L98: Introduction to Computational Semantics

Lecture 14: Scope

Simone Teufel and Weiwei Sun

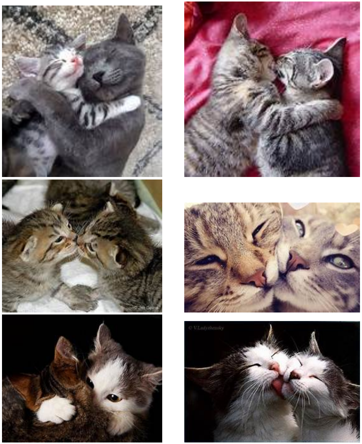
Natural Language and Information Processing Research Group
Department of Computer Science and Technology
University of Cambridge

Lent 2021/22

(1) Every cat loves a cat.

Lecture 14: Scope

1. What is scope?
2. Quantifier scope
3. Presuppositions of quantifiers
3. Negative scope
4. Other types of scope
5. Representing scope



What Is Scope?

Scope

Scope is an effect in syntax and semantics

- where a scopal lexical item casts its semantic effect over a particular part of the clause or phrase
- the entire part of the clause is then said to be in the scope of the scopal element
- e.g negative scope:

(2) a. He didn't see the cow

b. He saw no cow

c. He didn't only see the cow, but also the bull

Universal and Existential Scope

Reminder from lecture 8

(3) a. No student smokes
$\nexists x(\text{student'}(x) \wedge \text{smoke'}(x))$

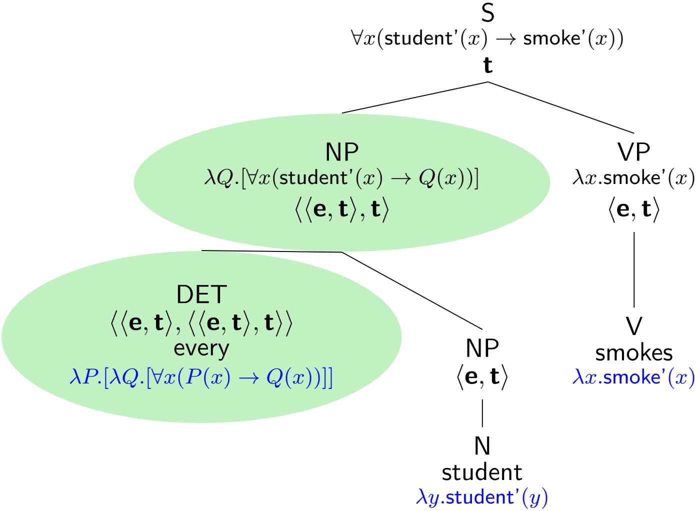
b. All/every student(s) smoke(s)
$\forall x(\text{student'}(x) \rightarrow \text{smoke'}(x))$

Lexical entries for the quantifiers:

$[\![\textit{no}]\!] = \lambda P.[\lambda Q.[\nexists x(P(x) \wedge Q(x))]]$
$[\![\textit{every}]\!] = \lambda P.[\lambda Q.[\forall x(P(x) \rightarrow Q(x))]]$

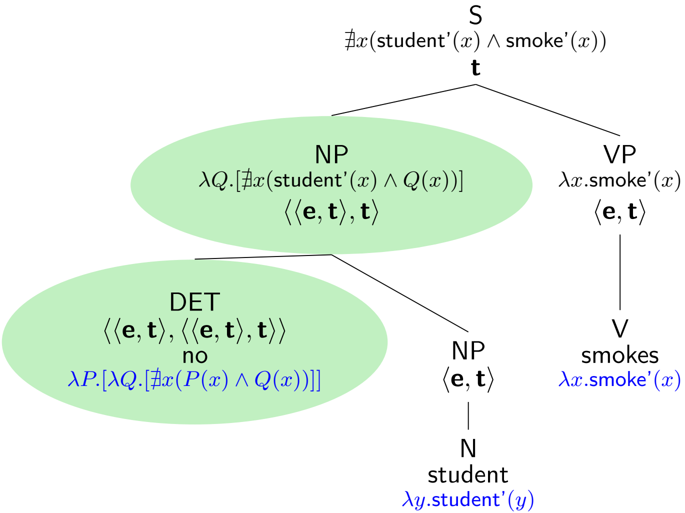
In order to do what they need to do (namely return a quantified NP of type $\langle\langle \mathbf{e}, \mathbf{t}\rangle, \mathbf{t}\rangle$), such quantifiers must be of type $\langle\langle \mathbf{e}, \mathbf{t}\rangle, \langle\langle \mathbf{e}, \mathbf{t}\rangle, \mathbf{t}\rangle\rangle$, which indicates that a quantifier identifies a relation between two sets.

