Compositional Semantics

Berkeley NLP

Jacob Andreas

Problem 1

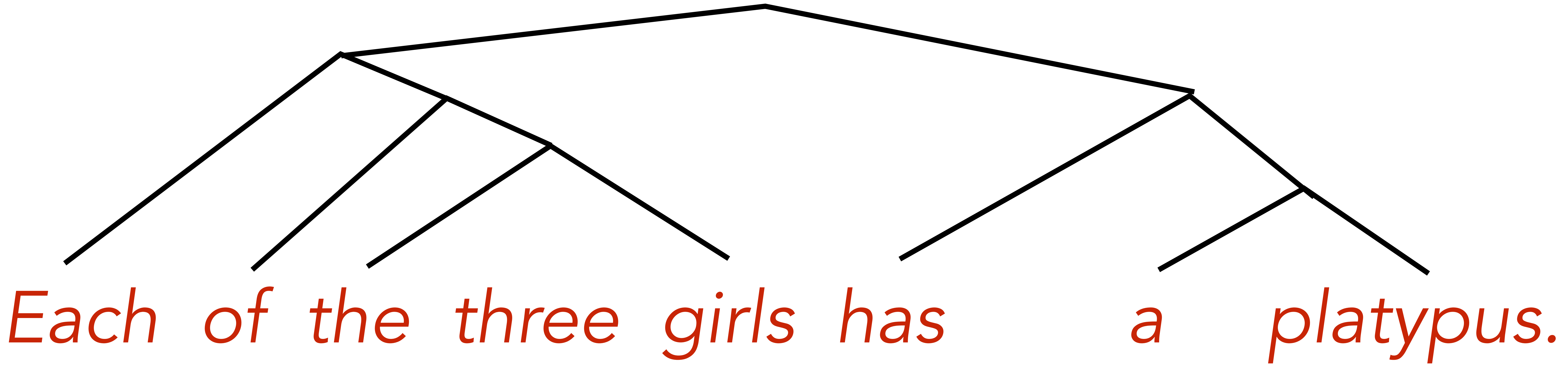
Each of the three girls has a platypus.

Each of the three girls climbed the mountain.

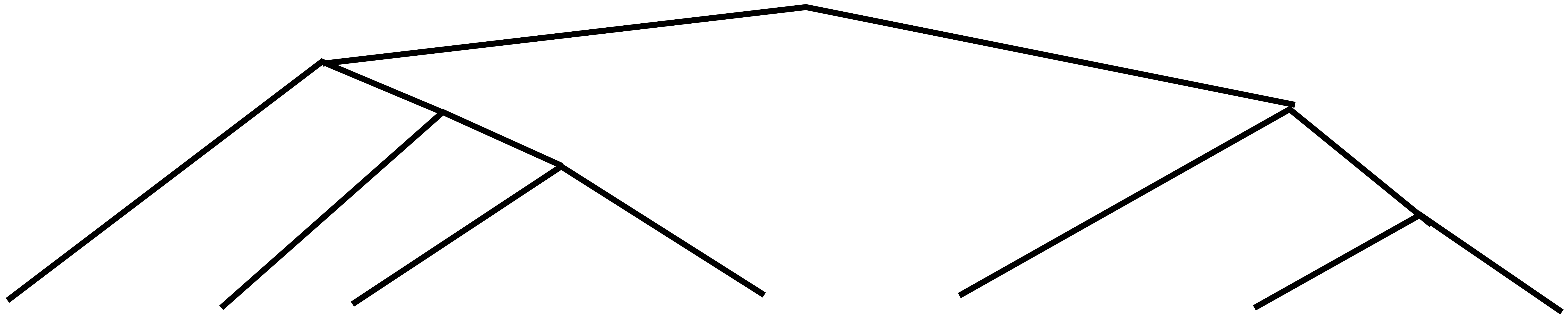
How many platypuses?

How many mountains?

Problem 1



Problem 1



Each of the three girls climbed the mountain.

Problem 2

There are 128 cities in South Carolina.

name	type	coastal
Columbia	city	no
Cooper	river	yes
Charleston	city	yes

Problem 3

Barack Obama was the 44th President of the United States. Obama was born on August 4 in Honolulu, Hawaii. In late August 1961, Obama's mother moved with him to the University of Washington in Seattle for a year…

Is Barack Obama from the United States?

Compositional semantics

It's not enough to have structured representations of syntax:
We also need structured representations of **meaning**.

Compositional semantics

It's not enough to have structured representations of syntax:
We also need structured representations of **meaning**.

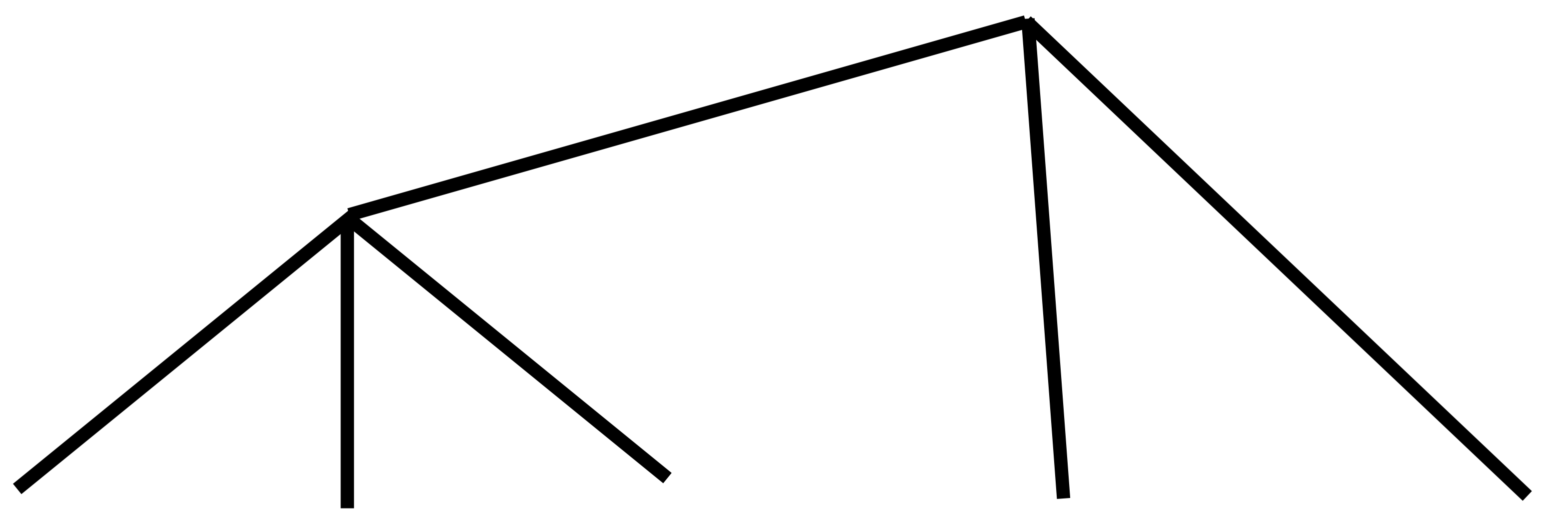
Today:
How do we get from **language** to **meaning?**

PART I
What is meaning?

