FLST: Semantics II

Vera Demberg
vera@coli.uni-sb.de

http://www.coli.uni-saarland.de/courses/FLST/2011/

Semantics: The Logical Paradigm

- Validation of semantic representations via truth-conditional interpretation
- Semantically controlled inference through entailment and deduction
- A rigid model of compositionality

Deduction: A Question Answering Example

- Question: *Which element is Thallium said to look like?*
- Support passage: *Thallium is a metallic element that resembles lead.*
- Answer: *Lead*

Watson Again

- We show that "lead" is a correct answer by deriving the representation of the question instantiated with "lead" (the "conclusion" or "hypothesis") from the representation of the answer passage (the "premiss").

- Given:
  - metallic(thallium) ∧ element(thallium) ∧ resemble(thallium, lead)

- Wanted:
  - element(lead) ∧ look_like(thallium, lead)

More Ingredients for the Derivation

- We need some more deduction rules. These are justified by corresponding entailment relations: truth preserving transition from premises to conclusion (please, check!).
  - $A \wedge B \vdash A$, $A \wedge B \vdash B$ (Conjunction Elimination)
  - $A, B \vdash A \wedge B$ (Conjunction Introduction)
  - $A, A \rightarrow B \vdash B$ (Modus Ponens)
  - $A \leftrightarrow B \vdash A \rightarrow B$, $A \leftrightarrow B \vdash B \rightarrow A$ (Equivalence Elimination)
  - $\forall x A \vdash A[b/x]$ (Universal Instantiation)

- We need some extra bits of knowledge (axioms, taken e.g. from a lexical-semantic knowledge base):
  - element(lead)
  - ∀x∀y(resemble(x,y) ↔ look_like(x,y))

Example Derivation

(1)	metallic(th) ∧ element(th) ∧ resemble(th, lead)	Premise
(2)	resemble(th, lead)	2x Conjunction Elim (1)
(3)	∀x∀y(resemble(x,y) ↔look_like(x,y))	Axiom
(4)	∀y(resemble(th,y) ↔look_like(th,y))	Univ. Instantiation th/x, (3)
(5)	resemble(th,lead) ↔look_like(th,lead)	Univ. Instantiation lead/y, (4)
(6)	resemble(th,lead) →look_like(th,lead)	Equivalence Elim, (5)
(7)	look_like(th,lead)	Modus Ponens (2), (6)
(8)	element(lead)	Axiom
(9)	element(lead) ∧ look_like(th, lead)	Conjunction Intro (7), (8)

Word Meaning in the Logical Paradigm

- Atomic predicates represent word senses, but are not very informative in themselves.

- Axioms express word-semantic information:
  - semantic relations between different words:

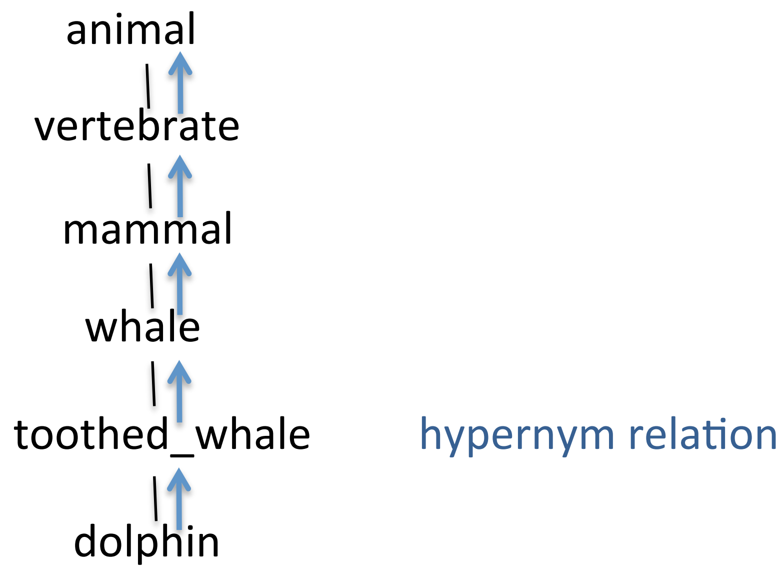
    $\forall x \forall y(\text{look\_like}(x, y) \leftrightarrow \text{resemble}(x, y))$
  - semantic properties of words:

    $\forall x \forall y(\text{resemble}(x, y) \rightarrow \text{resemble}(y, x))$

- Where can we get these axioms from???

→ Axioms can be read off lexical-semantic taxonomies like WordNet

WordNet Meaning Relations



Axioms Expressing Semantic Relations

- B hypernym of A ⇒ $\forall x(A(x) \rightarrow B(x))$

  $\forall x(dolphin(x) \rightarrow toothed\_whale(x))$

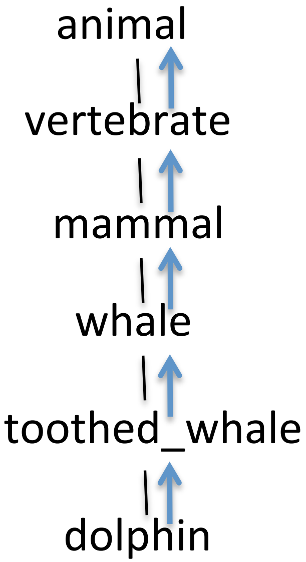
  $\forall x(toothed\_whale(x) \rightarrow whale(x))$

  $\forall x(whale(x) \rightarrow mammal(x))$

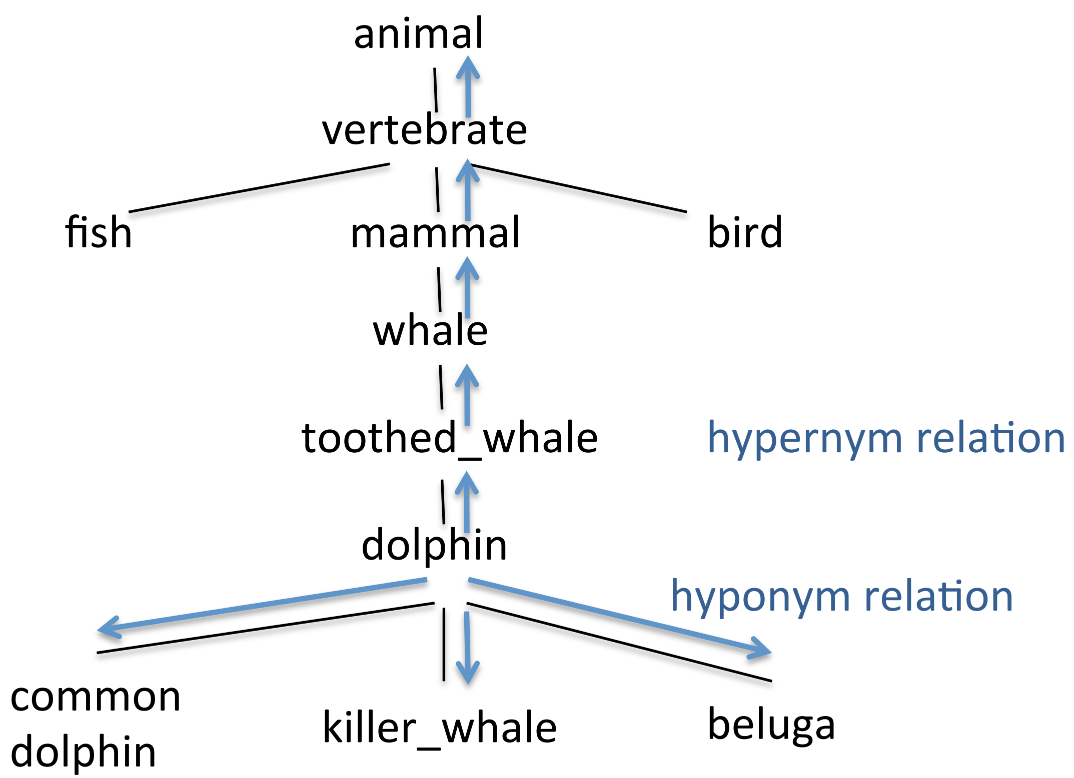
  $\forall x(mammal(x) \rightarrow vertebrate(x))$