Analysis from Lecture 8 (*every student*)



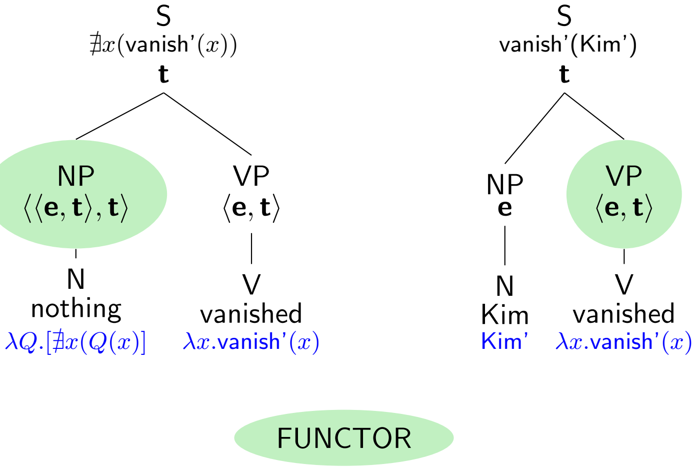
! Only Functional Application used

Analysis from Lecture 8 (*no student*)



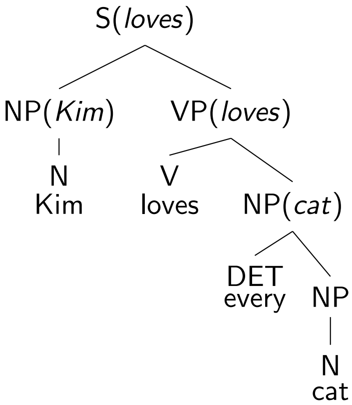
! Only Functional Application used

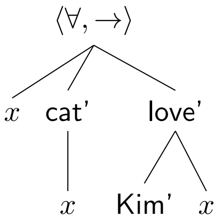
Nothing


S
$\nexists x(\text{vanish'}(x))$
t
NP
$\langle\langle \mathbf{e}, \mathbf{t}\rangle, \mathbf{t}\rangle$
N
nothing
$\lambda Q.[\nexists x(Q(x)]$
VP
$\langle \mathbf{e}, \mathbf{t}\rangle$
V
vanished
$\lambda x.\text{vanish'}(x)$
S
vanish'(Kim')
t
NP
e
N
Kim
Kim'
VP
$\langle \mathbf{e}, \mathbf{t}\rangle$
V
vanished
$\lambda x.\text{vanish'}(x)$
FUNCTOR

Syntax–semantics mismatch

(4) Kim loves every cat $\triangleright \forall x(\text{cat'}(x) \rightarrow \text{love'}(\text{Kim'}, x))$

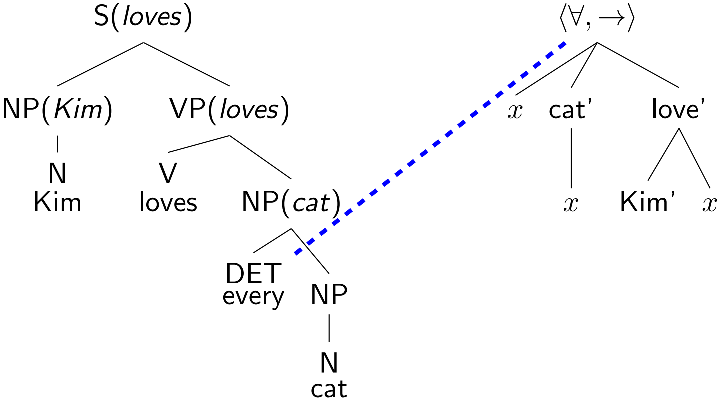




An alternative analysis of noun phrases: DET is the syntactic head.

