

Computing Natural Language Semantics

Informatics 2A: Lecture 24

John Longley

12 November 2013

- 1 Semantic Composition for NL
 - Logical Form
- 2 Semantic Composition for NL
- 3 Semantic (Scope) Ambiguity
 - Definition
 - Semantic Scope
 - Approaches to Scope Ambiguity
 - Underspecification: General Idea

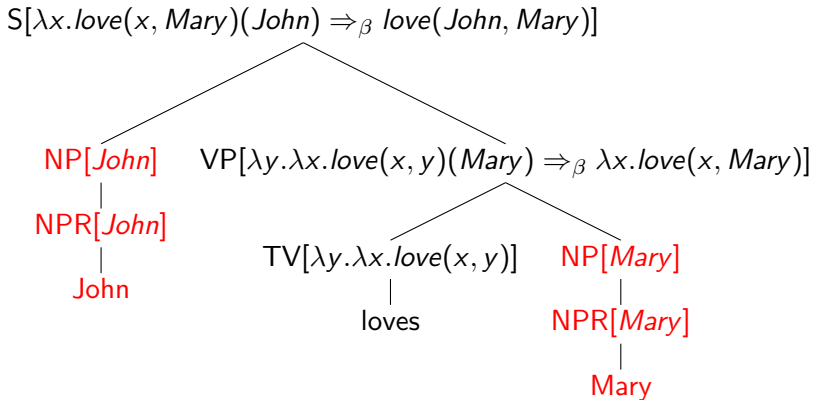
Compositional Semantics: the key idea

Grammar 1

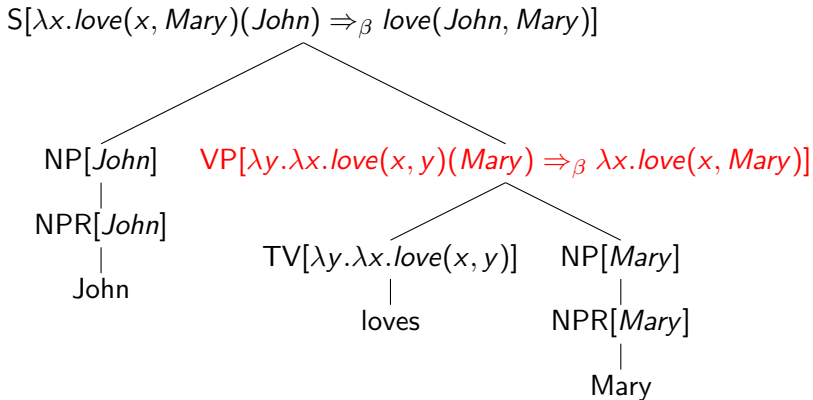
$S \rightarrow NP VP$	$\{VP.Sem(NP.Sem)\}$	t
$VP \rightarrow TV NP$	$\{TV.Sem(NP.Sem)\}$	$\langle e, t \rangle$
$NP \rightarrow NPR$	$\{NPR.Sem\}$	e
$TV \rightarrow \text{loves}$	$\{\lambda y. \lambda x. \text{love}(x, y)\}$	$\langle e, \langle e, t \rangle \rangle$
$NPR \rightarrow \text{Mary}$	$\{\text{Mary}\}$	e
$NPR \rightarrow \text{John}$	$\{\text{John}\}$	e

- To build a compositional semantics for NL, we attach **valuation functions** to grammar rules (**semantic attachments**).
- These show how to compute the interpretation of the LHS of the rule from the interpretations of its RHS components.
- For example, $VP.Sem(NP.Sem)$ means **apply** the interpretation of the VP to the interpretation of the NP.
- **Types** have been added to ease understanding.