Meaning in formal languages

$$a + b = 17$$

Meaning in formal languages



$$a + b = 17$$

Meaning in formal languages

$$a + b = 17$$

$$a = ?$$

$$b = ?$$

Meanings are sets of valid assignments

$$a + b = 17$$

{a=0, b=0}	{a=17, b=0}
{a=3, b=10}	{a=10, b=7}
{a=5, b=12}	{a=5, b=5}

Meanings are sets of valid assignments

$a + b = 17$

{a=0, b=0} ✘

{a=17, b=0} ✔

{a=3, b=10} ✘

{a=10, b=7} ✔

{a=5, b=12} ✔

{a=5, b=5} ✘

Meanings are sets of valid assignments

$$a + 3 = 20 - b$$

{a=0, b=0} ✘

{a=17, b=0} ✔

{a=3, b=10} ✘

{a=10, b=7} ✔

{a=5, b=12} ✔

{a=5, b=5} ✘

Meanings are *functions* that judge validity

$$[\![a + b = 17]\!]$$

$\{a=5, b=12\}$

✔

Meanings are *functions* that judge validity

$$[\![a + b = 17]\!]$$

$\{a=3,\ b=10\}$ → $[\![a + b = 17]\!]$ → ✗

Lessons from math

$$[\![a + b = 17]\!]$$

The meaning of a statement is the **set** of possible worlds consistent with that statement.

Here, a “possible world” is an assignment of values to variables.

$$\{a=3,\ b=10\}$$

Meaning in natural languages

Pat likes Sal.

