

- The interpretations of TINA and MARY must be of type $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$.
- Semantically, it means that we must characterize an entity using a set of sets of entities.

Proper names as generalized quantifiers

- The interpretations of TINA and MARY must be of type $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$.
- Semantically, it means that we must characterize an entity using a set of sets of entities.
- $\{\, S \in \mathcal{P}(\mathrm{E}) : \mathbf{tina} \in S \,\}$

Proper names as generalized quantifiers

- The interpretations of TINA and MARY must be of type $(\mathbf{e} \rightarrow \mathbf{t}) \rightarrow \mathbf{t}$.
- Semantically, it means that we must characterize an entity using a set of sets of entities.
- $\{ S \in \mathcal{P}(\mathrm{E}) : \mathbf{tina} \in S \}$
- $\lambda S.\ S\ \mathbf{tina}$

Applying type-raising to verb arguments

- The syntactic type of RAN is (NP S), the semantic interpretation of NP is $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$, and the one of S is $\mathbf{t}$. Accordingly the type of the interpretation of RAN must be $((\mathbf{e} \to \mathbf{t}) \to \mathbf{t}) \to \mathbf{t}$
- RAN := $\boldsymbol{\lambda} s.\ s\ (\boldsymbol{\lambda} x.\ \mathbf{ran}\ x)$
- Similarly, the type of the interpretation of PRAISED must be $((\mathbf{e} \to \mathbf{t}) \to \mathbf{t}) \to ((\mathbf{e} \to \mathbf{t}) \to \mathbf{t}) \to \mathbf{t}$.
- PRAISED := $\boldsymbol{\lambda} o s.\ s\ (\boldsymbol{\lambda} x.\ o\ (\boldsymbol{\lambda} y.\ \mathbf{praised}\ x\ y))$

Generalized quantifiers & type raising

Syntax/semantics interface:

$$\begin{aligned}
\textsc{tina} &: \text{NP} \\
\textsc{mary} &: \text{NP} \\
\textsc{everybody} &: \text{NP} \\
\textsc{somebody} &: \text{NP} \\
\textsc{ran} &: \text{NP} \to \text{S} \\
\textsc{praised} &: \text{NP} \to \text{NP} \to \text{S}
\end{aligned}$$

Semantic interpretation:

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

$$\begin{aligned}
\textsc{tina} &:= \lambda k.\, k\, \mathbf{tina} \\
\textsc{mary} &:= \lambda k.\, k\, \mathbf{mary} \\
\textsc{everybody} &:= \lambda k.\, \forall x.\, (\mathbf{human}\, x) \to (k\, x) \\
\textsc{somebody} &:= \lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x) \\
\textsc{ran} &:= \lambda s.\, s\, (\lambda x.\, \mathbf{ran}\, x) \\
\textsc{praised} &:= \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y))
\end{aligned}$$

Generalized quantifiers & type raising

Tina praised somebody.

Generalized quantifiers & type raising

Tina praised somebody.

PRAISED SOMEBODY TINA

Generalized quantifiers & type raising

Tina praised somebody.

$$\text{PRAISED SOMEBODY TINA}$$

$$= \quad (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \text{SOMEBODY TINA}$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
& \text{PRAISED SOMEBODY TINA} \\
= \quad & (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\,x\,y)))\ \text{SOMEBODY TINA} \\
\to_\beta \quad & (\lambda s.\, s\,(\lambda x.\, \text{SOMEBODY}\,(\lambda y.\, \mathbf{praised}\,x\,y)))\ \text{TINA}
\end{aligned}
$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
&\textsc{praised somebody tina} \\
&= (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody tina} \\
&\to_\beta (\lambda s.\, s\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
&\to_\beta \textsc{tina}\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y))
\end{aligned}
$$

Generalized quantifiers & type raising

Tina praised somebody.

$$\begin{aligned}
&\textsc{praised somebody tina} \\
&\quad = \quad (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody tina} \\
&\quad \to_\beta \quad (\lambda s.\, s\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
&\quad \to_\beta \quad \textsc{tina}\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)) \\
&\quad = \quad (\lambda k.\, k\, \mathbf{tina})\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y))
\end{aligned}$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
&\textsc{praised}\ \textsc{somebody}\ \textsc{tina} \\
&\quad = \quad (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody}\ \textsc{tina} \\
&\quad \rightarrow_\beta \quad (\lambda s.\, s\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
&\quad \rightarrow_\beta \quad \textsc{tina}\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)) \\
&\quad = \quad (\lambda k.\, k\, \mathbf{tina})\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y)) \\
&\quad \rightarrow_\beta \quad (\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\, x\, y))\, \mathbf{tina}
\end{aligned}
$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
& \textsc{praised}\ \textsc{somebody}\ \textsc{tina} \\
= \quad & (\lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody}\ \textsc{tina} \\
\to_\beta \quad & (\lambda s.\, s\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
\to_\beta \quad & \textsc{tina}\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
= \quad & (\lambda k.\, k\, \mathbf{tina})\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
\to_\beta \quad & (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y))\, \mathbf{tina} \\
\to_\beta \quad & \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y)
\end{aligned}
$$

Generalized quantifiers & type raising

Tina praised somebody.