  $\forall x(vertebrate(x) \rightarrow animal(x))$

hypernym relation



WordNet Meaning Relations



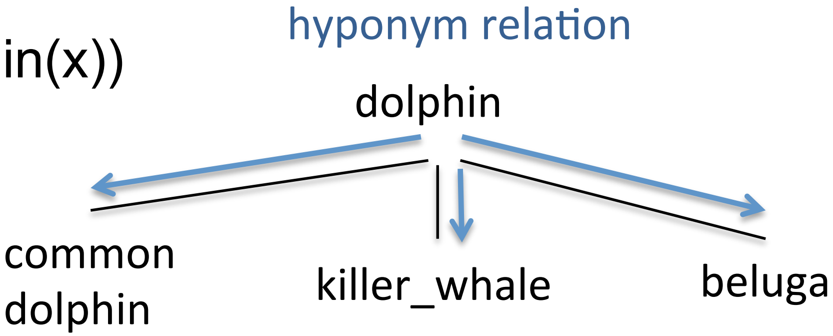
Axioms Expressing Semantic Relations

- B hypernym of A $\Rightarrow \forall x(A(x) \rightarrow B(x))$

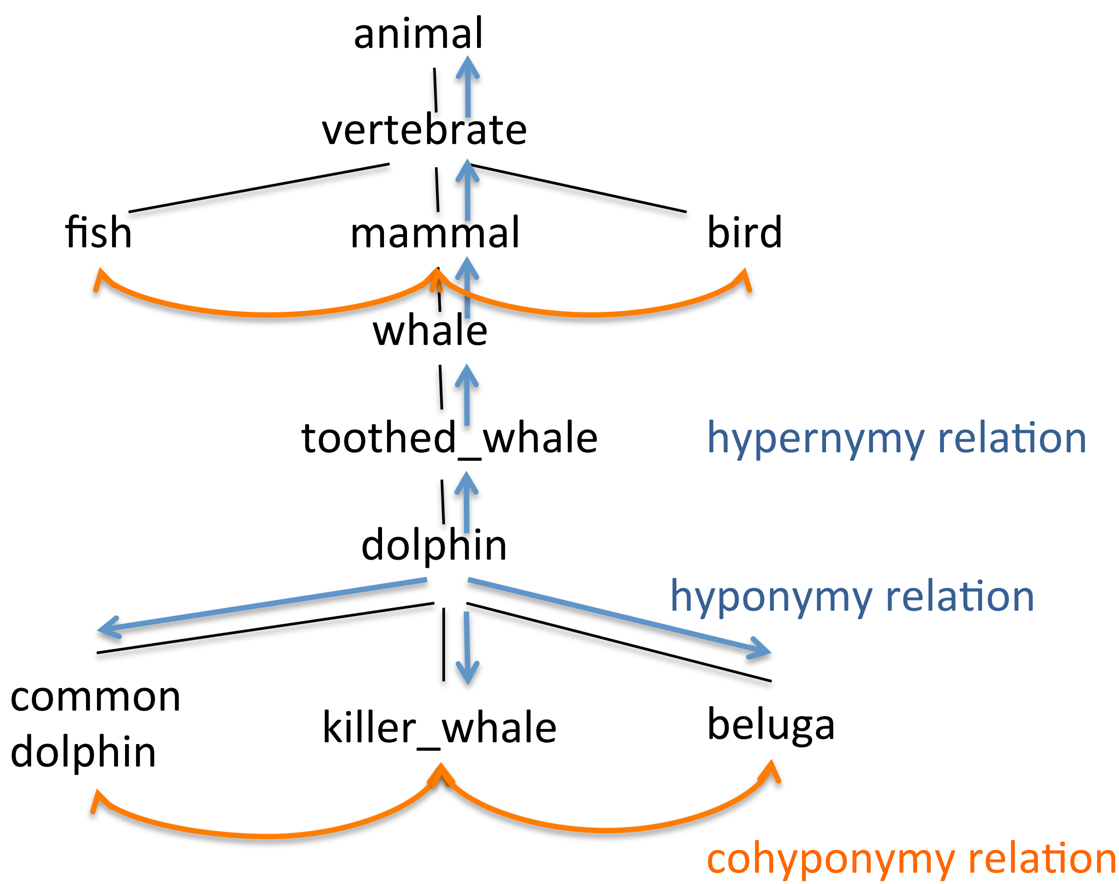
  $\forall x(dolphin(x) \rightarrow toothed\_whale(x))$
  $\forall x(toothed\_whale(x) \rightarrow whale(x))$
  $\forall x(whale(x) \rightarrow mammal(x))$
  $\forall x(mammal(x) \rightarrow vertebrate(x))$
  $\forall x(vertebrate(x) \rightarrow animal(x))$

- B hyponym of A $\Rightarrow \forall x(B(x) \rightarrow A(x))$

  $\forall x(common\_dolphin(x) \rightarrow dolphin(x))$
  $\forall x(killer\_whale(x) \rightarrow dolphin(x))$
  $\forall x(beluga(x) \rightarrow dolphin(x))$



WordNet Meaning Relations



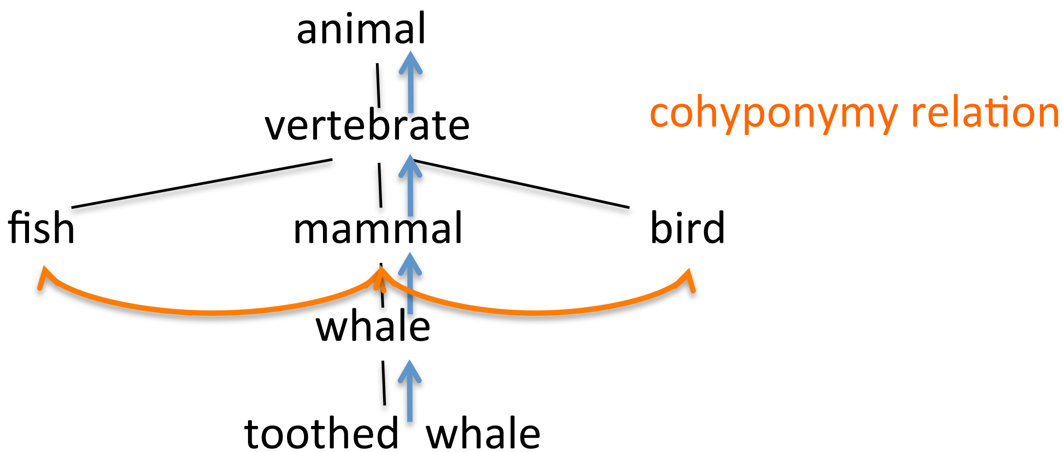
Axioms Expressing Semantic Relations

- A and B cohyponyms $\Rightarrow \forall x(A(x) \rightarrow \neg B(x))$

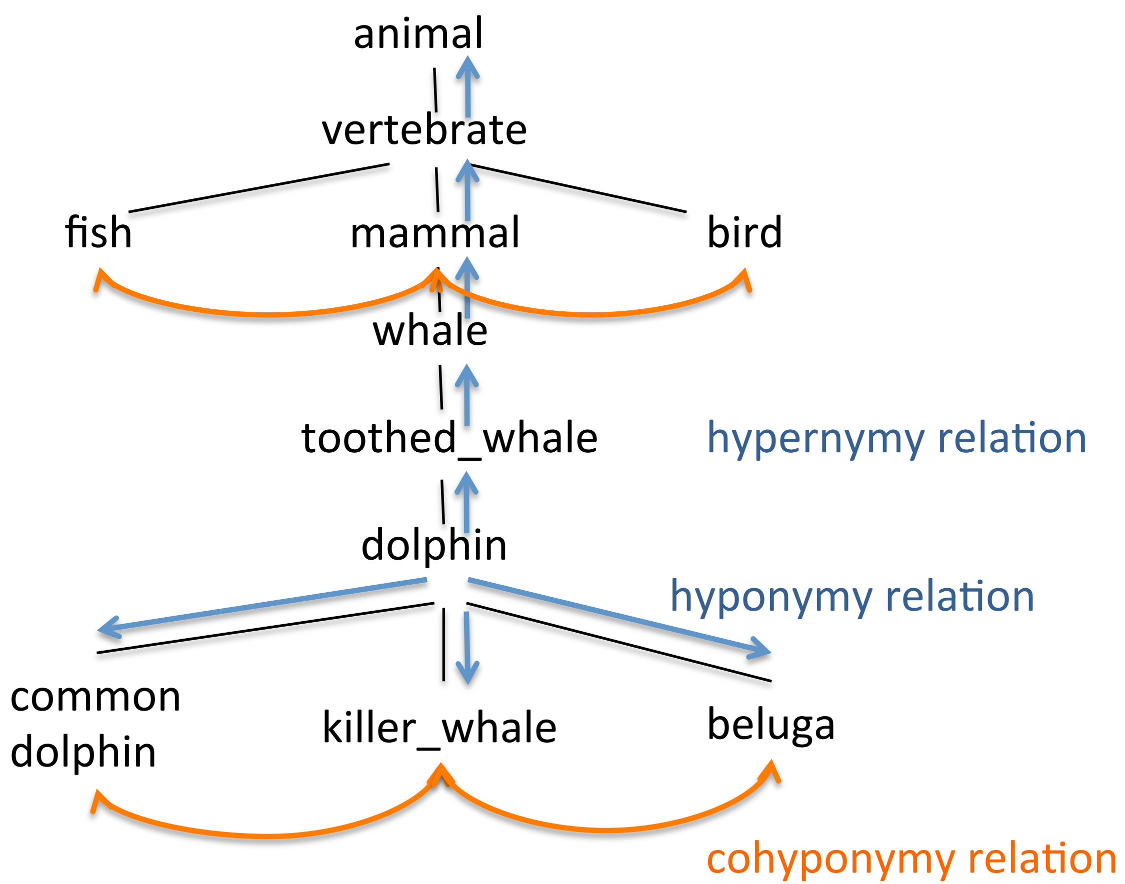
  $\forall x(mammal(x) \rightarrow \neg fish\ (x))$

