Logica e Linguaggio

Raffaella Bernardi

University of Trento

March 22, 2012

Layout

Logic

PL: Propositions can be: atomic (p, q, ...) or complex ($\neg P, P \wedge Q, P \vee Q, P \to Q$)

- an interpretation $\mathcal{I}$ either satisfies a proposition ψ ($\mathcal{I} \models \psi$) or falsifies it ($\mathcal{I} \not\models \psi$). In the first case it is called a model $\mathcal{M}$ of the formula.
- The interpretation of the whole depends on the interpretation of the parts and of the logical operators connecting them, E.g.:

	p	$\neg p$	$p \vee \neg p$
$\mathcal{I}_1$:	T	F	T
$\mathcal{I}_2$:	F	T	T

Logic

PL: Propositions can be: atomic (p, q, ...) or complex ($\neg P, P \wedge Q, P \vee Q, P \rightarrow Q$)

- an interpretation $\mathcal{I}$ either satisfies a proposition ψ ($\mathcal{I} \models \psi$) or falsifies it ($\mathcal{I} \not\models \psi$). In the first case it is called a model $\mathcal{M}$ of the formula.
- The interpretation of the whole depends on the interpretation of the parts and of the logical operators connecting them, E.g.:

	p	$\neg p$	$p \vee \neg p$
$\mathcal{I}_1$:	T	F	T
$\mathcal{I}_2$:	F	T	T

	p	q	$p \rightarrow q$
$\mathcal{I}_1$:	T	T	T
$\mathcal{I}_2$:	T	F	F
$\mathcal{I}_3$:	F	T	T
$\mathcal{I}_4$:	F	F	T

notation: $\mathcal{I}_1 \not\models \neg p \quad \mathcal{I}_2 \models \neg p$

$p \vee \neg p$ is a tautology: it's true for all interpretations.

Logic

Logical Entailment (Validity) vs. Satisfiability

$$\{\psi_1, \ldots \psi_n\} \models \phi$$

- is *satisfiable* when there is at least one interpretation for which the premises and the conclusion are true.

Logic

Logical Entailment (Validity) vs. Satisfiability

$$\{\psi_1, \ldots \psi_n\} \models \phi$$

- is *satisfiable* when there is at least one interpretation for which the premises and the conclusion are true.
- is *falsifiable* when there is at least one interpretation for which the premises are true and the conclusion is false.

Logic

Logical Entailment (Validity) vs. Satisfiability

$$\{\psi_1, \ldots \psi_n\} \models \phi$$

- is *satisfiable* when there is at least one interpretation for which the premises and the conclusion are true.
- is *falsifiable* when there is at least one interpretation for which the premises are true and the conclusion is false.
- is *valid* when the set of interpretations for which the premises are true is included in the set of interpretations for which the conclusion is true (viz. the set of models of the premises are a subset of the set of models of the conclusion).

Logic

Logical Entailment (Validity) vs. Satisfiability

$$\{\psi_1, \ldots \psi_n\} \models \phi$$

- is *satisfiable* when there is at least one interpretation for which the premises and the conclusion are true.
- is *falsifiable* when there is at least one interpretation for which the premises are true and the conclusion is false.
- is *valid* when the set of interpretations for which the premises are true is included in the set of interpretations for which the conclusion is true (viz. the set of models of the premises are a subset of the set of models of the conclusion).

	p	q	$p \rightarrow q$
$\mathcal{I}_1$:	T	T	T
$\mathcal{I}_2$:	T	F	F
$\mathcal{I}_3$:	F	T	T
$\mathcal{I}_4$:	F	F	T