Representing possible worlds

Individuals	Pat	Sal
Properties	whale •—	sad •—
Relations	—loves→	—contains→

Example world

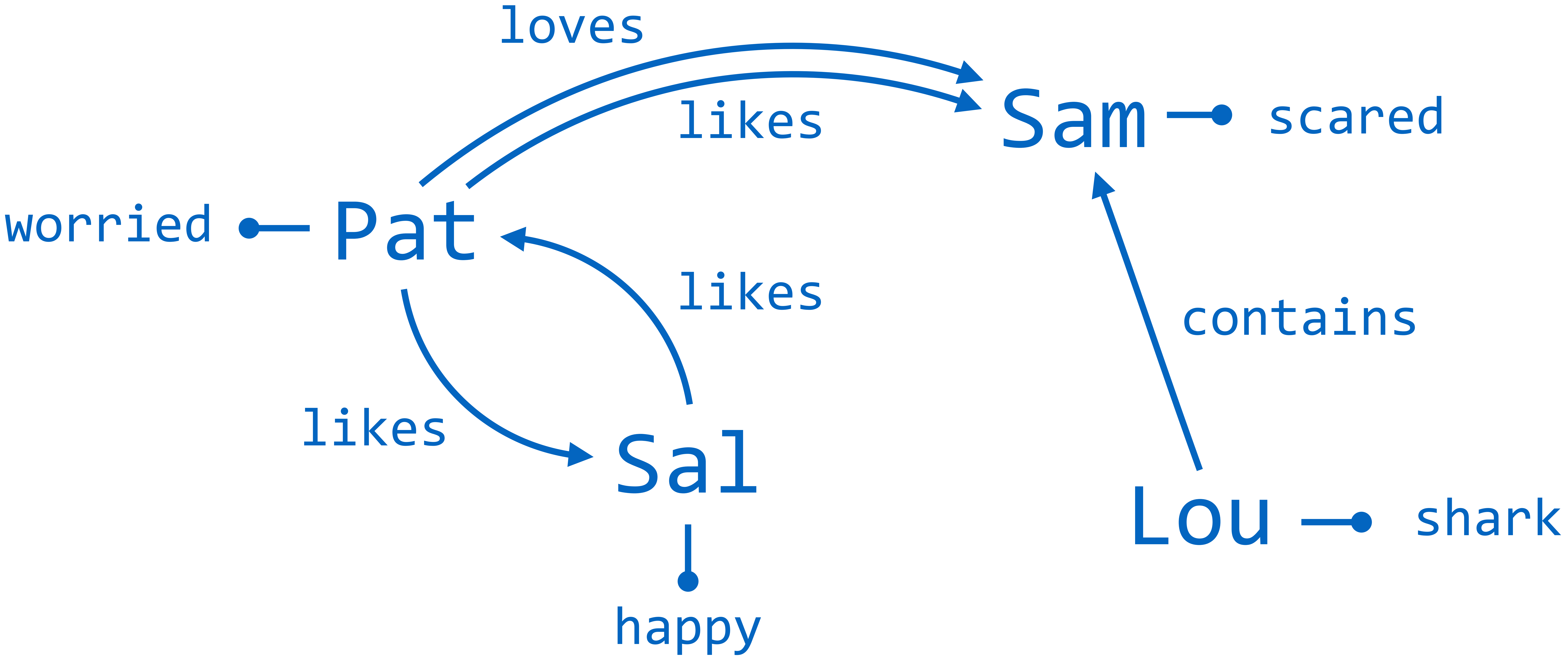
Sam

Pat

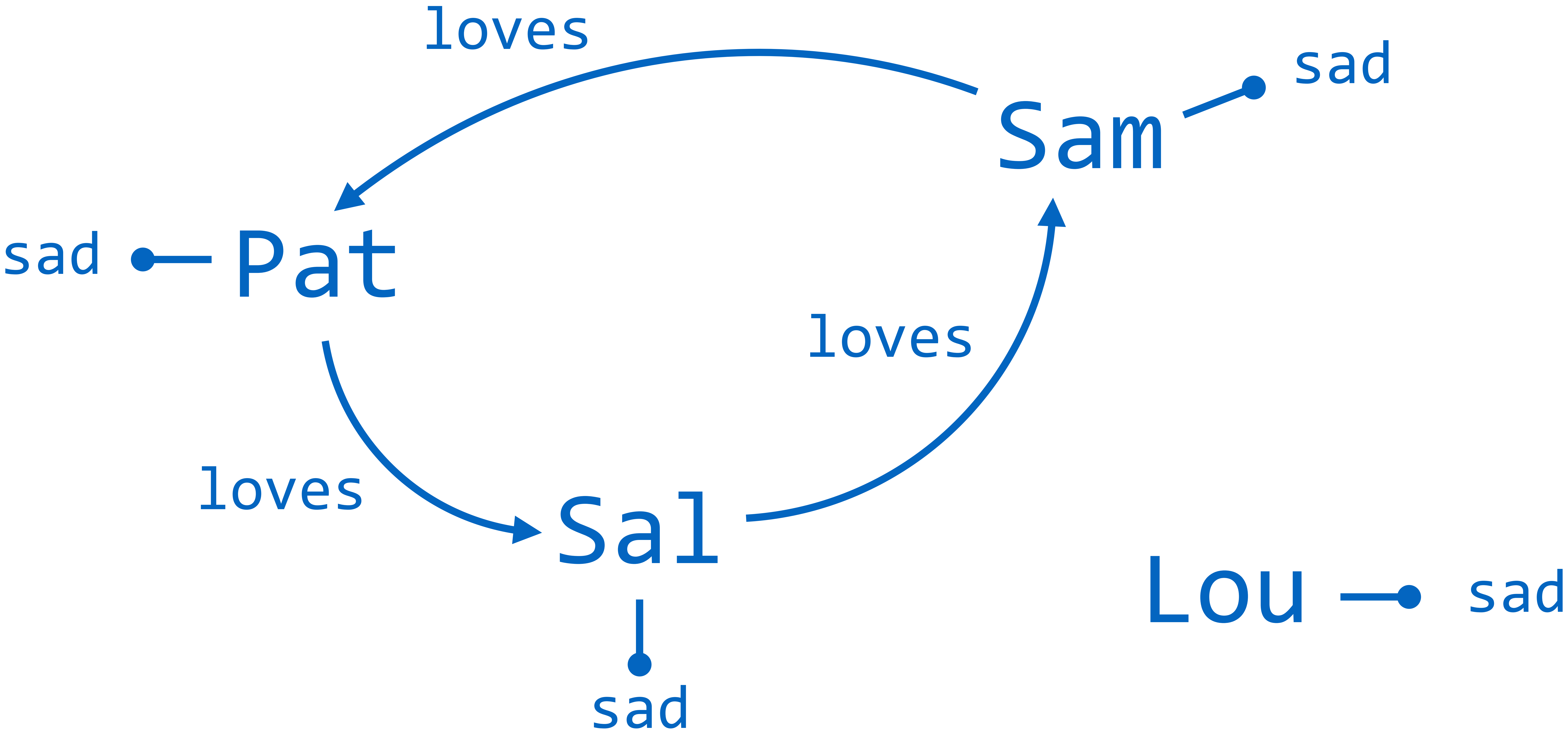
Sal

Lou

Example world



Different example world

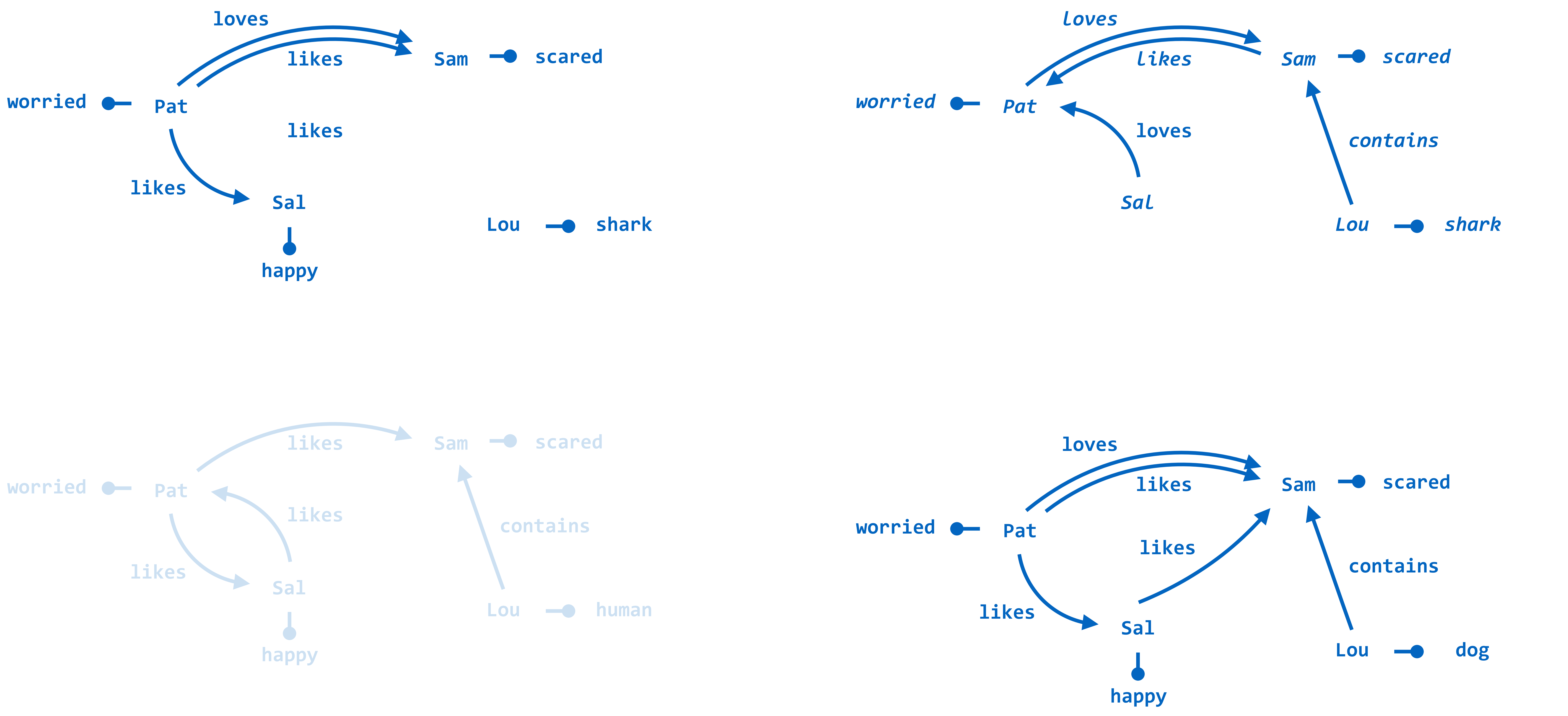


Representing possible worlds

Individuals	`Pat`	`Sal`
Properties	`whale={Lou}, sad={Pat,Sal}`	
Relations	`likes={(Pat,Sal),(Sal,Sam)}`	

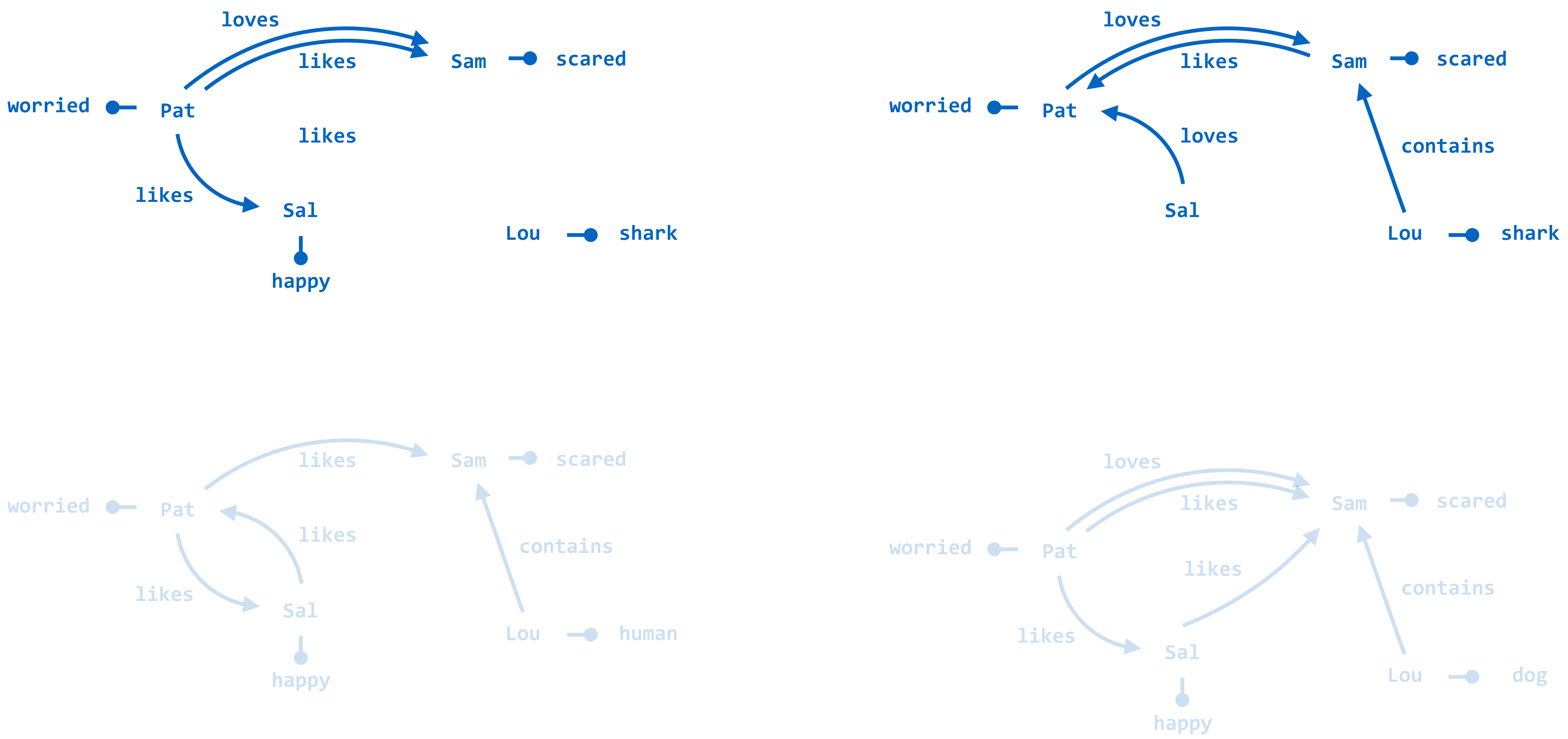
Interpretations of sentences

Pat likes Sal.