  $\forall x(fish(x) \rightarrow \neg\ bird\ (x))$

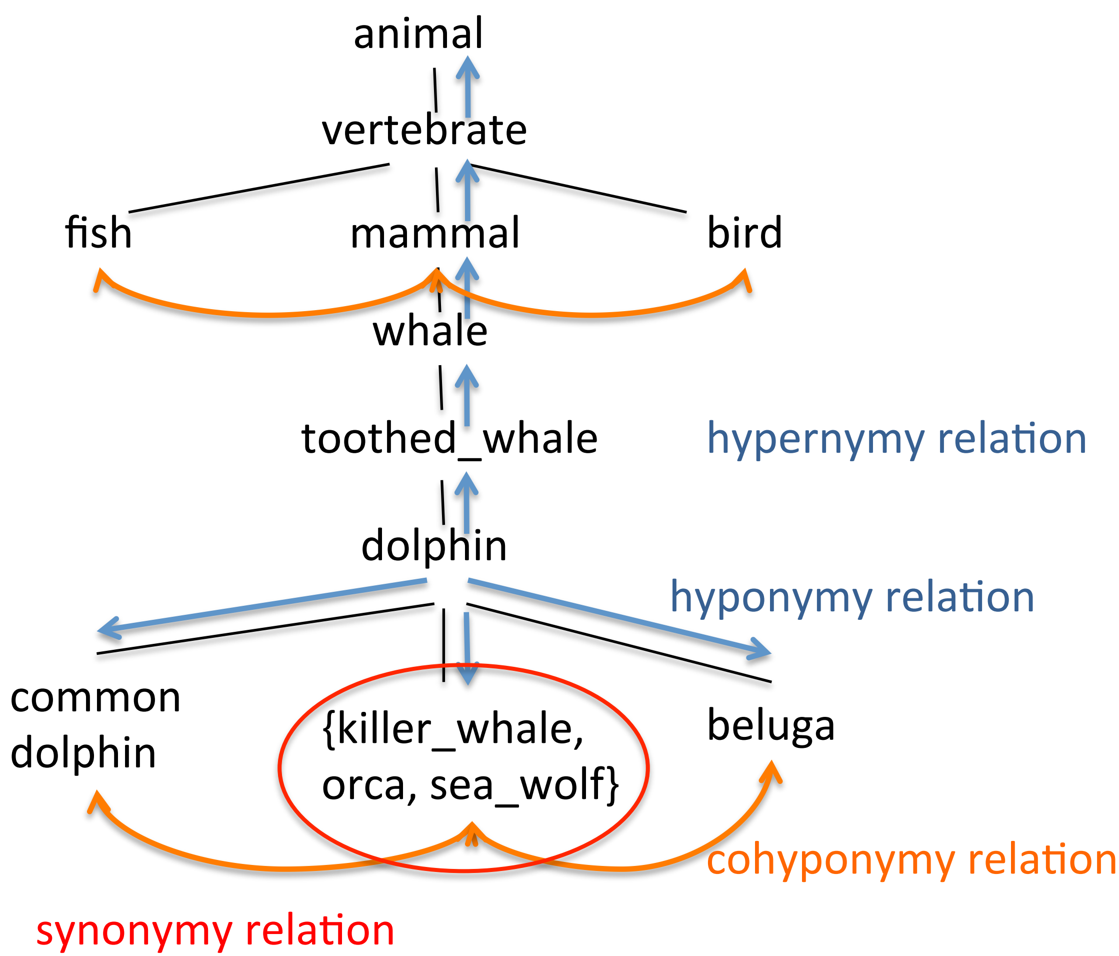
  $\forall x(bird(x) \rightarrow \neg\ mammal(x))$



WordNet Meaning Relations



WordNet Meaning Relations



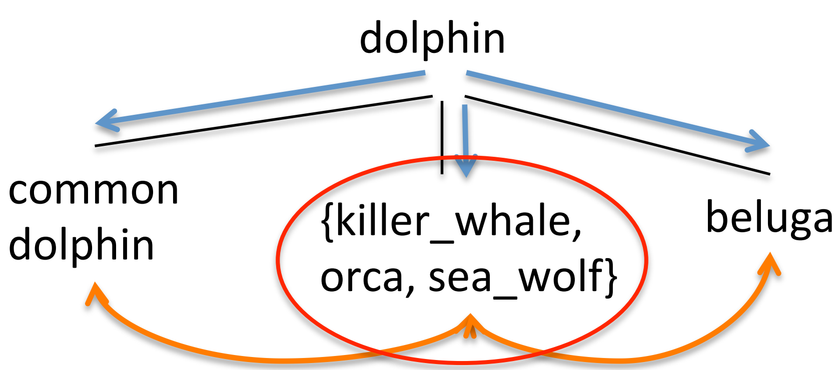
Axioms Expressing Semantic Relations

- A and B cohyponyms $\Rightarrow \forall x(A(x) \rightarrow \neg B(x))$

  $\forall x(\text{mammal}(x) \rightarrow \neg \text{fish}(x))$
  $\forall x(\text{fish}(x) \rightarrow \neg \text{bird}(x))$
  $\forall x(\text{bird}(x) \rightarrow \neg \text{mammal}(x))$

- A and B synonyms $\Rightarrow \forall x(A(x) \leftrightarrow B(x))$

  $\forall x(\text{killer\_whale}(x) \leftrightarrow \text{orca}(x))$
  $\forall x(\text{killer\_whale}(x) \leftrightarrow \text{sea\_wolf}(x))$



synonymy relation

Semantics: The Logical Paradigm

- Validation of semantic representations via truth-conditional interpretation
- Semantically controlled inference through entailment and deduction
- A rigid model of compositionality

Semantic Composition

Principle of Compositionality (Frege's Principle):

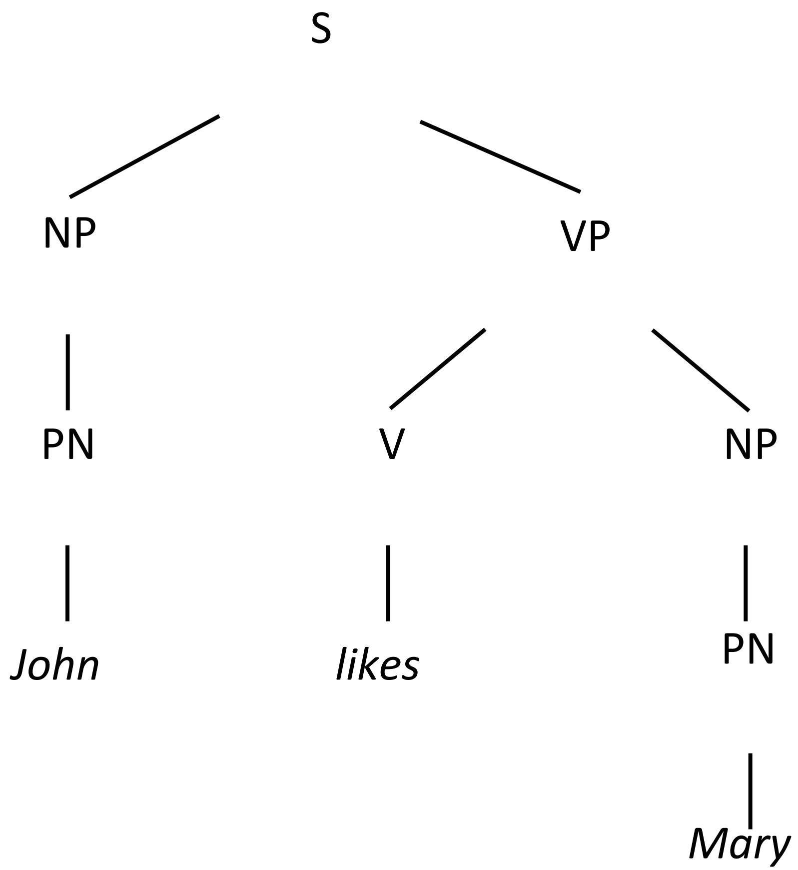
- The meaning of a complex expression is uniquely determined by the meanings of its sub-expressions and its syntactic structure.