Compositional Semantics: example

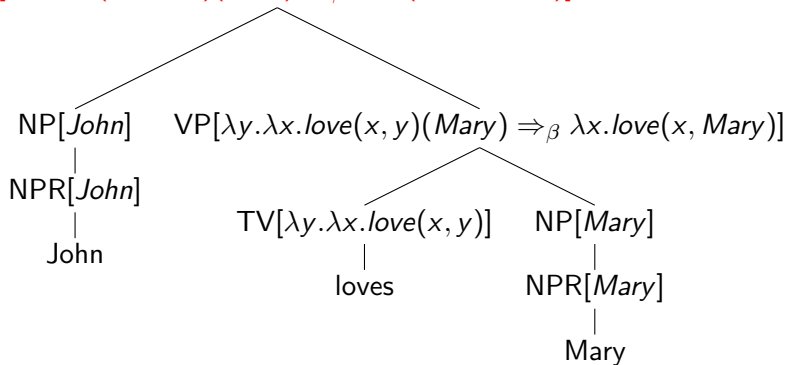


Compositional Semantics: example



Compositional Semantics: example

$S[\lambda x.love(x, Mary)(John) \Rightarrow_{\beta} love(John, Mary)]$



A minor variation

The following alternative semantics assigns the same overall meaning to sentences. Only the treatment of the arguments of 'love' is different.

Grammar I

$S \rightarrow NP VP$	$\{VP.Sem(NP.Sem)\}$	t
$VP \rightarrow TV NP$	$\{\lambda x. TV.Sem(x)(NP.Sem)\}$	$\langle e, t \rangle$
$NP \rightarrow NPR$	$\{NPR.Sem\}$	e
$TV \rightarrow loves$	$\{\lambda x. \lambda y. love(x,y)\}$	$\langle e, \langle e, t \rangle \rangle$
$NPR \rightarrow Mary$	$\{Mary\}$	e
$NPR \rightarrow John$	$\{John\}$	e

Compositional Semantics, continued

What about the interpretation of an NP other than a proper name? The FOPL interpretation should often contain an existential (\exists) or a universal (\forall) quantifier:

John has access to **a computer**.

$\exists x(\text{computer}(x) \wedge \text{have_access_to}(\text{john}, x))$

Every student has access to a computer.

$\forall x(\text{student}(x) \rightarrow \exists y(\text{computer}(y) \wedge \text{have_access_to}(x, y)))$

Can we build such interpretations up from their component parts in the same way as with proper names?

A halfway stage.

Grammar II

$S \rightarrow \text{NPR VP}$	$\{ \text{VP.Sem}(\text{NPR.Sem}) \}$	t
$\text{VP} \rightarrow \text{TV a Nom}$	$\{ \lambda x. \exists y. \text{Nom.Sem}(y) \&$ $\text{TV.Sem}(y)(x) \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{N}$	$\{ \text{N.Sem} \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{A Nom}$	$\{ \lambda x. \text{Nom.Sem}(x) \& \text{A.Sem}(x) \}$	$\langle e, t \rangle$
$\text{NPR} \rightarrow \text{John}$	$\{ \textit{John} \}$	e
$\text{TV} \rightarrow \text{loves}$	$\{ \lambda y. \lambda x. \textit{love}(x, y) \}$	$\langle e, \langle e, t \rangle \rangle$
$\text{N} \rightarrow \text{girl}$	$\{ \lambda z. \textit{girl}(z) \}$	$\langle e, t \rangle$
$\text{A} \rightarrow \text{tall}$	$\{ \lambda z. \textit{tall}(z) \}$	$\langle e, t \rangle$

- Note we haven't given a meaning here to **a tall girl**.
- Could take this to have the same meaning as **tall girl**.
- This would be fine for this example (also in Assignment 2).
 But what about **every tall girl**?

A halfway stage.