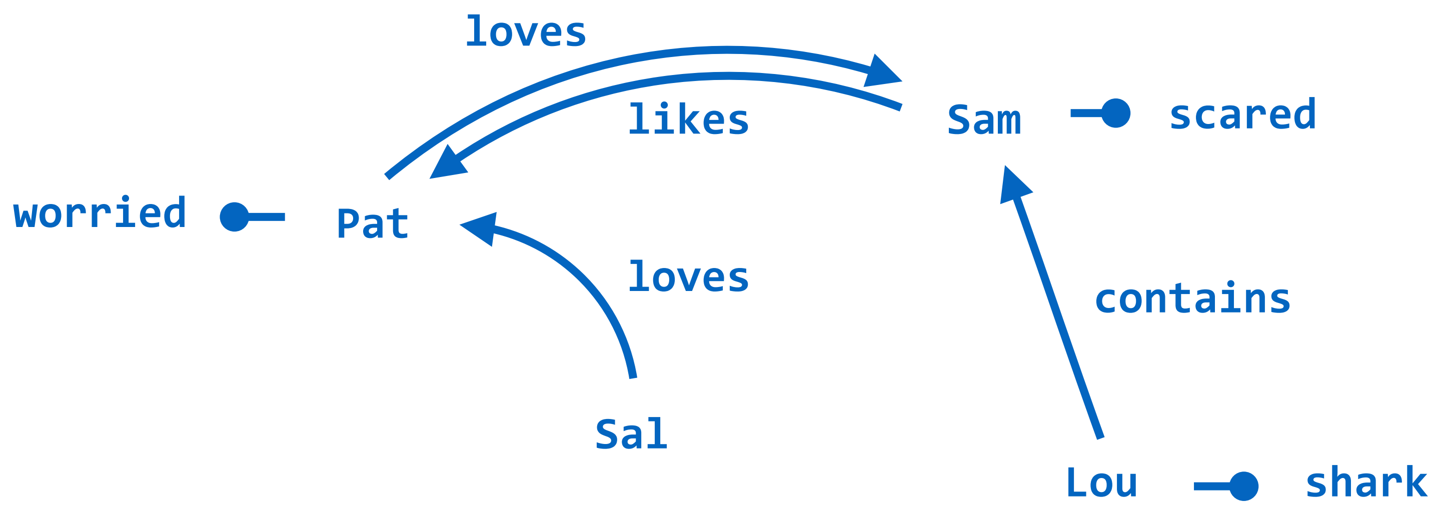
Interpretations of sentences

Lou is a shark.



Interpretations of sentences

Sam is inside Lou, a shark.



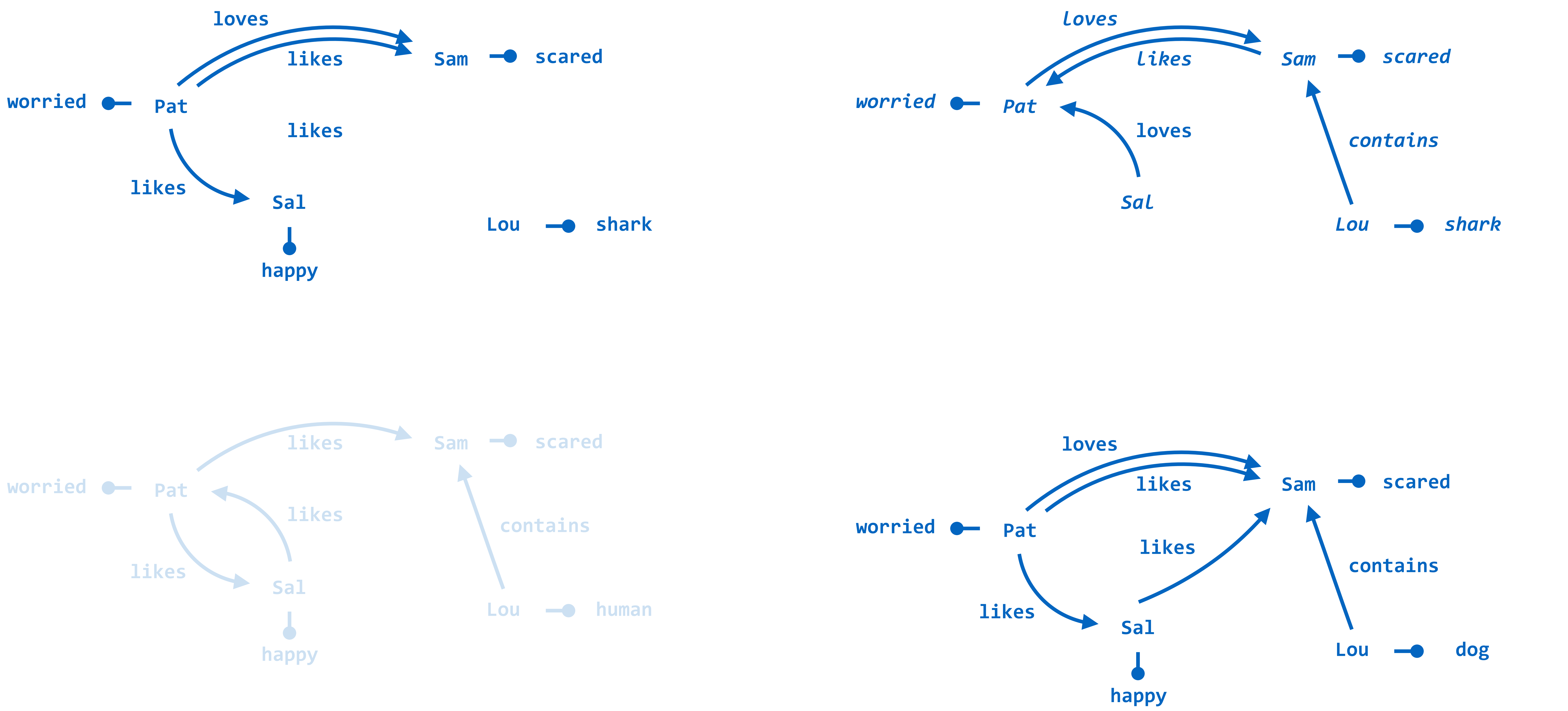
Key idea

The meaning of a sentence is the set of possible worlds it picks out.

Part II
How is meaning constructed?

Explicit representation is too hard

Pat likes Sal.