Syntax–semantics mismatch

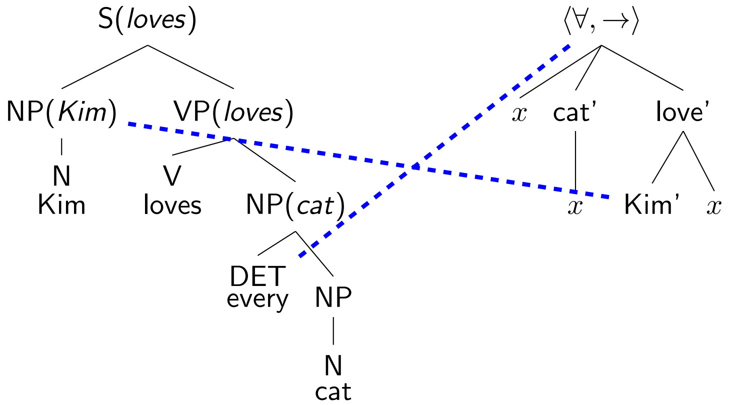
(4) Kim loves every cat $\triangleright \forall x(\text{cat'}(x) \rightarrow \text{love'}(\text{Kim'}, x))$



An alternative analysis of noun phrases: DET is the syntactic head.

Syntax–semantics mismatch

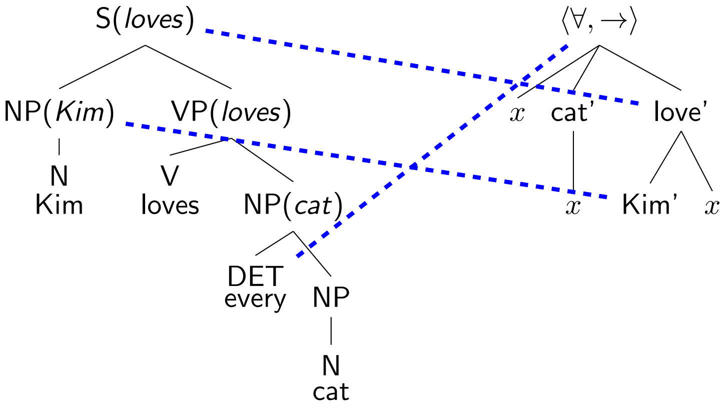
(4) Kim loves every cat $\triangleright \forall x(\text{cat'}(x) \rightarrow \text{love'}(\text{Kim'}, x))$



An alternative analysis of noun phrases: DET is the syntactic head.

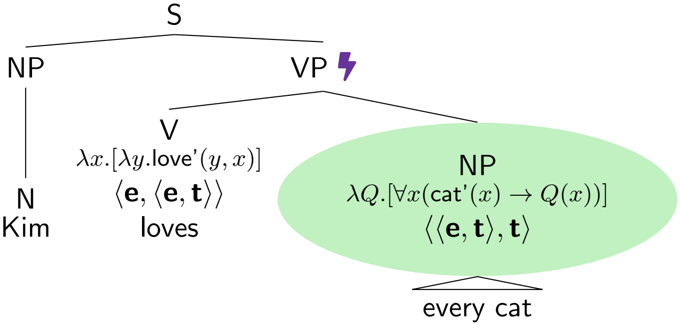
Syntax–semantics mismatch

(4) Kim loves every cat $\triangleright \forall x(\text{cat'}(x) \rightarrow \text{love'}(\text{Kim'}, x))$



An alternative analysis of noun phrases: DET is the syntactic head.