1. $\{p \rightarrow q, p\} \models q$ valid: $\{\mathcal{I}_1\} \subseteq \{\mathcal{I}_1, \mathcal{I}_3\}$
2. $\{p \rightarrow q, q\} \not\models p$ not valid: $\{\mathcal{I}_1, \mathcal{I}_3\} \not\subseteq \{\mathcal{I}_1, \mathcal{I}_2\}$

   it is satisfied by $\mathcal{I}_1$ and falsified by $\mathcal{I}_3$

Logic

FoL: Richer language

It quantifies over entities and expresses properties of entities, relations among entities. A formula is true or false in an interpretation $\mathcal{I}$: $(D, [\![\cdot]\!])$. The interpretation for which the formula is true are called a *model*.
Given the domain $D = \{1, 2, 3, \ldots\}$

$$
\begin{array}{lcll}
[\![\, \mathtt{a} \,]\!] & = & 1 \qquad [\![\mathtt{b}\,]\!] = 2 & \\
[\![\mathtt{E}\,]\!] & = & \{2, 4, 6 \ldots\} & \\
[\![\mathtt{B}\,]\!] & = & \{(1, 2), (2, 3) \ldots\} & \\
\mathcal{I} & \models & \mathtt{E(b)} & \text{since } [\![\mathtt{b}]\!] \in [\![E]\!] \\
\mathcal{I} & \not\models & \mathtt{B(b,a)} & \text{since } ([\![\mathtt{b}]\!], [\![\mathtt{a}]\!]) \not\in [\![B]\!] \\
\mathcal{I} & \models & \exists x.\mathtt{E}(x) & \\
\mathcal{I} & \models & \forall y.\exists x.\mathtt{B}(y, x) &
\end{array}
$$

Logic

FoL: Richer language

It quantifies over entities and expresses properties of entities, relations among entities. A formula is true or false in an interpretation $\mathcal{I}$: $(D, [\![\cdot]\!])$. The interpretation for which the formula is true are called a *model*. Given the domain $D = \{1, 2, 3, \ldots\}$

$$\begin{array}{lcll}
[\![\,\mathtt{a}\,]\!] & = & 1 \qquad [\![\mathtt{b}\,]\!] = 2 & \\
[\![\mathtt{E}\,]\!] & = & \{2, 4, 6 \ldots\} & \\
[\![\mathtt{B}\,]\!] & = & \{(1, 2), (2, 3) \ldots\} & \\
\mathcal{I} & \models & \mathtt{E(b)} & \text{since} \quad [\![\mathtt{b}]\!] \in [\![E]\!] \\
\mathcal{I} & \not\models & \mathtt{B(b,a)} & \text{since} \quad ([\![\mathtt{b}]\!], [\![\mathtt{a}]\!]) \not\in [\![B]\!] \\
\mathcal{I} & \models & \exists x.\mathtt{E}(x) & \\
\mathcal{I} & \models & \forall y.\exists x.\mathtt{B}(y, x) &
\end{array}$$

$$\{\psi_1, \ldots \psi_n\} \models \phi$$

Again, the entailment is *valid*, if the set of models of the premises is included in the set of models of the conclusion. *Satisfiable* if there is at least one model of the premises that is also a model of the conclusion.

Logic

FoL: logical entailment

An example of *logical entailment* in FoL is:

$$\exists x.\forall y.R(x,y) \models \forall y.\exists x.R(x,y)$$

all models of $\exists x.\forall y.R(x,y)$ are models of $\forall y.\exists x.R(x,y)$.

Whereas:

$$\forall y.\exists x.R(x,y) \not\models \exists x.\forall y.R(x,y)$$

- satisfiable: $\mathcal{I}_1 : D = \{a\}, [\![R]\!] = \{(a,a)\}$

$$\mathcal{I}_1 \models \forall y.\exists x.R(x,y) \text{ and } \mathcal{I}_1 \models \exists x.\forall y.R(x,y)$$

Logic

FoL: logical entailment

An example of *logical entailment* in FoL is:

$$\exists x.\forall y.R(x,y) \models \forall y.\exists x.R(x,y)$$

all models of $\exists x.\forall y.R(x,y)$ are models of $\forall y.\exists x.R(x,y)$.