Grammar II

$S \rightarrow \text{NPR VP}$	$\{ \text{VP.Sem}(\text{NPR.Sem}) \}$	t
$\text{VP} \rightarrow \text{TV a Nom}$	$\{ \lambda x. \exists y. \text{Nom.Sem}(y) \&$ $\text{TV.Sem}(y)(x) \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{N}$	$\{ \text{N.Sem} \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{A Nom}$	$\{ \lambda x. \text{Nom.Sem}(x) \& \text{A.Sem}(x) \}$	$\langle e, t \rangle$
$\text{NPR} \rightarrow \text{John}$	$\{ \textit{John} \}$	e
$\text{TV} \rightarrow \text{loves}$	$\{ \lambda y. \lambda x. \textit{love}(x, y) \}$	$\langle e, \langle e, t \rangle \rangle$
$\text{N} \rightarrow \text{girl}$	$\{ \lambda z. \textit{girl}(z) \}$	$\langle e, t \rangle$
$\text{A} \rightarrow \text{tall}$	$\{ \lambda z. \textit{tall}(z) \}$	$\langle e, t \rangle$

- Note we haven't given a meaning here to **a tall girl**.
- Could take this to have the same meaning as **tall girl**.
- This would be fine for this example (also in Assignment 2).
 But what about **every tall girl**?

A halfway stage.

Grammar II

$S \rightarrow \text{NPR VP}$	$\{ \text{VP.Sem}(\text{NPR.Sem}) \}$	t
$\text{VP} \rightarrow \text{TV a Nom}$	$\{ \lambda x. \exists y. \text{Nom.Sem}(y) \& \text{TV.Sem}(y)(x) \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{N}$	$\{ \text{N.Sem} \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{A Nom}$	$\{ \lambda x. \text{Nom.Sem}(x) \& \text{A.Sem}(x) \}$	$\langle e, t \rangle$
$\text{NPR} \rightarrow \text{John}$	$\{ \text{John} \}$	e
$\text{TV} \rightarrow \text{loves}$	$\{ \lambda y. \lambda x. \text{love}(x, y) \}$	$\langle e, \langle e, t \rangle \rangle$
$\text{N} \rightarrow \text{girl}$	$\{ \lambda z. \text{girl}(z) \}$	$\langle e, t \rangle$
$\text{A} \rightarrow \text{tall}$	$\{ \lambda z. \text{tall}(z) \}$	$\langle e, t \rangle$

- Note we haven't given a meaning here to **a tall girl**.
- Could take this to have the same meaning as **tall girl**.
- This would be fine for this example (also in Assignment 2).
 But what about **every tall girl**?

A halfway stage.

Grammar II

$S \rightarrow \text{NPR VP}$	$\{ \text{VP.Sem}(\text{NPR.Sem}) \}$	t
$\text{VP} \rightarrow \text{TV a Nom}$	$\{ \lambda x. \exists y. \text{Nom.Sem}(y) \& \text{TV.Sem}(y)(x) \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{N}$	$\{ \text{N.Sem} \}$	$\langle e, t \rangle$
$\text{Nom} \rightarrow \text{A Nom}$	$\{ \lambda x. \text{Nom.Sem}(x) \& \text{A.Sem}(x) \}$	$\langle e, t \rangle$
$\text{NPR} \rightarrow \text{John}$	$\{ \text{John} \}$	e
$\text{TV} \rightarrow \text{loves}$	$\{ \lambda y. \lambda x. \text{love}(x, y) \}$	$\langle e, \langle e, t \rangle \rangle$
$\text{N} \rightarrow \text{girl}$	$\{ \lambda z. \text{girl}(z) \}$	$\langle e, t \rangle$
$\text{A} \rightarrow \text{tall}$	$\{ \lambda z. \text{tall}(z) \}$	$\langle e, t \rangle$

- Note we haven't given a meaning here to **a tall girl**.
- Could take this to have the same meaning as **tall girl**.
- This would be fine for this example (also in Assignment 2).
 But what about **every tall girl**?