Problem with quantified NPs in object position



- Type mismatch
- VP: $\forall x(\text{cat'}(x) \rightarrow \lambda y.\text{love'}(y, x))$

Problem with quantified NPs in object position

$$\forall x(\text{cat'}(x) \rightarrow \text{love'}(\text{Kim'}, x))$$

"slot" for the expected subject

"semantic materials" correspond to *every cat*

"semantic materials" correspond to *loves*

$[\![$*every cat*$]\!]$ is separated into two parts

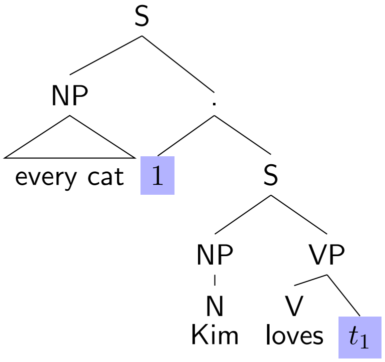
- an unbound variable x
- universal quantifier $\forall x(\text{cat'}(x) \rightarrow \ldots)$

We now need some heavy machinery

- Movement
- Traces
- Predicate abstraction rule for binding of traces
- Different shaped trees

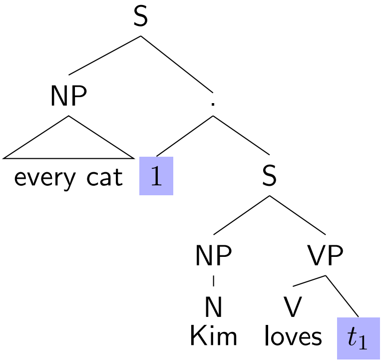
Movement and traces

What if in reality the tree looks like this:



Movement and traces

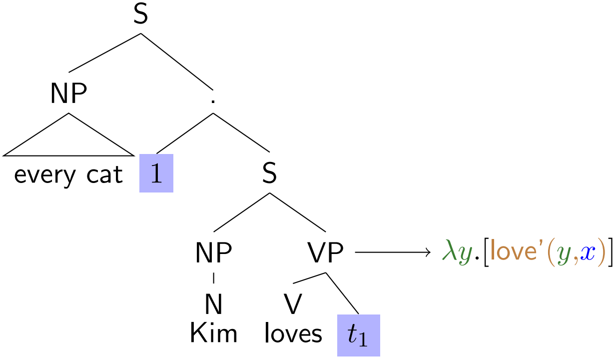
What if in reality the tree looks like this:



- When a constituent is moved, a trace (here: t_1) is left in its place. It's bound to its index (here: 1).

Movement and traces

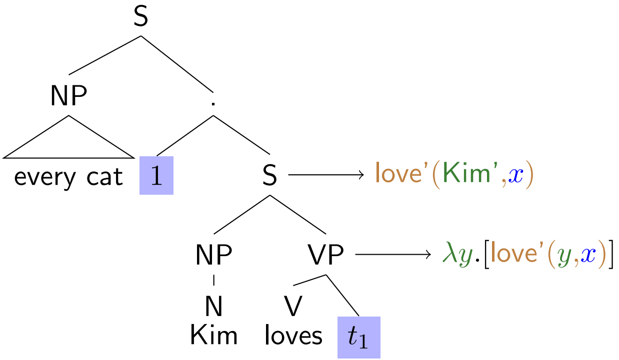
What if in reality the tree looks like this:



- When a constituent is moved, a trace (here: t_1) is left in its place. It's bound to its index (here: 1).

Movement and traces

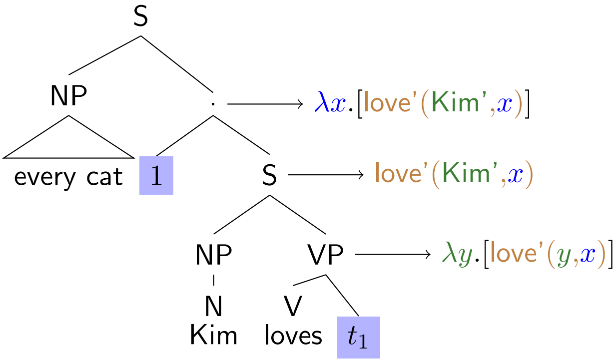
What if in reality the tree looks like this:



- When a constituent is moved, a trace (here: t_1) is left in its place. It's bound to its index (here: 1).