Whereas:

$$\forall y.\exists x.R(x,y) \not\models \exists x.\forall y.R(x,y)$$

- satisfiable: $\mathcal{I}_1 : D = \{a\}, [\![R]\!] = \{(a,a)\}$

$$\mathcal{I}_1 \models \forall y.\exists x.R(x,y) \text{ and } \mathcal{I}_1 \models \exists x.\forall y.R(x,y)$$

- falsifiable: $\mathcal{I}_2 : D = \{a,b\}, [\![R]\!] = \{(a,a),(b,b)\}$

Logic

FoL: logical entailment

An example of *logical entailment* in FoL is:

$$\exists x.\forall y.R(x,y) \models \forall y.\exists x.R(x,y)$$

all models of $\exists x.\forall y.R(x,y)$ are models of $\forall y.\exists x.R(x,y)$.

Whereas:

$$\forall y.\exists x.R(x,y) \not\models \exists x.\forall y.R(x,y)$$

- satisfiable: $\mathcal{I}_1 : D = \{a\}, [\![R]\!] = \{(a,a)\}$

  $$\mathcal{I}_1 \models \forall y.\exists x.R(x,y) \text{ and } \mathcal{I}_1 \models \exists x.\forall y.R(x,y)$$

- falsifiable: $\mathcal{I}_2 : D = \{a,b\}, [\![R]\!] = \{(a,a),(b,b)\}$

  $$\mathcal{I}_2 \models \forall y.\exists x.R(x,y) \text{ and } \mathcal{I}_2 \not\models \exists x.\forall y.R(x,y)$$

Layout

What's the meaning of linguistic signs?

Logic view: Reference

There is the star a called "venus", "morning star", "evening star" that are represented in FoL by $venus'$, $morningst'$, $eveningst'$:

$$[\![venus']\!] = a$$
$$[\![morningst']\!] = a$$
$$[\![eveningst']\!] = a$$

a is the meaning (*reference*) of these linguistic signs.

What's the meaning of linguistic signs?

Logic view: Reference

There is the star a called "venus", "morning star", "evening star" that are represented in FoL by $venus'$, $morningst'$, $eveningst'$:

$$[\![venus']\!] = a$$
$$[\![morningst']\!] = a$$
$$[\![eveningst']\!] = a$$

a is the meaning (*reference*) of these linguistic signs.

Checking whether it is true that (i) "the morning star is the morning star" or that (ii) "the morning star is the evening star" ends up checking that

(i) $[\![morningst']\!] = [\![morningst']\!]$ and (ii) $[\![morningst']\!] = [\![eveningst']\!]$

Both of which reduce to checking

$$a = a$$

What's the meaning of linguistic signs?

Bedeutung vs. Sinn

checking whether (i) "the morning star is the morning star" or that (ii) "the morning star is the evening star" can't amount to the same operation since (ii) is cognitively more difficult than (i).

What's the meaning of linguistic signs?

checking whether (i) "the morning star is the morning star" or that (ii) "the morning star is the evening star" can't amount to the same operation since (ii) is cognitively more difficult than (i).

Frege's answer: A linguistic sign consists of a:

- **Bedeutung**: the object that the expression refers to
- **Sinn**: mode of presentation of the referent.

What's the meaning of linguistic signs?

Bedeutung vs. Sinn

checking whether (i) "the morning star is the morning star" or that (ii) "the morning star is the evening star" can't amount to the same operation since (ii) is cognitively more difficult than (i).

Frege's answer: A linguistic sign consists of a:

- **Bedeutung**: the object that the expression refers to
- **Sinn**: mode of presentation of the referent.

Two schools of thought

- **Formal Semantics**: meaning based on references.
- **Distributional Semantics/Language as use**: meaning based on the words' context (use).