Semantic Composition

- *John likes Mary* ⇒ like(john, mary)

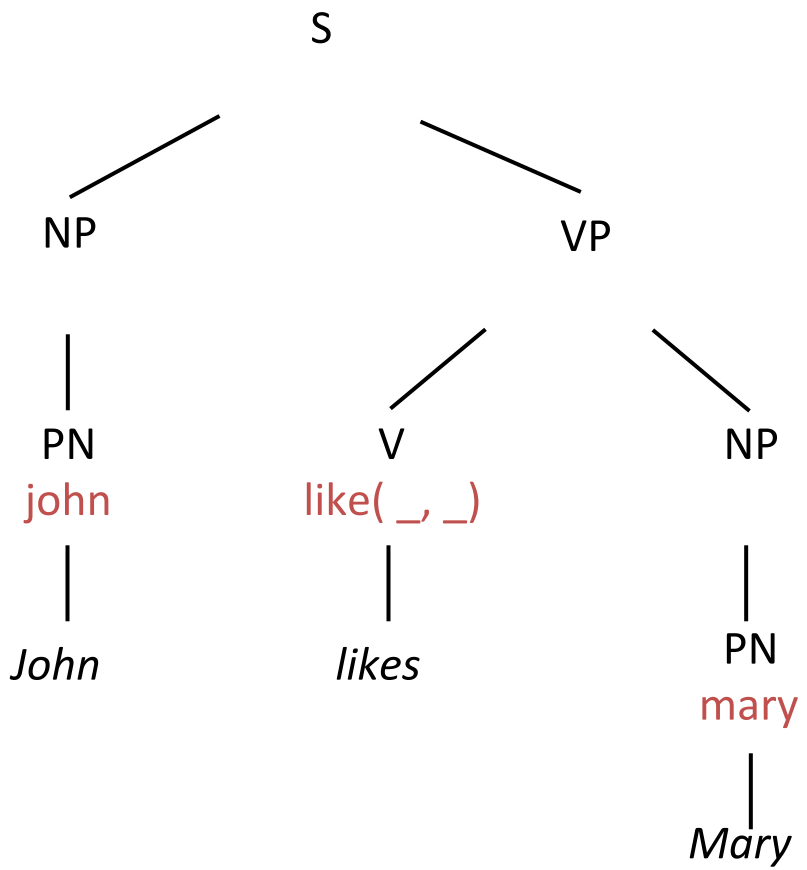
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



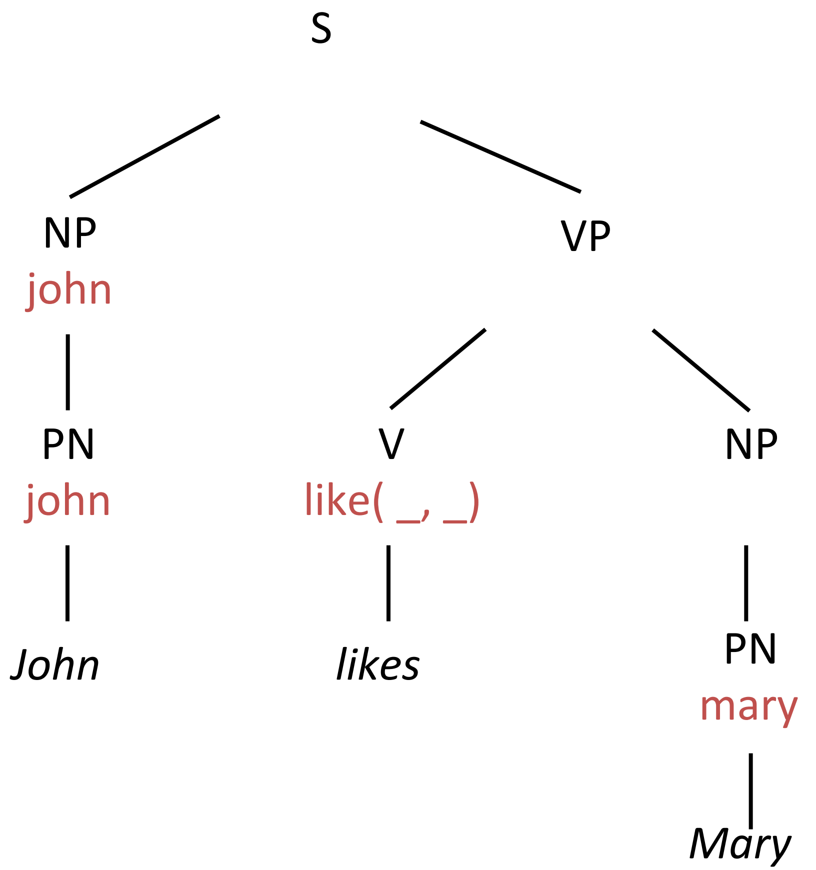
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



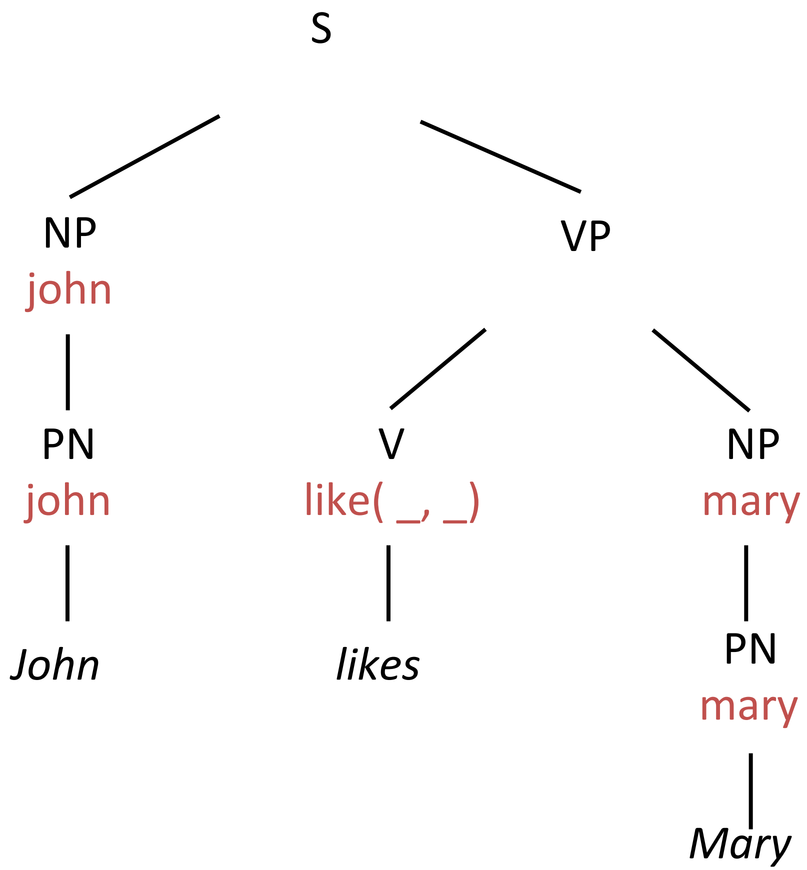
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



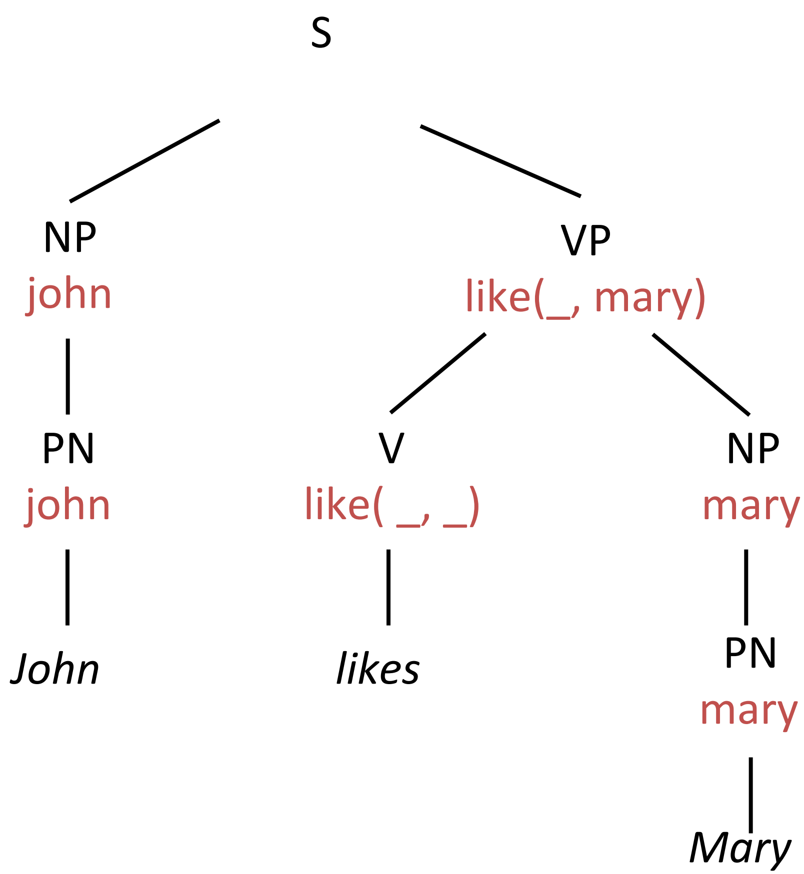
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