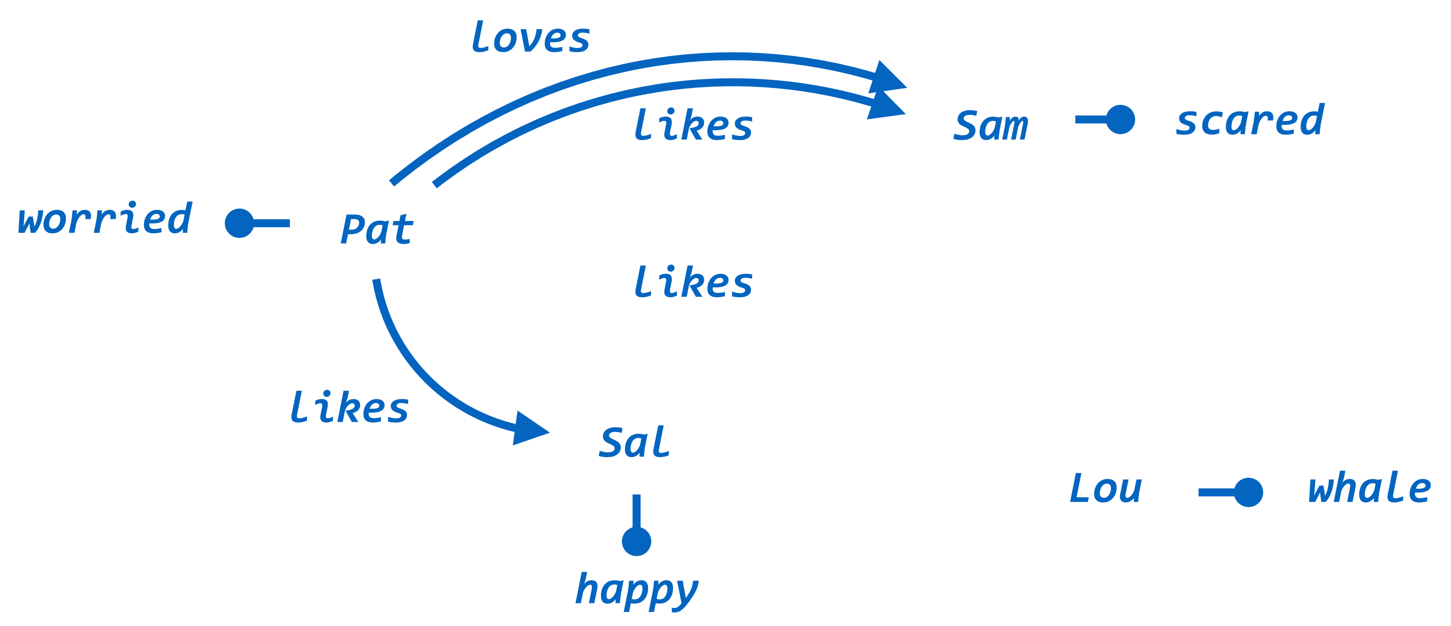
Meanings as functions

⟦*Pat likes Sal*⟧

Meanings as logical statements

⟦*Pat likes Sal*⟧

`likes(Pat, Sal)`



✔

Expressing functions with logic

Pat likes Sal

```
likes(Pat, Sal)
```

Meanings as logical statements

Lou is a shark

`shark(Lou)`

Meanings as logical statements

Sam is inside Lou, a shark

Meanings as logical statements

Sam is inside Lou, a shark

`shark(Lou) ∧ contains(Lou, Sam)`

Meanings as logical statements

Nobody likes Lou

Meanings as logical statements

Nobody likes Lou

$\forall x.\ \neg \text{likes}(x, \text{Lou})$

Meanings as logical statements

Everyone who knows Sal is happy

Meanings as logical statements

Everyone who knows Sal is happy

$\forall x.\ \mathrm{knows}(x, \mathrm{Sal}) \rightarrow \mathrm{happy}(x)$

KEY IDEA

Collections of possible worlds can be compactly represented with logical forms.

Compositionality of meaning

Pat likes Sal	`likes(Pat, Sal)`
Lou is a shark	`shark(Lou)`
Sam is inside Lou, a shark	`shark(Lou) ∧ contains(Lou, Sam)`
Nobody likes Lou	`∀x.¬likes(x, Lou)`

Compositionality of meaning

Pat likes Sal	`likes(Pat, Sal)`
Lou is a shark	`shark(Lou)`
Sam is inside Lou, a shark	`shark(Lou) ∧ contains(Lou, Sam)`
Nobody likes Lou	`∀x.¬likes(x, Lou)`

Compositionality of meaning

Pat likes Sal	`likes(Pat, Sal)`
Lou is a shark	`shark(Lou)`
Sam is inside Lou, a shark	`shark(Lou) ∧ contains(Lou, Sam)`
Nobody likes Lou	`∀x.¬likes(x, Lou)`

Compositionality of meaning

A Sal le gusta Pat — `likes(Pat, Sal)`

Lou es un tiburón — `shark(Lou)`

Sam está dentro de Lou, un tiburón — `shark(Lou) ∧ contains(Lou, Sam)`

A nadie le gusta Lou — `∀x.¬likes(x, Lou)`

Compositionality of meaning

a12 b5 c67 a8	`likes(Pat, Sal)`
a12 b5 c0 a0	`shark(Lou)`
a12 b16 c12 c12	`shark(Lou) ∧ contains(Lou, Sam)`
a53	`∀x.¬likes(x, Lou)`

Key idea

Pieces of logical forms correspond to pieces of language

Building a lexicon

Sam is inside Lou, a shark `shark(Lou) ∧ contains(Lou, Sam)`

Pat: `Pat`

Sal: `Sal`

Sam: `Sam`

Lou: `Lou`

Building a lexicon

Sam is inside Lou, a shark `shark(Lou) ∧ contains(Lou, Sam)`

Pat: `Pat`

Sal: `Sal`

Sam: `Sam`

Lou: `Lou`

shark:

Building a lexicon

Sam is inside Lou, a shark `shark(Lou) ∧ contains(Lou, Sam)`

Pat: `Pat`

Sal: `Sal`

Sam: `Sam`

Lou: `Lou`

shark: `λx.shark(x)`

Building a lexicon

Sam is inside Lou, a shark `shark(Lou) ∧ contains(Lou, Sam)`

Pat: `Pat`

Sal: `Sal`

Sam: `Sam`

Lou: `Lou`

shark: `λx.shark(x)`

likes: `λyx.likes(x, y)`

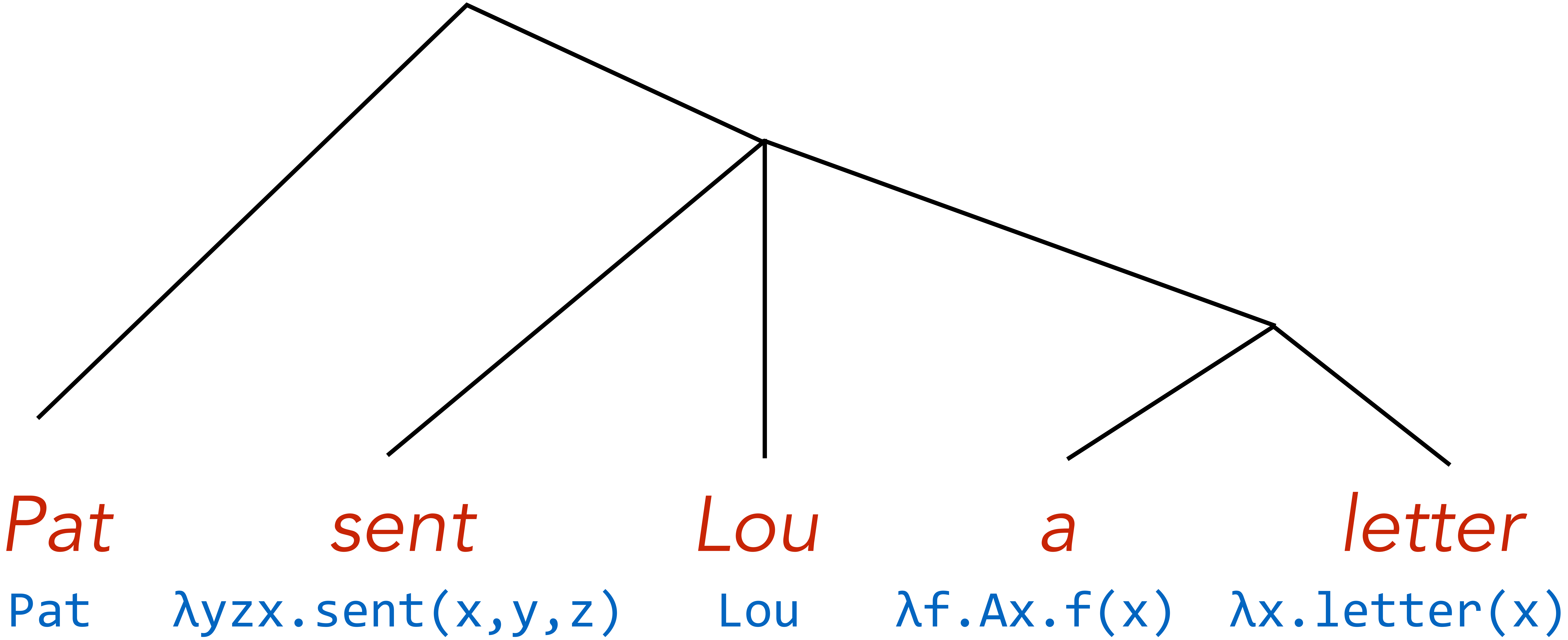
nobody: `λf.∀x.¬f(x)`

...