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Computing semantics with Grammar II

Before we add more, let's use Grammar II to compute the semantics of **John loves a tall girl**.

loves	TV	$\lambda yx. \text{love}(x, y)$
tall girl	Nom	$\lambda x. (\lambda z. \text{girl}(z))(x) \ \& \ (\lambda z. \text{tall}(z))(x)$
	\Rightarrow_{β}	$\lambda x. \text{girl}(x) \ \& \ \text{tall}(x)$
loves a tall girl	VP	$\lambda x. \exists y. (\lambda x. \text{girl}(x) \ \& \ \text{tall}(x))(y) \ \& \ (\lambda yx. \text{love}(x, y))(y)(x)$
	\Rightarrow_{β}	$\lambda x. \exists y. (\text{girl}(y) \ \& \ \text{tall}(y)) \ \& \ \text{love}(x, y)$
John loves a tall girl	S	$(\lambda x. \exists y. \dots)(\text{John})$
	\Rightarrow_{β}	$\exists y. \text{girl}(y) \ \& \ \text{tall}(y) \ \& \ \text{love}(\text{John}, y)$

Type raising

- We've given **John**, **Mary** the semantic type e , and **girl** the semantic type $\langle e, t \rangle$.
- But what type should **some girl** or **every girl** have?
- **Idea:** Since we wish to combine an NP.Sem with a VP.Sem (of type $\langle e, t \rangle$) to get an S.Sem (of type t), let's try again with NP.Sem having type $\langle \langle e, t \rangle, t \rangle$.

John $\lambda P. P(\text{John})$ (type raising)
 every girl $\lambda P. \forall x. \text{girl}(x) \Rightarrow P(x)$

The appropriate semantic attachment for NP VP is then

$S \rightarrow \text{NP VP} \quad \{\text{NP.Sem (VP.Sem)}\}$

Semantics of determiners

- Using this approach, we can also derive the semantics of 'every girl' from that of 'every' and 'girl'.
- We've seen that 'girl' has semantic type $\langle e, t \rangle$, and 'every girl' has semantic type $\langle \langle e, t \rangle, t \rangle$.
- So the interpretation of 'every' should have type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$. Similarly for other *determiners* (e.g. every, a, no, not every).

girl	$\lambda x. \text{girl}(x)$	$\langle e, t \rangle$
every	$\lambda Q. \lambda P. \forall x. Q(x) \Rightarrow P(x)$	$\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$
a	$\lambda Q. \lambda P. \exists x. Q(x) \wedge P(x)$	$\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$
NP \rightarrow Det N	{ Det.Sem (N.Sem) }	$\langle \langle e, t \rangle, t \rangle$

We can now compute the semantics of 'every girl' and check that it β -reduces to $\lambda P. \forall x. \text{girl}(x) \Rightarrow P(x)$.

More on type raising

- The natural rule for VP is now $VP \rightarrow TV NP$.
- Since the semantic type for NP has now been **raised** to $\langle \langle e, t \rangle, t \rangle$, and we want VP to have semantic type $\langle e, t \rangle$, what should the semantic type for TV be?

More on type raising

- The natural rule for VP is now $VP \rightarrow TV\ NP$.
- Since the semantic type for NP has now been **raised** to $\langle\langle e, t \rangle, t \rangle$, and we want VP to have semantic type $\langle e, t \rangle$, what should the semantic type for TV be?

It had better be $\langle\langle\langle e, t \rangle, t \rangle, \langle e, t \rangle\rangle$.
(A 3rd order function type!)