$$\begin{aligned}
& \textsc{praised somebody tina} \\
= \quad & (\lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody tina} \\
\to_\beta \quad & (\lambda s.\, s\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
\to_\beta \quad & \textsc{tina}\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
= \quad & (\lambda k.\, k\, \mathbf{tina})\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
\to_\beta \quad & (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y))\, \mathbf{tina} \\
\to_\beta \quad & \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y) \\
= \quad & (\lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x))\, (\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y)
\end{aligned}$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
& \textsc{praised}\ \textsc{somebody}\ \textsc{tina} \\
=\quad & (\lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{somebody}\ \textsc{tina} \\
\to_\beta\quad & (\lambda s.\, s\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)))\ \textsc{tina} \\
\to_\beta\quad & \textsc{tina}\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
=\quad & (\lambda k.\, k\, \mathbf{tina})\, (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y)) \\
\to_\beta\quad & (\lambda x.\, \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, x\, y))\, \mathbf{tina} \\
\to_\beta\quad & \textsc{somebody}\, (\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y) \\
=\quad & (\lambda k.\, \exists x.\, (\mathbf{human}\, x) \wedge (k\, x))\, (\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y) \\
\to_\beta\quad & \exists x.\, (\mathbf{human}\, x) \wedge ((\lambda y.\, \mathbf{praised}\, \mathbf{tina}\, y)\, x)
\end{aligned}
$$

Generalized quantifiers & type raising

Tina praised somebody.

$$
\begin{aligned}
& \textsc{praised}\;\textsc{somebody}\;\textsc{tina} \\
= \quad & (\lambda o.\, \lambda s.\, s\,(\lambda x.\, o\,(\lambda y.\, \mathbf{praised}\; x\; y)))\;\textsc{somebody}\;\textsc{tina} \\
\rightarrow_\beta \quad & (\lambda s.\, s\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\; x\; y)))\;\textsc{tina} \\
\rightarrow_\beta \quad & \textsc{tina}\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\; x\; y)) \\
= \quad & (\lambda k.\, k\;\mathbf{tina})\,(\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\; x\; y)) \\
\rightarrow_\beta \quad & (\lambda x.\, \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\; x\; y))\;\mathbf{tina} \\
\rightarrow_\beta \quad & \textsc{somebody}\,(\lambda y.\, \mathbf{praised}\;\mathbf{tina}\; y) \\
= \quad & (\lambda k.\, \exists x.\, (\mathbf{human}\; x) \wedge (k\; x))\,(\lambda y.\, \mathbf{praised}\;\mathbf{tina}\; y) \\
\rightarrow_\beta \quad & \exists x.\, (\mathbf{human}\; x) \wedge ((\lambda y.\, \mathbf{praised}\;\mathbf{tina}\; y)\; x) \\
\rightarrow_\beta \quad & \exists x.\, (\mathbf{human}\; x) \wedge (\mathbf{praised}\;\mathbf{tina}\; x)
\end{aligned}
$$

Nouns & Determiners

Syntax/semantics interface:

$$
\begin{aligned}
\textsc{tina} &: \mathrm{NP} \\
\textsc{mary} &: \mathrm{NP} \\
\textsc{everybody} &: \mathrm{NP} \\
\textsc{somebody} &: \mathrm{NP} \\
\textsc{man} &: \mathrm{N} \\
\textsc{woman} &: \mathrm{N} \\
\textsc{every} &: \mathrm{N} \to \mathrm{NP} \\
\textsc{some} &: \mathrm{N} \to \mathrm{NP} \\
\textsc{ran} &: \mathrm{NP} \to \mathrm{S} \\
\textsc{praised} &: \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}
\end{aligned}
$$

Semantic interpretation:

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

Nouns & Determiners

Semantic interpretation:

$$
\begin{aligned}
\text{TINA} &:= \lambda k.\, k\, \mathbf{tina} \\
\text{MARY} &:= \lambda k.\, k\, \mathbf{mary} \\
\text{EVERYBODY} &:= \lambda k.\, \forall x.\, (\mathbf{human}\, x) \to (k\, x) \\
\text{SOMEBODY} &:= \lambda k.\, \exists x.\, (\mathbf{human}\, x) \land (k\, x) \\
\text{MAN} &:= \lambda x.\, \mathbf{man}\, x \\
\text{WOMAN} &:= \lambda x.\, \mathbf{woman}\, x \\
\text{EVERY} &:= \lambda n.\, \lambda m.\, \forall x.\, n\, x \to m\, x \\
\text{SOME} &:= \lambda n.\, \lambda m.\, \exists x.\, n\, x \land m\, x \\
\text{RAN} &:= \lambda s.\, s\, (\lambda x.\, \mathbf{ran}\, x) \\
\text{PRAISED} &:= \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y))
\end{aligned}
$$

where:

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

Determiners as binary generalized quantifiers

Determiners as binary generalized quantifiers

- The type of the interpretations of EVERY and SOME is **$(e \rightarrow t) \rightarrow (e \rightarrow t) \rightarrow t$**.

Determiners as binary generalized quantifiers

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

Determiners as binary generalized quantifiers

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

Determiners as binary generalized quantifiers

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

Scope ambiguities

Scope ambiguities

Every man praised a woman

Scope ambiguities

Every man praised a woman

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

Scope ambiguities

Every man praised a woman

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

Subject wide scope:

$$\textsc{praised} = \lambda o.\, \lambda s.\, s\, (\lambda x.\, o\, (\lambda y.\, \mathbf{praised}\, x\, y))$$

Object wide scope:

$$\textsc{praised}_{\textsc{ows}} = \lambda o.\, \lambda s.\, o\, (\lambda y.\, s\, (\lambda x.\, \mathbf{praised}\, x\, y))$$