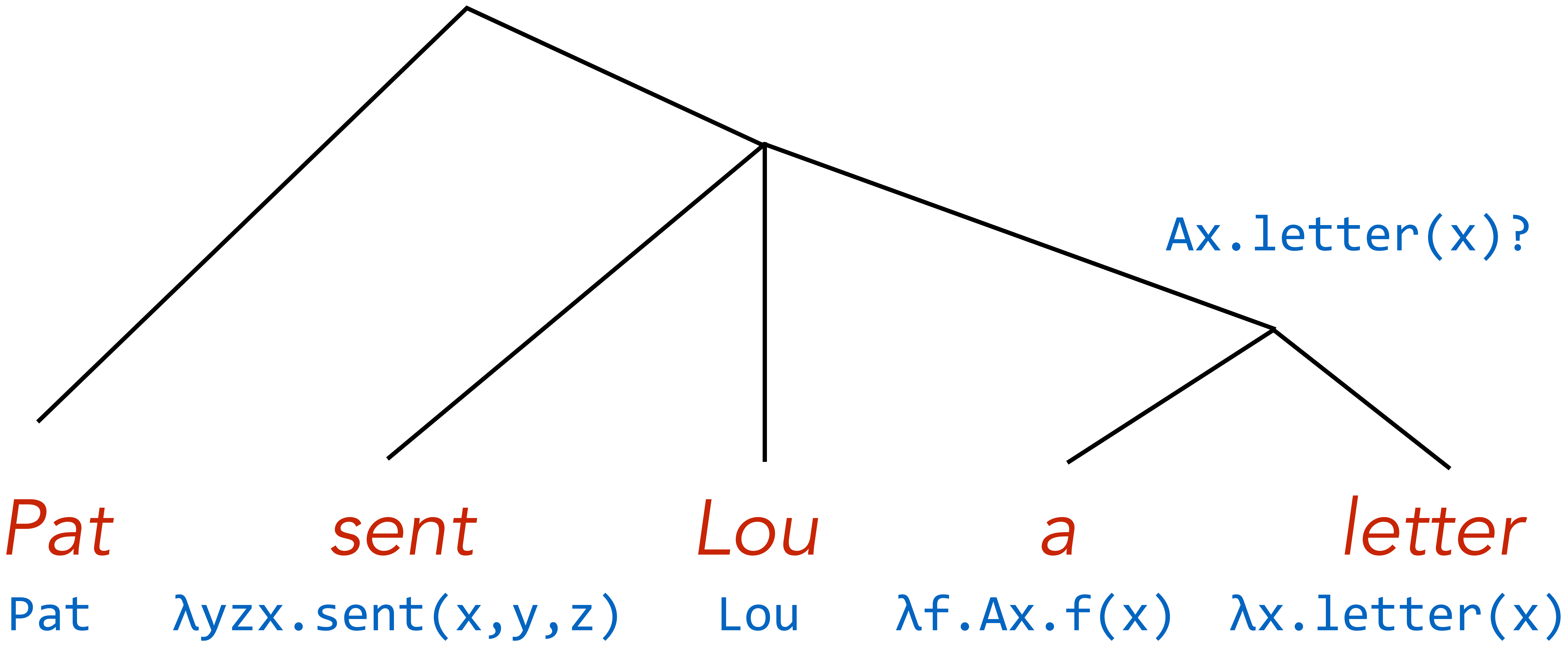
What do we do now?

Pat	*sent*	*Lou*	*a*	*letter*
`Pat`	`λyzx.sent(x,y,z)`	`Lou`	`λf.Ax.f(x)`	`λx.letter(x)`

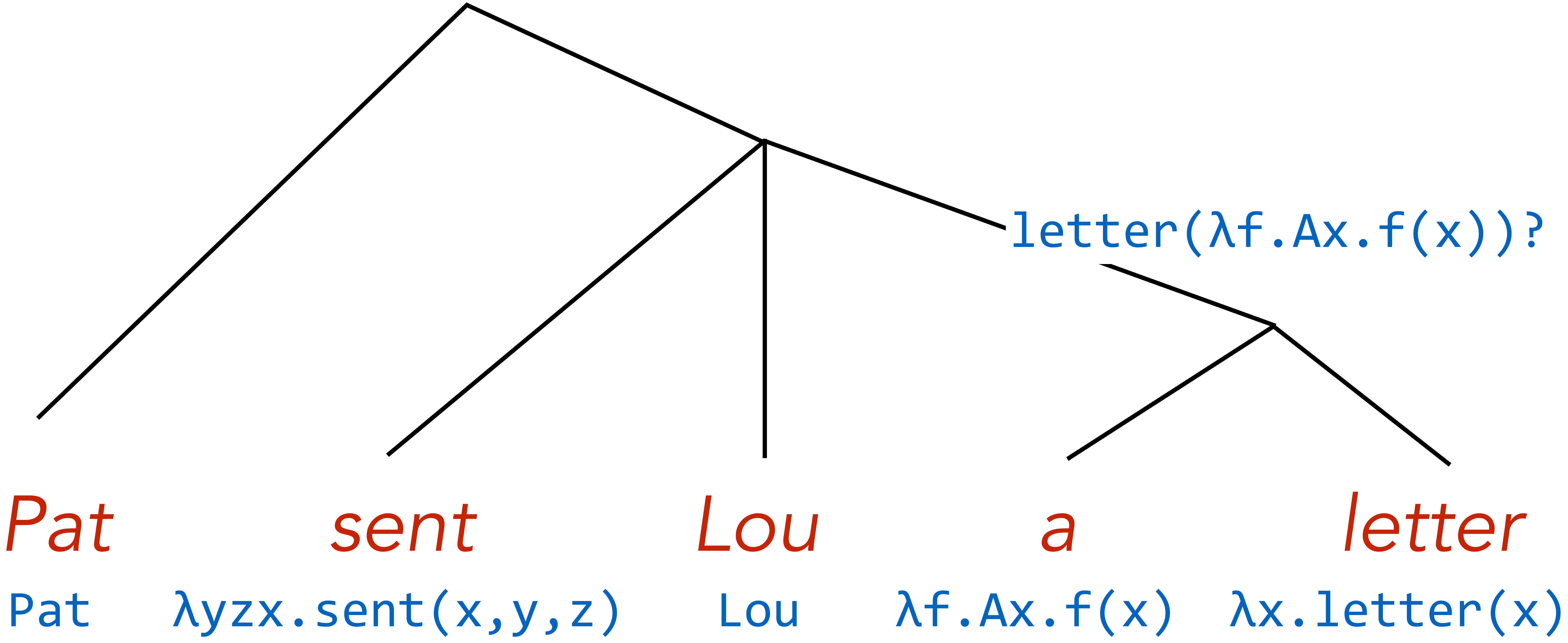
What do we do now?