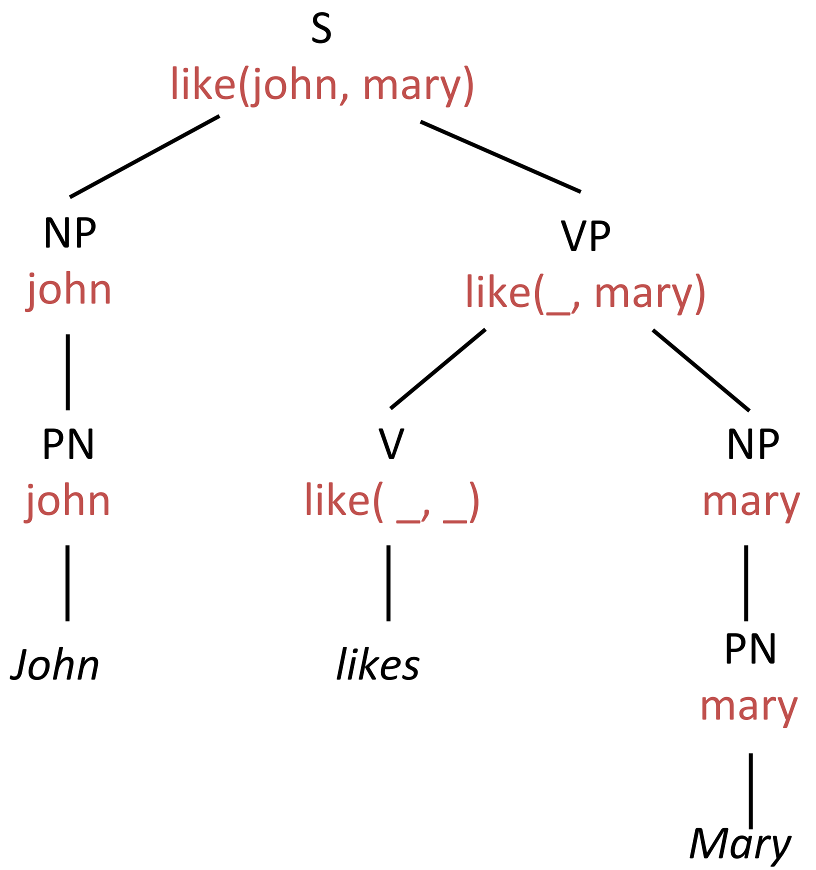
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



Semantic Composition

- How do meanings of syntactic complements find their appropriate argument positions in the composition process?

- The answer is: λ-Abstraction

Mary

λ-Abstraction

- student: a one-place predicate
- student(x): a formula containing a free variable
- λx[student(x)]: a one-place-predicate again: „to be a student“
- λx[student(x)](john): a formula: application of a one-place predicate (the λ-expression) to the individual constant "john“,
- which is equivalent to student(john)

Interpretation of λ-expressions

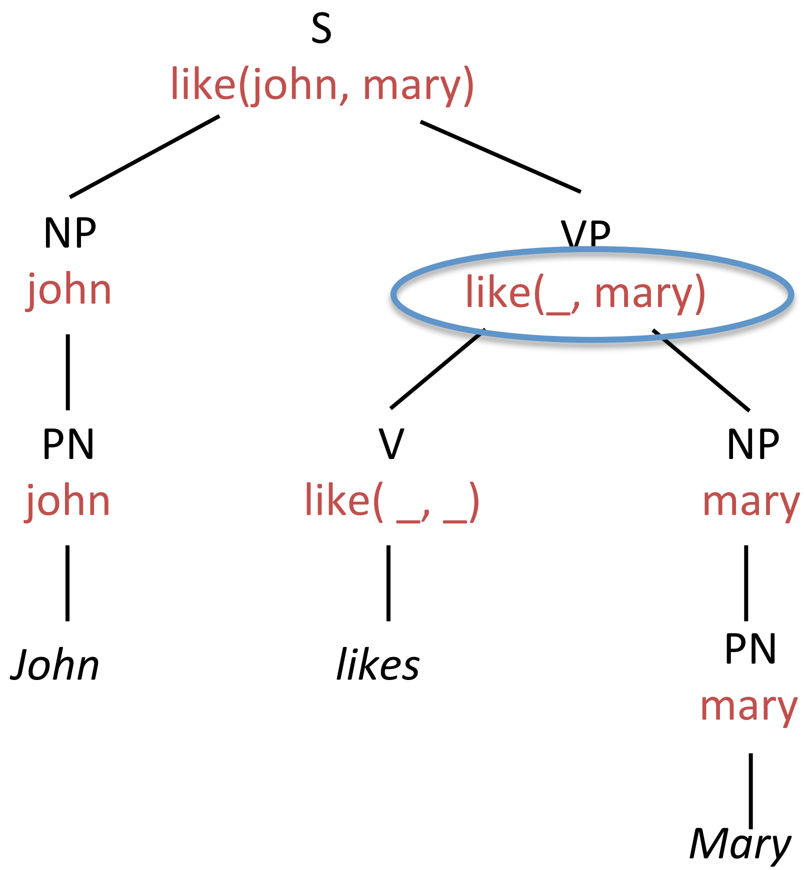
- $[\![\lambda x A]\!]^{M,g} = \{a \in U_M | [\![A]\!]^{M,g[x/a]} = 1\}$

- $[\![\lambda x[student(x)]]\!]^{M,g} = \{a \in U_M | [\![student(x)]\!]^{M,g[x/a]} = 1\}$
  $= \{a \in U_M | a \in V_M(student)\}$
  i.e., the set of individuals who are students, that is $V_M(student)$

- $[\![\lambda x[like(x, mary)]]\!]^{M,g} = \{a \in U_M | [\![like(x, mary)]\!]^{M,g[x/a]} = 1\}$
  $= \{a \in U_M | \langle a, V_M(mary)\rangle \in V_M(like)\}$,
  i.e., the set of individuals who like Mary.
  This is not necessarily identical to the denotation of any predicate constant.

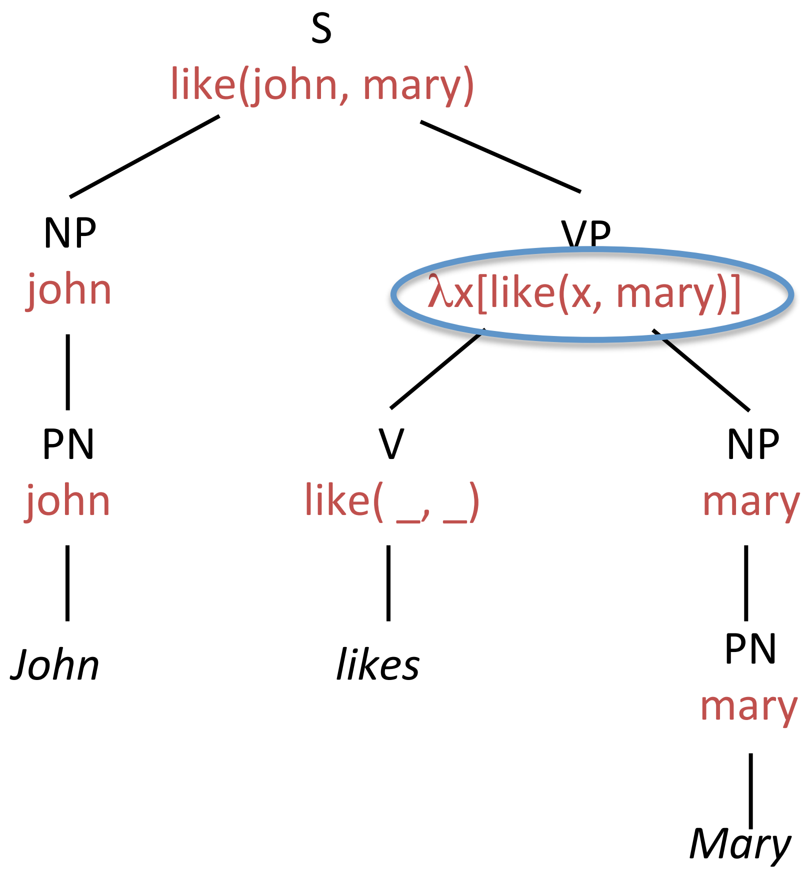
Semantic Composition

- ❑ *John likes Mary* ⇒ like(john, mary)



Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



Application of λ-Expressions

John ⇒ john

likes Mary ⇒ $\lambda x[\text{like}(x, \text{mary})]$

John likes Mary ⇒ $\lambda x[\text{like}(x, \text{mary})](\text{john})$

⇔ like(john, mary)

$[\![\lambda x[\text{like}(x, \text{mary})](\text{john})]\!]^{M,g} = 1$

iff $[\![\text{john}]\!]^{M,g} \in [\![\lambda x[\text{like}(x, \text{mary})]]\!]^{M,g}$

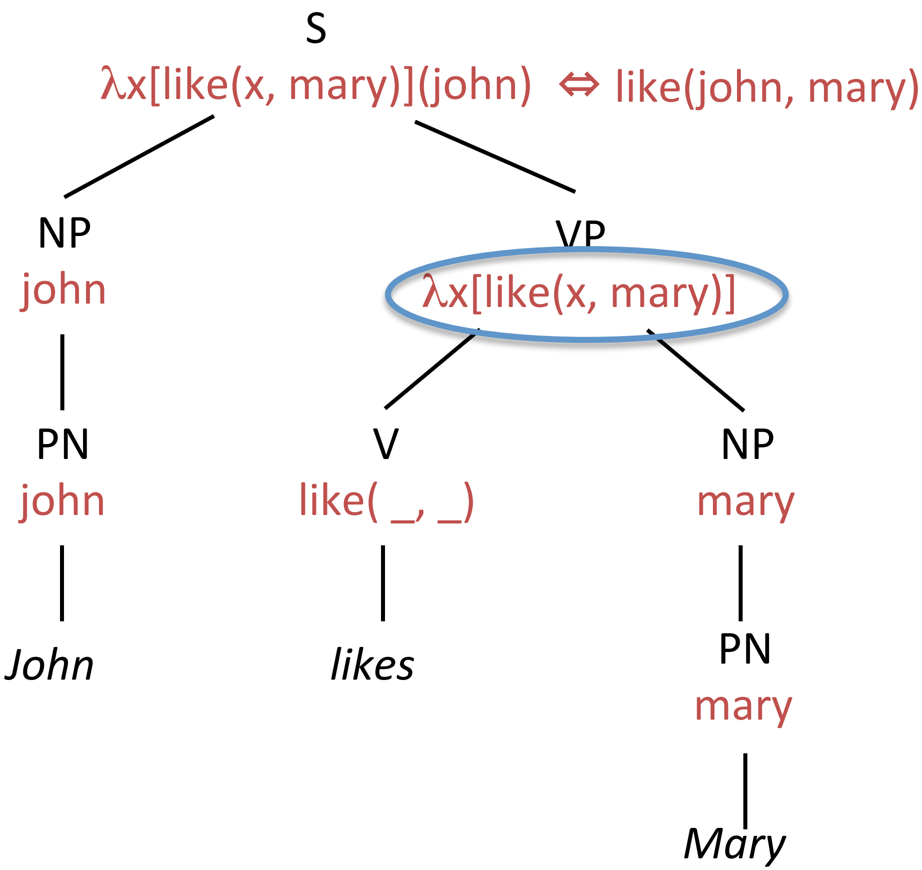
iff $V_M(\text{john}) \in \{a \in U_M | \langle a, V_M(\text{mary})\rangle \in V_M(\text{like})\}$

iff $\langle V_M(\text{john}) , V_M(\text{mary})\rangle \in V_M(\text{like})$

iff $[\![\text{like}(\text{john}, \text{mary})]\!]^{M,g} = 1$

Semantic Composition

- *John likes Mary* ⇒ like(john, mary)

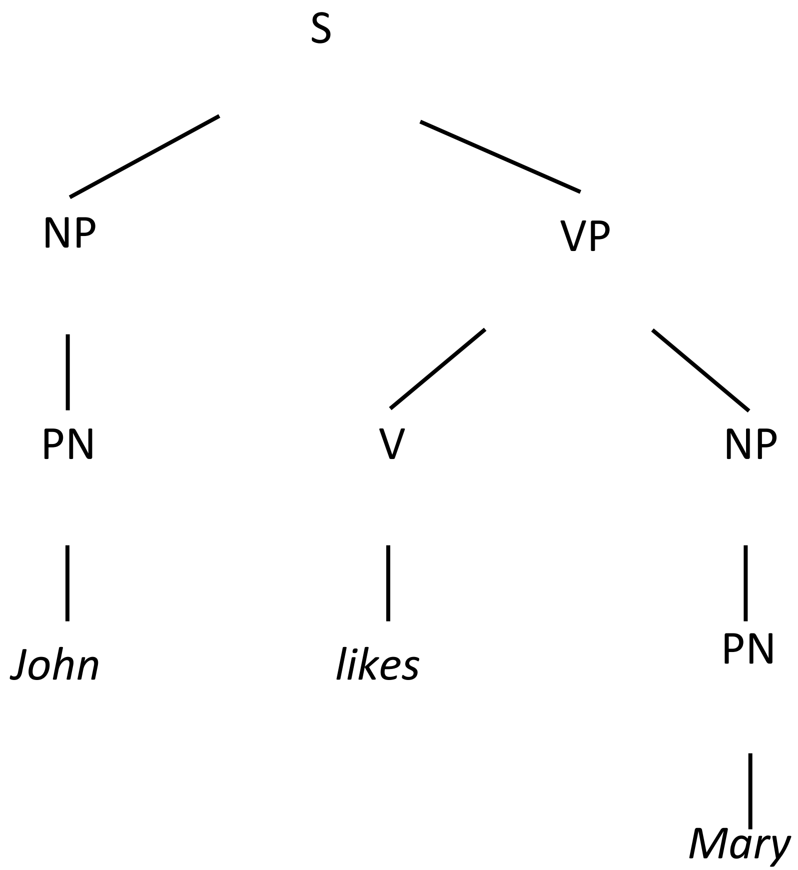


λ-Conversion

- λx[student(x)](john) and student(john) are equivalent, and so are λx[like(x, mary)](john) and like(john, mary).
- In general: λxA(b) ⇔ A[x/b] , where A[x/b] is the result of replacing all free occurrences of variable x in A with b. This equivalence holds independent of the choice of A and b.
- Thus, we can rewrite any application of a λ-expression λxA to an argument b by the result of substituting all free occurrences of the λ-variable x in A with b (without considering truth conditions).
- λxA(b) ⇒ A[x/b] as a rewrite rule is called the rule of λ-conversion or λ-reduction.

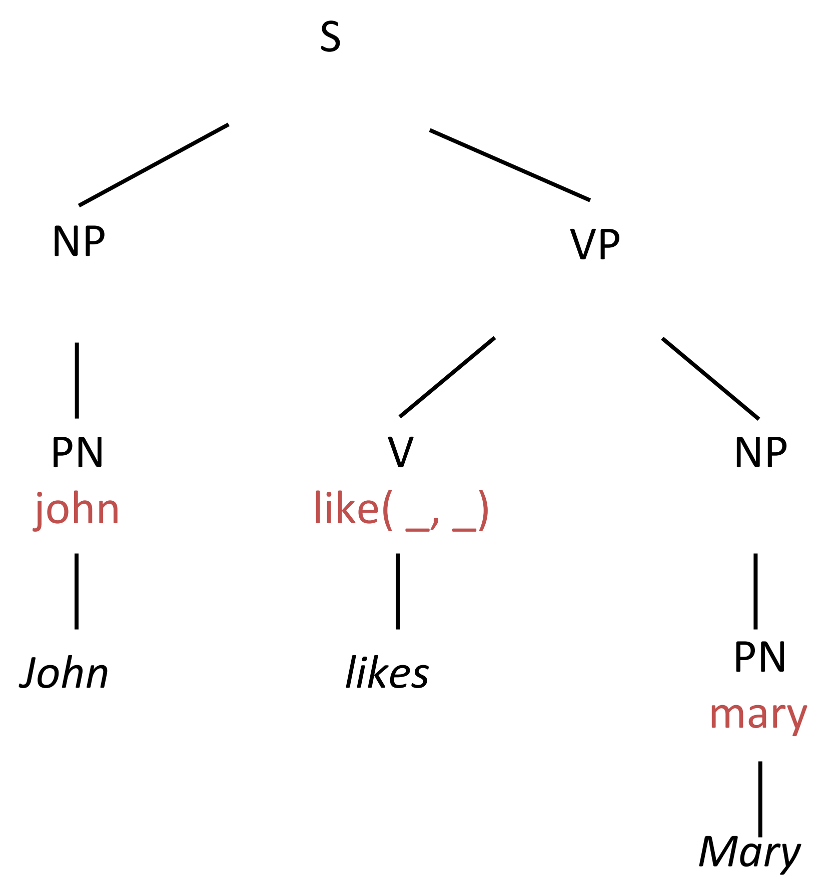
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