TV \rightarrow loves $\{\lambda R^{\langle\langle e, t \rangle, t \rangle} . \lambda z^e . R(\lambda w^e . loves(z, w))\}$
VP \rightarrow TV NP $\{TV.Sem(NP.Sem)\}$

To summarize where we've got to:

Grammar III

S → NP VP	{ NP.Sem(VP.Sem) }	t
VP → TV NP	{ TV.Sem(NP.Sem) }	$\langle e, t \rangle$
NP → John	{ $\lambda P.P(\text{John})$ }	$\langle \langle e, t \rangle, t \rangle$
NP → Det Nom	{ Det.Sem(Nom.Sem) }	$\langle \langle e, t \rangle, t \rangle$
Det → a	{ $\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x)$ }	$\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$
Det → every	{ $\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x)$ }	$\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$
Nom → N	{ N.Sem }	$\langle e, t \rangle$
Nom → A Nom	{ $\lambda x.\text{Nom.Sem}(x) \& \text{A.Sem}(x)$ }	$\langle e, t \rangle$
TV → loves	{ $\{\lambda R.\lambda z.R(\lambda w.\text{loves}(z, w))\}$	$\langle \langle \langle e, t \rangle, t \rangle, \langle e, t \rangle \rangle$
N → girl	{ $\lambda z.\text{girl}(z)$ }	$\langle e, t \rangle$
A → tall	{ $\lambda z.\text{tall}(z)$ }	$\langle e, t \rangle$

Can add similar entries for 'student', 'computer', 'has access to'.

Example

The semantics for 'every student has access to a computer'.

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
 $\rightarrow_{\beta} \lambda P.\forall x.student(x) \Rightarrow P(x)$

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
 $\rightarrow_{\beta} \lambda P.\forall x.student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x))$
 $\rightarrow_{\beta} \lambda P.\exists x.computer(x) \wedge P(x)$

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
 $\rightarrow_{\beta} \lambda P.\forall x.student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x))$
 $\rightarrow_{\beta} \lambda P.\exists x.computer(x) \wedge P(x)$

h.a.t. a computer $\dots \rightarrow_{\beta} \dots$
 $\rightarrow_{\beta} \lambda z.\exists x.computer(x) \wedge h_a_t(z, x)$

Example

The semantics for 'every student has access to a computer'.

every student $(\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x))$
 $\rightarrow_{\beta} \lambda P.\forall x.student(x) \Rightarrow P(x)$

a computer $(\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x))$
 $\rightarrow_{\beta} \lambda P.\exists x.computer(x) \wedge P(x)$

h.a.t. a computer $\dots \rightarrow_{\beta} \dots$
 $\rightarrow_{\beta} \lambda z.\exists x.computer(x) \wedge h_a_t(z, x)$

(whole sentence) $\dots \rightarrow_{\beta} \dots$
 $\rightarrow_{\beta} \forall x.student(x) \Rightarrow \exists y.computer(y) \wedge h_a_t(x, y)$

Example

The semantics for 'every student has access to a computer'.

$$\begin{aligned} \text{every student} & (\lambda Q.\lambda P.\forall x.Q(x) \Rightarrow P(x))(\lambda x.student(x)) \\ & \rightarrow_{\beta} \lambda P.\forall x.student(x) \Rightarrow P(x) \end{aligned}$$

$$\begin{aligned} \text{a computer} & (\lambda Q.\lambda P.\exists x.Q(x) \wedge P(x))(\lambda x.computer(x)) \\ & \rightarrow_{\beta} \lambda P.\exists x.computer(x) \wedge P(x) \end{aligned}$$

$$\begin{aligned} \text{h.a.t. a computer} & \dots \rightarrow_{\beta} \dots \\ & \rightarrow_{\beta} \lambda z.\exists x.computer(x) \wedge h_a_t(z, x) \end{aligned}$$

$$\begin{aligned} (\text{whole sentence}) & \dots \rightarrow_{\beta} \dots \\ & \rightarrow_{\beta} \forall x.student(x) \Rightarrow \exists y.computer(y) \wedge h_a_t(x, y) \end{aligned}$$

Note: In the last β -step, we've renamed 'x' to 'y' to avoid capture. 13 / 20

Clicker Question

Suppose that the predicate $L(x, y)$ means x loves y . Which of the following is **not** a possible representation of the meaning of *Everybody loves somebody*?