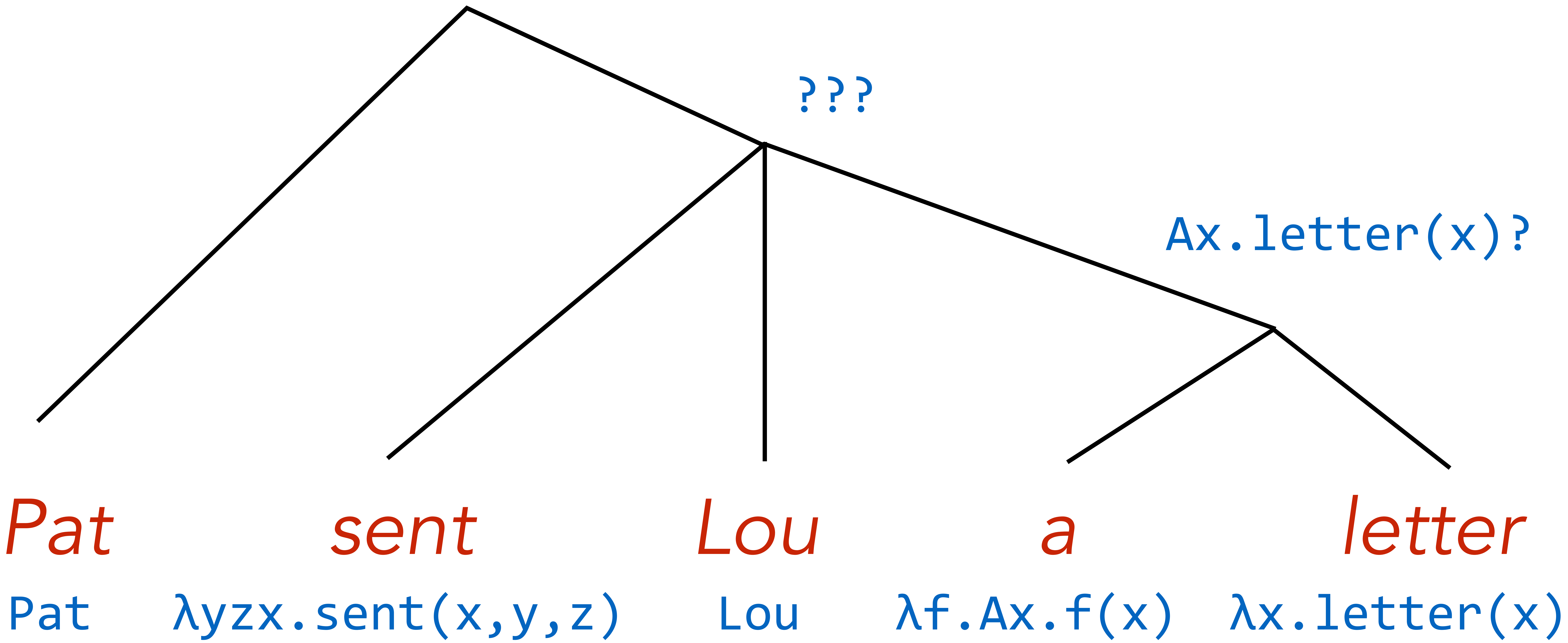
What do we do now?



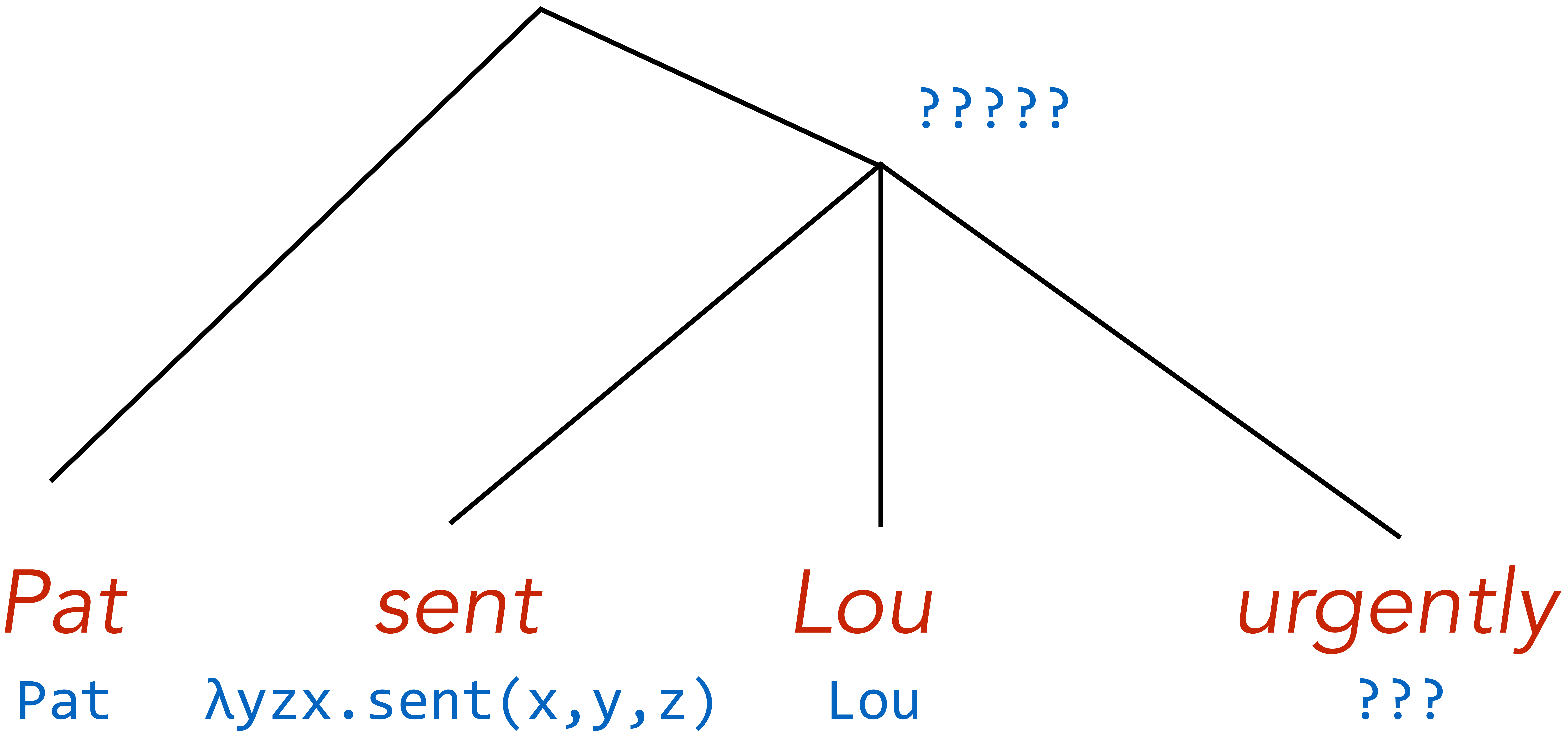
What do we do now?



What do we do now?



What do we do now?