Layout

Formal Semantics

Starting point: meaning based on reference

Starting point: set-theoretical view of meaning and the assumption that the meaning of a proposition is its truth value.
Eg. take the interpretation $\mathcal{I}$: $D = \{$sara, lori, pim, alex$\}$ and :

$$
\begin{array}{lcl}
[\![\texttt{sara}']\!] & = & \text{sara}; \ldots \\
[\![\texttt{walk}']\!] & = & \{\text{lori}\}; \\
[\![\texttt{know}']\!] & = & \{(\text{lori, alex}), (\text{alex,lori}), (\text{sara, lori}), \\
 & & (\text{lori, lori}), (\text{alex, alex}), (\text{sara, sara}), (\text{pim, pim})\}; \\
[\![\texttt{student}']\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\texttt{professor}']\!] & = & \{\}; \\
[\![\texttt{tall}']\!] & = & \{\text{lori, pim}\}.
\end{array}
$$

$\mathcal{I} \models walks'(lori')$ $\mathcal{I} \models \exists x.student'(x) \wedge walk'(x)$
$\mathcal{I} \not\models walks'(sara')$ $\mathcal{I} \not\models \forall x.student(x) \rightarrow walk'(x)$

FS aim: To obtain these FoL representations compositionaly. Hence, questions: What is the meaning representation of the lexical words? Which operation(s) put the lexical meaning representation together.

Formal Semantics

Syntax guides function application

Order of composition Syntax gives the order of composition:

- Lori [knows Alex]
- [A student] [knows Alex]
- [A [student [who [Alex [knows [. . .]]]]]$_N$]$_{DP}$ [studies Logic].

From sets to functions A set X and *its characteristic function* f_X amount to the same thing. In other words, the assertion $y \in X$ and $f_X(y) = \mathit{true}$ are equivalent.

Formal Semantics

Syntax guides function application

Order of composition Syntax gives the order of composition:

- Lori [knows Alex]
- [A student] [knows Alex]
- [A [student [who [Alex [knows [...]]]]]$_N$]$_{DP}$ [studies Logic].

From sets to functions A set X and *its characteristic function* f_X amount to the same thing. In other words, the assertion $y \in X$ and $f_X(y) = true$ are equivalent.
Hence, since the meaning of the word "student" is a set of objects:

$$[\![\texttt{student}']\!] = \{\text{lori, alex, sara}\}$$

Formal Semantics

Syntax guides function application

Order of composition Syntax gives the order of composition:

- Lori [knows Alex]
- [A student] [knows Alex]
- [A [student [who [Alex [knows [...]]]]]$_N$]$_{DP}$ [studies Logic].

From sets to functions A set X and *its characteristic function* f_X amount to the same thing. In other words, the assertion $y \in X$ and $f_X(y) = true$ are equivalent.
Hence, since the meaning of the word "student" is a set of objects:

$$[\![\texttt{student}']\!] = \{\text{lori, alex, sara}\}$$

it can be represented as a function from entities to truth values

$$[\![\lambda x.\texttt{student}'(x)]\!] = \{x | \texttt{student}'(x) = T\}$$

Formal Semantics

Syntax guides function application

Order of composition Syntax gives the order of composition:

- Lori [knows Alex]
- [A student] [knows Alex]
- [A [student [who [Alex [knows [...]]]]]$_N$]$_{DP}$ [studies Logic].

From sets to functions A set X and *its characteristic function* f_X amount to the same thing. In other words, the assertion $y \in X$ and $f_X(y) = true$ are equivalent.
Hence, since the meaning of the word "student" is a set of objects:

$$[\![\mathtt{student}']\!] = \{\text{lori, alex, sara}\}$$

it can be represented as a function from entities to truth values

$$[\![\lambda x.\mathtt{student}'(x)]\!] = \{x | \mathtt{student}'(x) = T\}$$

Frege/Montague Starting from lexical representations we reach FoL by function application following the syntactic structure.