- 1 $\forall x. \exists y. L(x, y)$
- 2 $(\lambda P. \forall x. \exists y. P(x, y))(\lambda x. \lambda y. L(x, y))$
- 3 $(\lambda P. \forall x. \exists y. P(x, y))(\lambda x. \lambda y. L(y, x))$
- 4 $(\lambda P. \forall x. \exists y. P(y, x))(\lambda x. \lambda y. L(y, x))$

Semantic Ambiguity

Whilst **every student has access to a computer** is neither syntactically nor lexically ambiguous, it has **two different interpretations** because of its determiners:

- **every**: interpreted as \forall (*universal quantifier*)
- **a**: interpreted as \exists (*existential quantifier*)

Meaning 1

Possibly a different computer per student

$$\forall x(\text{student}(x) \rightarrow \exists y(\text{computer}(y) \wedge \text{have_access_to}(x, y)))$$

Meaning 2

Possibly the same computer for all students

$$\exists y(\text{computer}(y) \wedge \forall x(\text{student}(x) \rightarrow \text{have_access_to}(x, y)))$$

Scope

The ambiguity arises because **every** and **a** each has its own **scope**:

Interpretation 1: **every** has scope over **a**

Interpretation 2: **a** has scope over **every**

- Scope is not uniquely determined either by left-to-right order, or by position in the parse tree.
- We therefore need other mechanisms to ensure that the ambiguity is reflected by there being multiple interpretations assigned to S.

Scope ambiguity, continued

The number of interpretations grows exponentially with the number of scope operators:

Every student at some university has access to a laptop.

1. Not necessarily same laptop, not necessarily same university

$\forall x(stud(x) \wedge \exists y(univ(y) \wedge at(x, y)) \rightarrow \exists z(laptop(z) \wedge have_access(x, z)))$

2. Same laptop, not necessarily same university

$\exists z(laptop(z) \wedge \forall x(stud(x) \wedge \exists y(univ(y) \wedge at(x, y)) \rightarrow have_access(x, z)))$

3. Not necessarily same laptop, same university

$\exists y(univ(y) \wedge \forall x((stud(x) \wedge at(x, y)) \rightarrow \exists z(laptop(z) \wedge have_access(x, z))))$

4. Same university, same laptop

$\exists y(univ(y) \wedge \exists z(laptop(z) \wedge \forall x((stud(x) \wedge at(x, y)) \rightarrow have_access(x, z))))$

5. Same laptop, same university

$\exists z(laptop(z) \wedge \exists y(univ(y) \wedge \forall x((stud(x) \wedge at(x, y)) \rightarrow have_access(x, z))))$

where 4 & 5 are equivalent

Every student at some university does not have access to a computer.

→ 18 interpretations

Coping with Scope: options

- 1 **Enumerate all interpretations.** Computationally unattractive!
- 2 Use an **underspecified representation** that can be further specified to each of the multiple interpretations on demand.

Sometimes the surrounding context will help us choose between interpretations:

Every student has access to a computer. It can be borrowed from the ITO. (⇒ Meaning 2)

Underspecification

- The idea in underspecified representations is that instead of trying to associate a **single FOPL formula** with a sentence, we associate **fragments of formulae** with various parts of the sentence.
- These fragments can have **holes** into which other fragments can be plugged. Since there may be some freedom in the order of plugging, the same bunch of fragments can give rise to several formulae with different scoping orders.
- There may also be **constraints** on the order of plugging, corresponding to partial information about the intended interpretation derived e.g. from the discourse context.

See J&M Chapter 18.3 for more on this.

Summary

- Syntax guides semantic composition in a systematic way.
- Lambda expressions facilitate the construction of compositional semantic interpretations.
- Logical forms can be constructed by attaching valuation functions to grammar rules.
- However, this approach is not adequate enough for quantified NPs, as LFs are not always isomorphic with syntax.
- We can elegantly handle scope by building an abstract underspecified representation and disambiguate on demand.