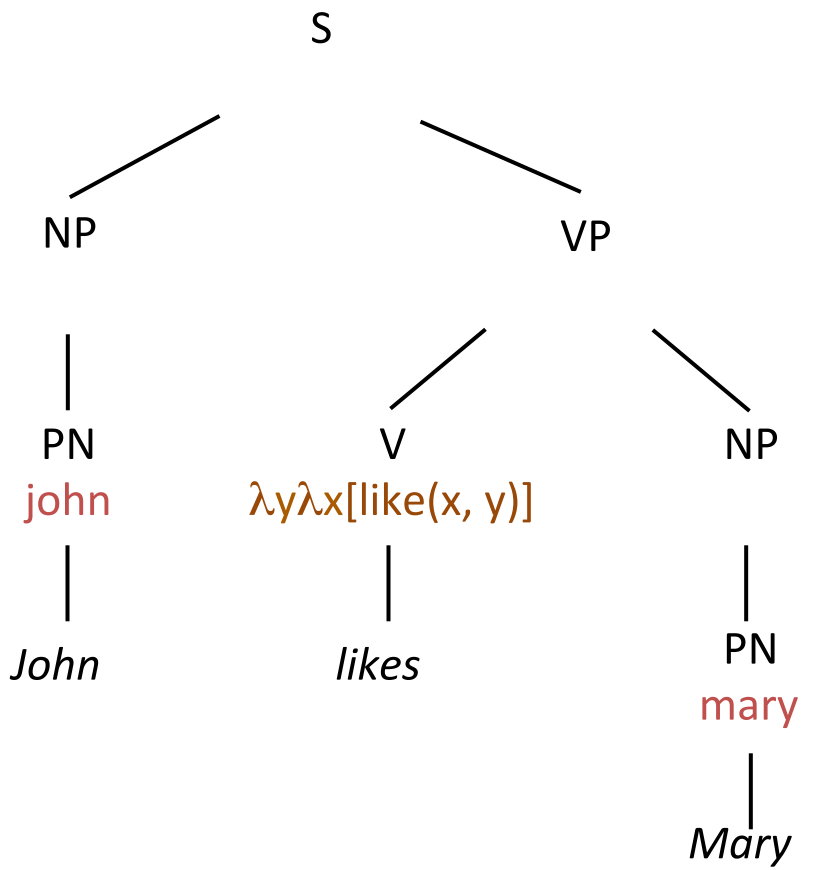
Semantic Composition

- *John likes Mary* ⇒ like(john, mary)



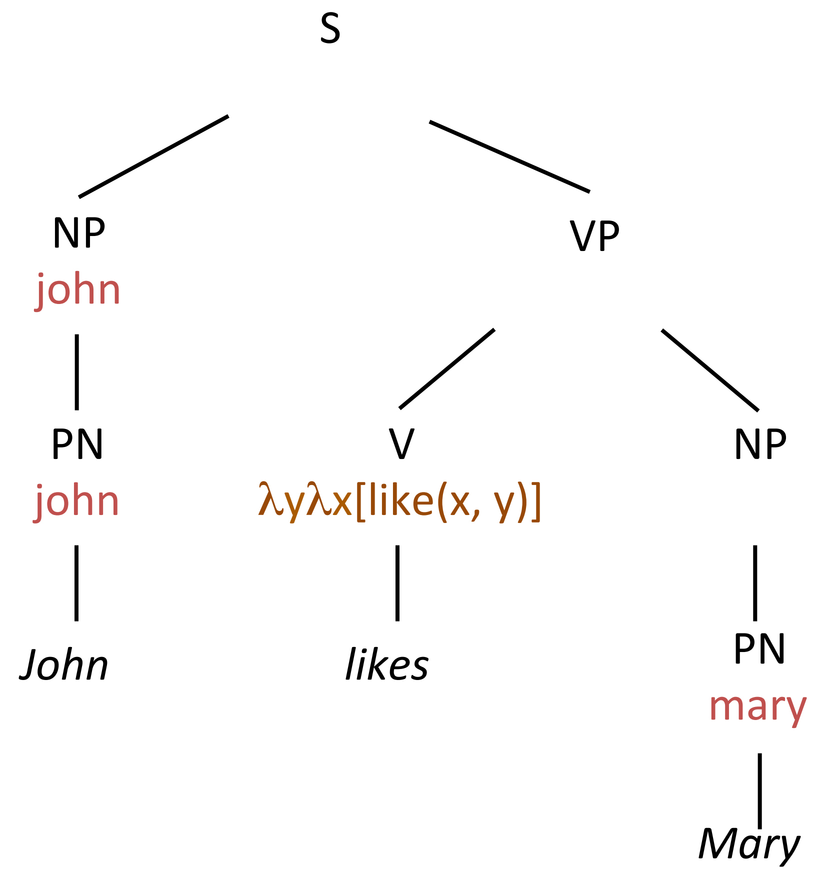
Semantic Composition: Lexical Information

- *John likes Mary* ⇒ like(john, mary)



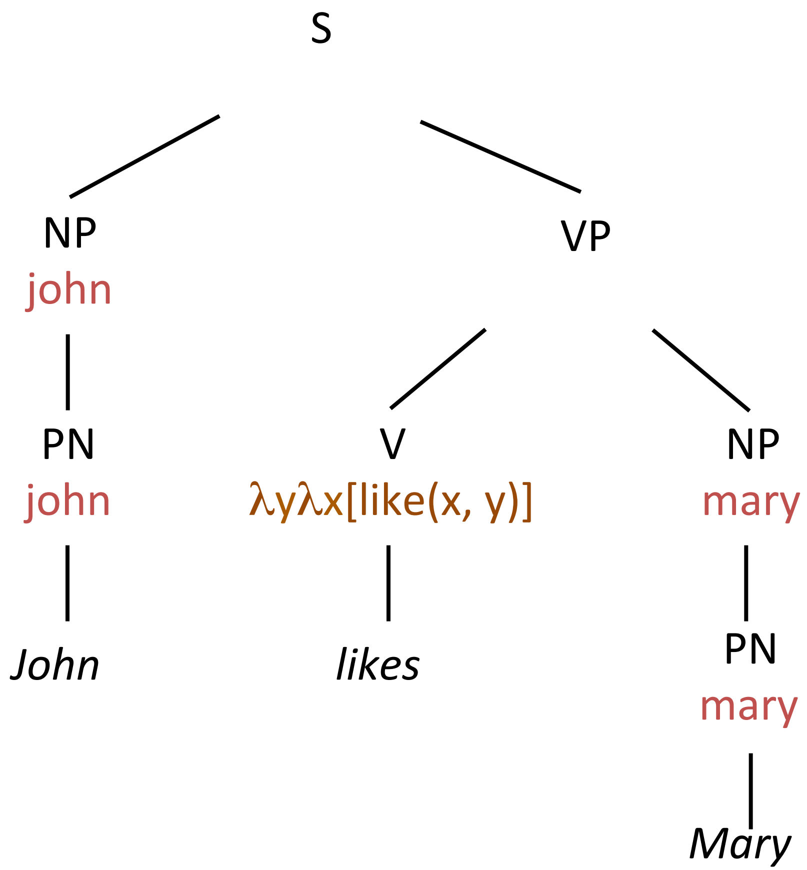
Semantic Composition: Projection

- *John likes Mary* ⇒ like(john, mary)



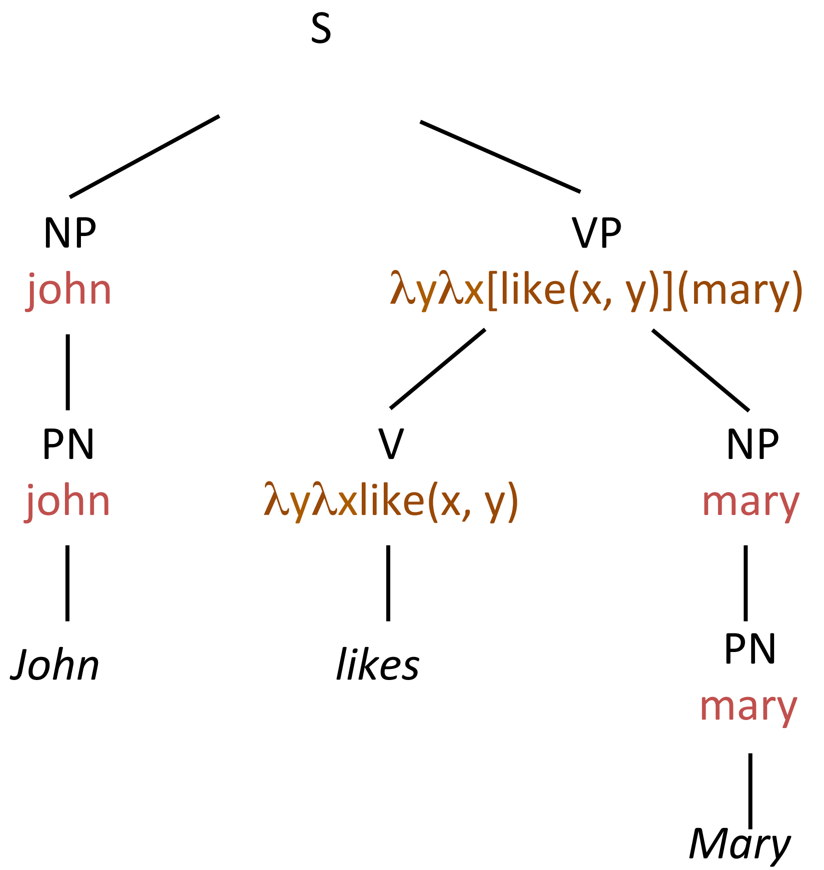
Semantic Composition: Projection

- *John likes Mary* ⇒ like(john, mary)