Semantic types

Pat	*sent*	*Lou*	*a*	*letter*
Pat	λyzx.sent(x,y,z)	Lou	λf.Ax.f(x)	λx.letter(x)
Object	{Obj,Obj,Obj} → Bool	Object	(Object → Bool) → Bool	Object → Bool

Semantic types & syntax

Pat	*sent*	*Lou*	*a*	*letter*
`Pat`	`λyzx.sent(x,y,z)`	`Lou`	`λf.Ax.f(x)`	`λx.letter(x)`
NP	NP,NP,NP ↓ S	NP	[NP ↓ S] ↓ S	NP ↓ S

Semantic types & syntax

Pat	*sent*	*Lou*	*a*	*letter*
Pat	λyzx.sent(x,y,z)	Lou	λf.Ax.f(x)	λx.letter(x)
NP	((S|NP)|NP)|NP	NP	S|(S|NP)	S|NP

Categorial grammar

Pat	*sent*	*Lou*	*a*	*letter*
`Pat`	`λyzx.sent(x,y,z)`	`Lou`	`λf.Ax.f(x)`	`λx.letter(x)`
NP	((S\NP)/NP)/NP	NP	NP/(S/NP)	S/NP

Parsing with a categorial grammar

Pat	*sent*	*Lou*	*a*	*letter*
`Pat`	`λyzx.sent(x,y,z)`	`Lou`	`λf.Ax.f(x)`	`λx.letter(x)`
`NP`	`((S\NP)/NP)/NP`	`NP`	`NP/(S/NP)`	`S/NP`

―――――――――――――――――――――

`Ax.letter(x)`

`NP`

Parsing with a categorial grammar

Pat	*sent*	*Lou*	*a*	*letter*
Pat	λyzx.sent(x,y,z)	Lou	λf.Ax.f(x)	λx.letter(x)
NP	((S\NP)/NP)/NP	NP	NP/(S/NP)	S/NP

sent + Lou:

λzx.sent(x,Lou,z)

(S\NP)/NP

a + letter:

Ax.letter(x)

NP

Parsing with a categorial grammar

Pat	*sent*	*Lou*	*a*	*letter*
Pat	λyzx.sent(x,y,z)	Lou	λf.Ax.f(x)	λx.letter(x)
NP	((S\NP)/NP)/NP	NP	NP/(S/NP)	S/NP

λzx.sent(x,Lou,z)
(S\NP)/NP

Ax.letter(x)
NP

λx.sent(x,Lou,Ax.letter(x)) S\NP

Parsing with a categorial grammar

Pat	*sent*	*Lou*	*a*	*letter*
Pat	λyzx.sent(x,y,z)	Lou	λf.Ax.f(x)	λx.letter(x)
NP	((S\NP)/NP)/NP	NP	NP/(S/NP)	S/NP

λzx.sent(x,Lou,z)
(S\NP)/NP

Ax.letter(x)
NP

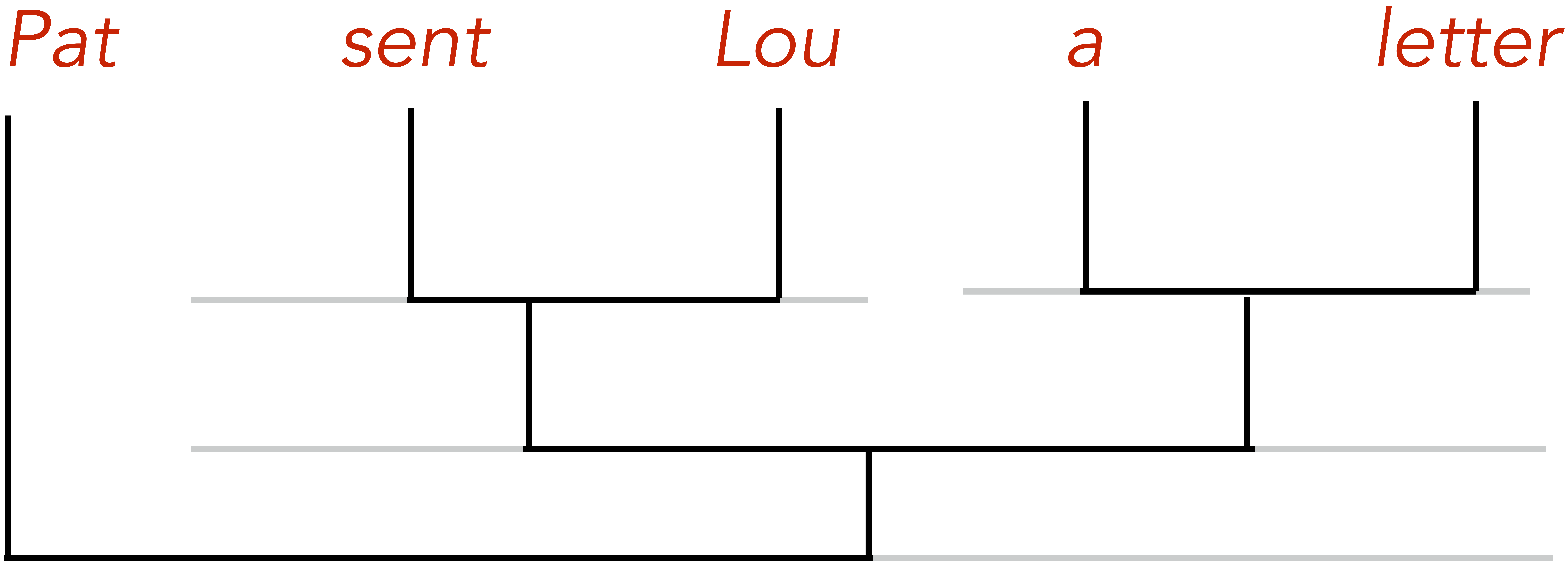
λx.sent(x,Lou,Ax.letter(x)) S\NP

sent(Pat,Lou,Ax.letter(x)) S

Semantics → Synax!

Pat *sent* *Lou* *a* *letter*

Semantics → Synax!



Key idea

Types in logic correspond to grammatical categories in language

Problem 1

Each of the three girls has a platypus.

Each of the three girls climbed the mountain.

$$\forall x.\mathrm{girl}(x) \rightarrow \exists y.\mathrm{platypus}(y) \wedge \mathrm{has}(x, y)$$

$$\exists y.\mathrm{mountain}(y) \wedge \forall x.\mathrm{girl}(x) \rightarrow \mathrm{climbed}(x, y)$$

Problem 2

There are 128 cities in South Carolina

name	type	coastal
Columbia	city	no
Cooper	river	yes
Charleston	city	yes

Problem 2

There are 128 cities in South Carolina

```
same(128,
  count x. city(x) ∧
    in(x, SouthCarolina)
```

name	type	coastal
Columbia	city	no
Cooper	river	yes
Charleston	city	yes

Problem 3

Barack Obama was the 44th President of the United States. Obama was born on August 4 in Honolulu, Hawaii. In late August 1961, Obama's mother moved with him to the University of Washington in Seattle for a year…

Is Barack Obama from the United States?

Problem 3

Barack *United*
States. *onolulu,*
Hawaii. *ed with*
him to *year…*

`born(Obama, Hawaii, August 4)`

Is l *es?*

`from(Obama, United States)`

Problem 3

Barack … United

States … onolulu,

Ha … ith

hir … r…

born(Obama, Hawaii, August 4)

born(x, y, z) → from(x, y)

from(x, y) ∧ in(y, z) → from(x, z)

in(Hawaii, United States)

Is l … es?

from(Obama, United States)

Key idea

The meaning of a sentence is the set of possible worlds it picks out.

KEY IDEA

Collections of possible worlds can be compactly represented with logical forms.

Key idea

Pieces of logical forms correspond to pieces of language

Key idea

Types in logic correspond to grammatical categories in language

Bonus round
What's missing?

Saying what we mean

Q: *How do you like my cooking?*

Saying what we mean

Q: *How do you like my cooking?*

A: *It's extremely interesting.*

Saying what we mean

Q: *How do you like my cooking?*

A: *It's extremely interesting.*

Q: *Do you know what time it is?*

Saying what we mean

Q: *How do you like my cooking?*

A: *It's extremely interesting.*

Q: *Do you know what time it is?*

A: *Yes, I do.*

Belief & possibility

Sal might have seen a unicorn.

Pat thinks Sal saw a unicorn.

Pat wants to find a unicorn.

Key idea

Not all meaning is literal!

Bonus round
Historical Notes

Alfred Tarski

Richard Montague

Learn more

ling121: "Logical Semantics"

Ted Briscoe's lecture notes:
https://www.cl.cam.ac.uk/teaching/1011/L107/semantics.pdf

Mark Steedman, "The Syntactic Process"