Formal Semantics

Many domain of denotations

Not only one domain. Now we care of the meaning of words/phrases. Words/Phrases denote in different domains:

- S (sentences): domain of truth values: D_t
- PN (e.g. Lori): domain of entities: D_e

Formal Semantics

Many domain of denotations

Not only one domain. Now we care of the meaning of words/phrases. Words/Phrases denote in different domains:

- S (sentences): domain of truth values: D_t
- PN (e.g. Lori): domain of entities: D_e
- N (e.g. student, tall student, student who Alex knows) and VP (e.g. walk, knows lori):
  domain of sets of entities, ie. domain of functions: $D_e \to D_t$.
- TV: (e.g. knows): domain of sets of pairs, i.e. domain of functions: $(D_e \times D_e) \to D_t$ $(= D_e \to (D_e \to D_t))$

E.g. "walk" and "knows lori" meaning is a set of entities (those entities who walks, those entities who know lori). Hence, they

- denote in the domain $D_e \to D_t$
- are of semantic type $(e \to t)$,
- are represented by the terms
  $\lambda x_e.(\mathtt{walk}'(x))_t$ and $\lambda x_e.(\mathtt{knows}'(x, lori'))_t$

Given $\mathcal{I}$ (i.e. typed domains and a $[\![\cdot]\!]$):

- D_t: $[\![\phi]\!] \leq_t [\![\psi]\!]$ iff $\mathcal{I}$ satisfies $\phi \models \psi$.
- $D_{(a \to b)}$: $[\![X]\!] \leq_{(a \to b)} [\![Y]\!]$ iff $\forall [\![\alpha]\!] \in D_a \;\; [\![X(\alpha)]\!] \leq_b [\![Y(\alpha)]\!]$

Take the set-theoretical view:

- N, VP: inclusion among sets of entities.
- TV: inclusion among sets of pairs

Formal Semantics

Lexical entailment (partially ordered domains)

Given the interpretation: $D_e = \{\text{lori, alex, sara}\}$, $[\![\texttt{walk'}]\!] = \{lori\}$, $[\![\texttt{move'}]\!] = \{\text{lori,alex}\}$

$[\![\texttt{walk'}]\!] \leq_{(e \to t)} [\![\texttt{move'}]\!]$ iff $\forall [\![x]\!] \in D_e, [\![\texttt{walk'}(x)]\!] \leq_t [\![\texttt{move'}(x)]\!]$

- $F \leq_t T$ for $[\![x]\!]=$ alex
- $T \leq_t T$ for $[\![x]\!]=$ lori
- $F \leq_t F$ for $[\![x]\!]=$ sara

Formal Semantics

Set-theoretical meaning of “some”

What is the set-theoretical meaning of “Some student”

Formal Semantics

Set-theoretical meaning of "some"

What is the set-theoretical meaning of "Some student" It's not an object, it's not a set of objects, it's not a set of pairs of objects.

Formal Semantics

Set-theoretical meaning of “some”

What is the set-theoretical meaning of “Some student” It’s not an object, it’s not a set of objects, it’s not a set of pairs of objects.
Take the interpretation $\mathcal{I}$: $D = \{\text{sara, lori, pim, alex}\}$ and :

$$
\begin{array}{lcl}
[\![\texttt{sara}']\!] & = & \text{sara}; \ldots \\
[\![\texttt{walk}']\!] & = & \{\text{lori}\}; \\
[\![\texttt{know}']\!] & = & \{(\text{lori, alex}), (\text{alex,lori}), (\text{sara, lori}), \\
 & & (\text{lori, lori}), (\text{alex, alex}), (\text{sara, sara}), (\text{pim, pim})\}; \\
[\![\texttt{student}']\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\texttt{professor}']\!] & = & \{\}; \\
[\![\texttt{tall}']\!] & = & \{\text{lori, pim}\}.
\end{array}
$$

In this interpretation, “some student are tall”, “some student walk”.