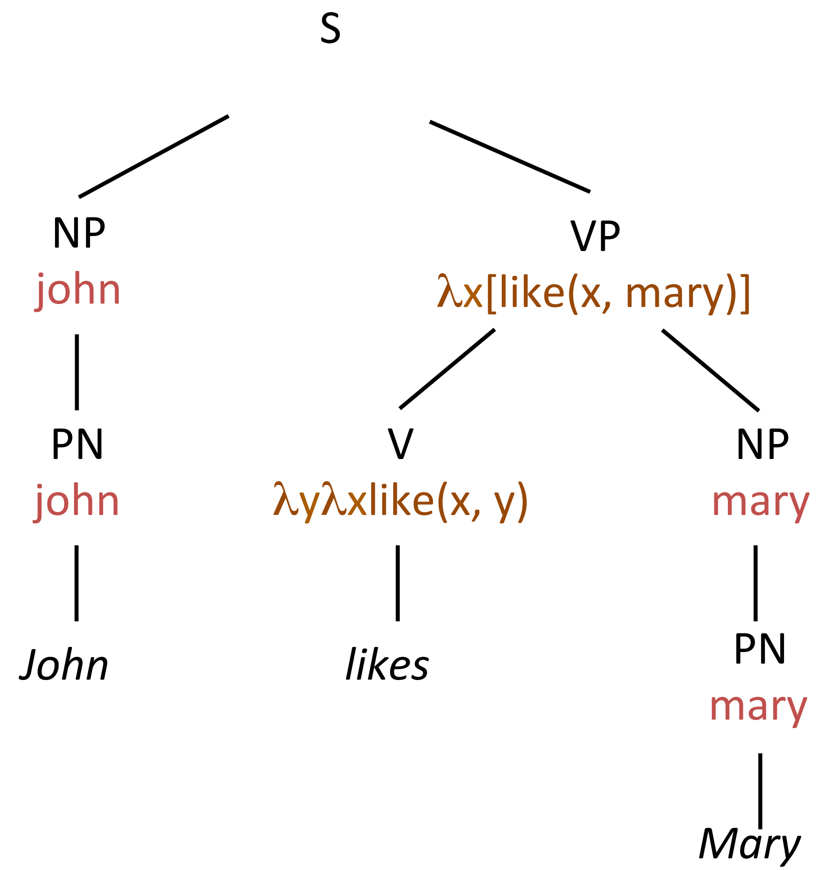
Semantic Composition: Application

- *John likes Mary* ⇒ like(john, mary)



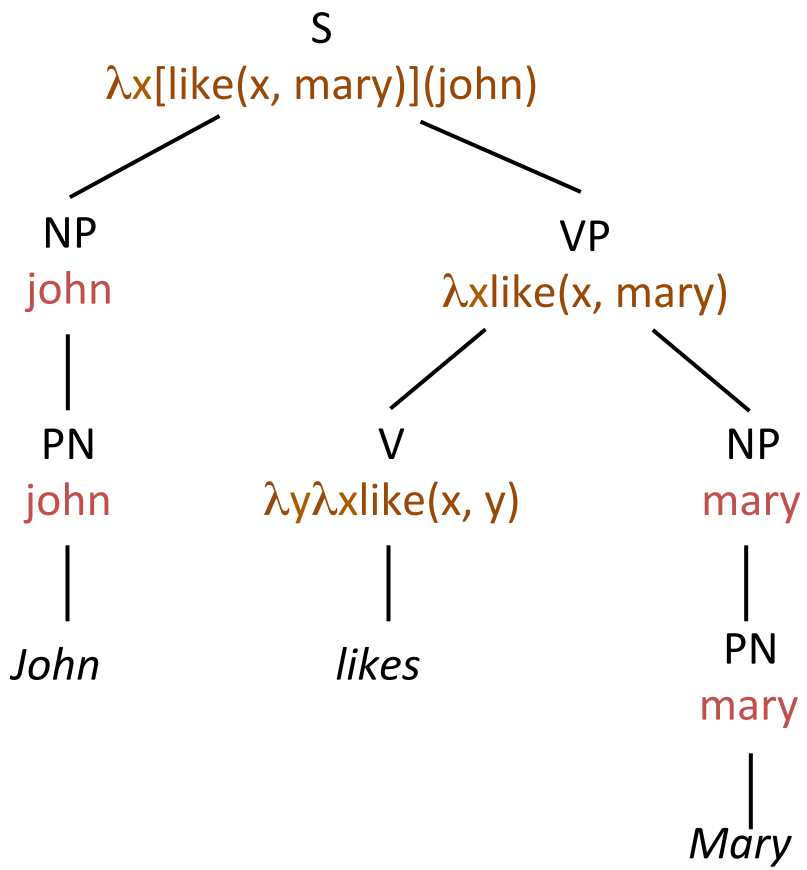
Semantic Composition: Reduction

- *John likes Mary* ⇒ like(john, mary)



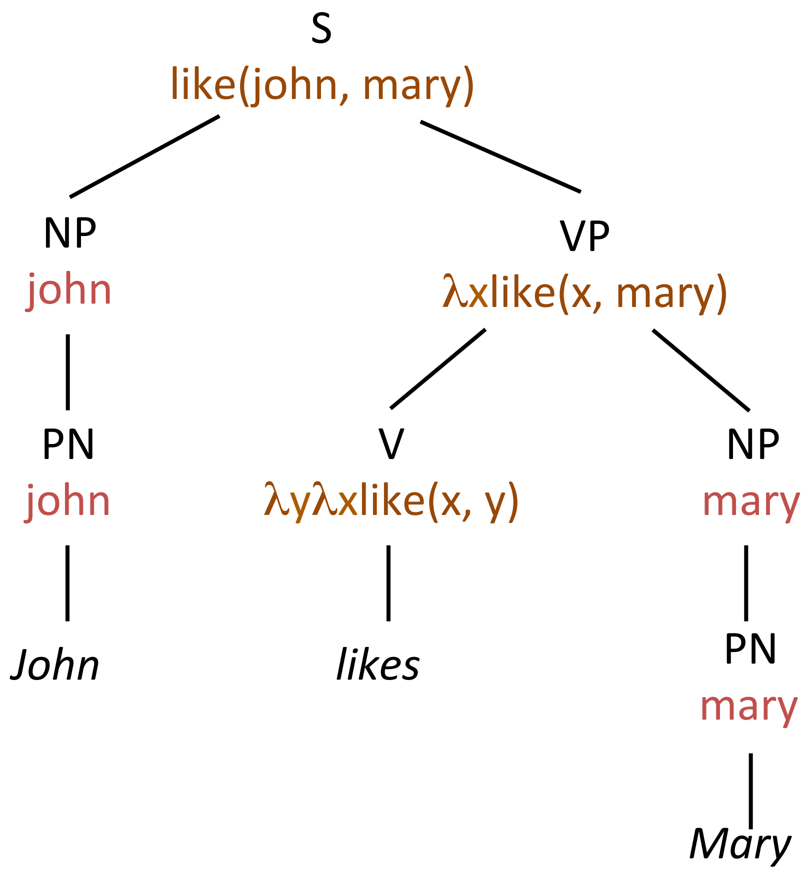
Semantic Composition: Application

- *John likes Mary* ⇒ like(john, mary)



Semantic Composition: Reduction

- *John likes Mary* ⇒ like(john, mary)



More λ-Expressions

"to like Mary"
$\lambda x[like(x, mary)]$

"to be liked by Mary"
$\lambda x[like(mary, x)]$

"to like oneself"
$\lambda x[like(x, x)]$

"to sing and dance"
$\lambda x[sing(x) \wedge dance(x)]$

"to be somebody, whom everyone likes"
$\lambda x[\forall y\ like(y, x)]$