Formal Semantics

Set-theoretical meaning of “some”

What is the set-theoretical meaning of “Some student” It’s not an object, it’s not a set of objects, it’s not a set of pairs of objects.
Take the interpretation $\mathcal{I}$: $D = \{\text{sara, lori, pim, alex}\}$ and :

$$
\begin{array}{lcl}
[\![\mathtt{sara}']\!] & = & \text{sara}; \ldots \\
[\![\mathtt{walk}']\!] & = & \{\text{lori}\}; \\
[\![\mathtt{know}']\!] & = & \{(\text{lori, alex}), (\text{alex,lori}), (\text{sara, lori}), \\
 & & (\text{lori, lori}), (\text{alex, alex}), (\text{sara, sara}), (\text{pim, pim})\}; \\
[\![\mathtt{student}']\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\mathtt{professor}']\!] & = & \{\}; \\
[\![\mathtt{tall}']\!] & = & \{\text{lori, pim}\}.
\end{array}
$$

In this interpretation, “some student are tall”, “some student walk”.

$$
\begin{array}{lcl}
[\![\mathtt{some}'\ (\mathtt{student}')]\!] & = & \{[\![\mathtt{tall}']\!], [\![\mathtt{walk}']\!]\} \\
 & = & \{Y \mid [\![\mathtt{student}']\!] \cap Y \neq \varnothing\}
\end{array}
$$

Formal Semantics

Set-theoretical meaning of "some"

What is the set-theoretical meaning of "Some student" It's not an object, it's not a set of objects, it's not a set of pairs of objects.
Take the interpretation $\mathcal{I}$: $D = \{\text{sara, lori, pim, alex}\}$ and :

$$
\begin{array}{lcl}
[\![\texttt{sara}']\!] & = & \text{sara}; \ldots \\
[\![\texttt{walk}']\!] & = & \{\text{lori}\}; \\
[\![\texttt{know}']\!] & = & \{(\text{lori, alex}), (\text{alex,lori}), (\text{sara, lori}), \\
 & & (\text{lori, lori}), (\text{alex, alex}), (\text{sara, sara}), (\text{pim, pim})\}; \\
[\![\texttt{student}']\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\texttt{professor}']\!] & = & \{\}; \\
[\![\texttt{tall}']\!] & = & \{\text{lori, pim}\}.
\end{array}
$$

In this interpretation, "some student are tall", "some student walk".

$$
\begin{array}{lcl}
[\![\texttt{some}'\ (\texttt{student}')]\!] & = & \{[\![\texttt{tall}']\!], [\![\texttt{walk}']\!]\} \\
 & = & \{Y | [\![\texttt{student}']\!] \cap Y \neq \varnothing\}
\end{array}
$$

As a function: $\text{D}_{(e \to t) \to t}$. $\lambda Y.\exists x.\texttt{student}'(x) \wedge Y(x)$:

Formal Semantics

Set-theoretical meaning of Quantifiers

$$
\begin{array}{lcl}
[\![\texttt{some}'(\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \cap Y \neq \varnothing\} \\
[\![\texttt{no}'\ (\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \cap Y = \varnothing\} \\
[\![\texttt{every}'\ (\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \subseteq Y\}
\end{array}
$$

Formal Semantics

Set-theoretical meaning of Quantifiers

$$\begin{array}{lcl} [\![\texttt{some}'(\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \cap Y \neq \varnothing\} \\ [\![\texttt{no}'\ (\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \cap Y = \varnothing\} \\ [\![\texttt{every}'\ (\texttt{student}')]\!] & = & \{Y | [\![\texttt{student}']\!] \subseteq Y\} \end{array}$$

All QP denote in the domains $D_{(e \to t) \to t}$. They are represented by the lambda terms below

- $\lambda Y.\exists x.Student'(x) \wedge Y(x)$:
- $\lambda Y.\neg\exists x.(Student'(x) \wedge Y(x))$:
- $\lambda Y.\forall x.Student'(x) \to Y(x)$:

Hence, the quantifiers denote in $D_{(e \to t) \to ((e \to t) \to t)}$:

- Some: $\lambda Z.\lambda Y.\exists x.Z(x) \wedge Y(x)$
- No: $\lambda Z.\lambda Y.\neg\exists x.(Z(x) \wedge Y(x))$
- Every: $\lambda Z.\lambda Y.\forall x.Z(x) \to Y(x)$

Formal Semantics

Relative pronoun

$$
\begin{array}{lcl}
[\![\texttt{lori}']\!] & = & \text{lori}; \ldots \\
[\![\lambda x.\texttt{student}'(x)]\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\lambda x.\lambda y.\texttt{know}'(y,x)]\!] & = & \{(\text{lori, alex}), (\text{lori, pim}), (\text{sara, alex})\} \\
[\![\lambda \texttt{y}.\texttt{know}'(\texttt{y}, \texttt{alex}')]\!] & = & \{\text{lori, sara}\} \\
[\![\lambda x.\texttt{know}'(\texttt{lori}', \texttt{x})]\!] & = & \{\text{alex, pim}\}
\end{array}
$$

The meaning of "student who lori knows" is a set of entities: {alex}.

"who" creates the intersection between the set of students and the set of those people who lori knows:

$$[\![N\ who\ VP]\!] = [\![\texttt{N}]\!] \cap [\![\texttt{VP}]\!]$$

Formal Semantics

Relative pronoun

$$
\begin{array}{lcl}
[\![\mathtt{lori}']\!] & = & \text{lori}; \ldots \\
[\![\lambda x.\mathtt{student}'(x)]\!] & = & \{\text{lori, alex, sara}\}; \\
[\![\lambda x.\lambda y.\mathtt{know}'(y, x)]\!] & = & \{(\text{lori, alex}), (\text{lori, pim}), (\text{sara, alex})\} \\
[\![\lambda \mathtt{y}.\mathtt{know}'(\mathtt{y}, \mathtt{alex}')]\!] & = & \{\text{lori, sara}\} \\
[\![\lambda x.\mathtt{know}'(\mathtt{lori}', \mathtt{x})]\!] & = & \{\text{alex, pim}\}
\end{array}
$$

The meaning of "student who lori knows" is a set of entities: {alex}.

"who" creates the intersection between the set of students and the set of those people who lori knows:

$$[\![N \ who \ VP]\!] = [\![\mathtt{N}]\!] \cap [\![\mathtt{VP}]\!]$$

"who": $\lambda VP.\lambda N.\lambda x.N(x) \wedge VP(x)$

To build the meaning representation of linguistic structure, besides function application, we need abstraction:

1. knows z: $\lambda y.\texttt{knows}'(y, z)$
2. lori [knows z]: $\texttt{knows}'(\texttt{lori}', z)$

To build the meaning representation of linguistic structure, besides function application, we need abstraction:

1. knows z: $\lambda y.\texttt{knows}'(y, z)$
2. lori [knows z]: $\texttt{knows}'(\texttt{lori}', z)$
3. lori knows: $\lambda z.\texttt{knows}'(\texttt{lori}', z)$

Formal Semantics

Abstraction

To build the meaning representation of linguistic structure, besides function application, we need abstraction:

1. knows z: $\lambda y.\mathtt{knows}'(y, z)$
2. lori [knows z]: $\mathtt{knows}'(\mathtt{lori}', z)$
3. lori knows: $\lambda z.\mathtt{knows}'(\mathtt{lori}', z)$
4. who lori knows: $\lambda N.\lambda x.N(x) \wedge \mathtt{knows}'(\mathtt{lori}, x)$
5. student who lori knows: $\lambda x.\mathtt{student}'(x) \wedge \mathtt{knows}'(\mathtt{lori}, x)$

Abstraction is caused by "who": $\lambda VP.\lambda N.\lambda x.N(x) \wedge VP(x)$