Movement and traces

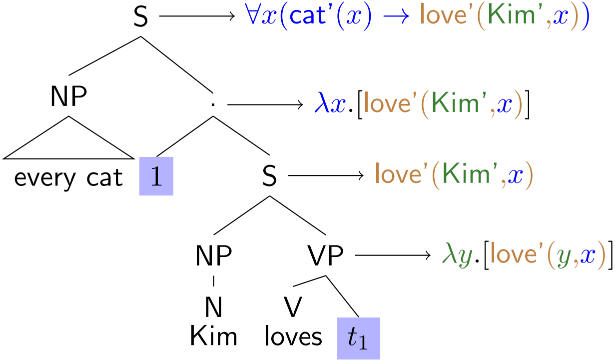
What if in reality the tree looks like this:



- When a constituent is moved, a trace (here: t_1) is left in its place. It's bound to its index (here: 1).
- What is the functionality of 1 ?
  Binding x – adding λx. This is function abstraction in λ-calculus.

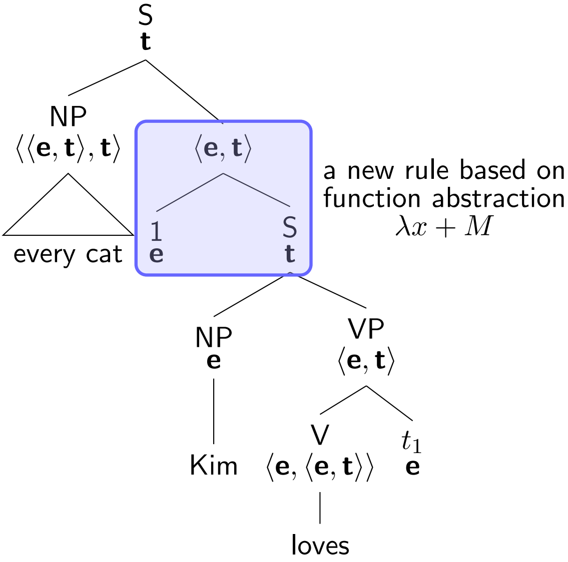
Movement and traces

What if in reality the tree looks like this:



- When a constituent is moved, a trace (here: t_1) is left in its place. It's bound to its index (here: 1).
- What is the functionality of 1 ?
  Binding x – adding λx. This is function abstraction in λ-calculus.

Now our types work out

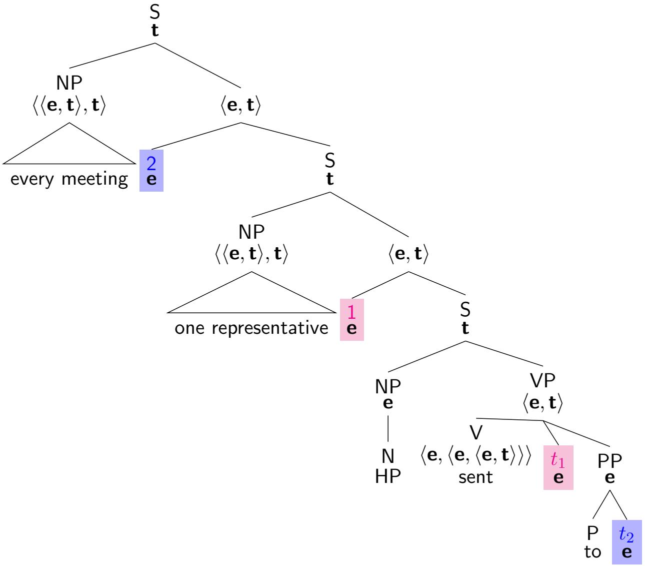


Heim and Kratzer, p. 112 and chapter 5.4 on Variable binding

Multiple quantification

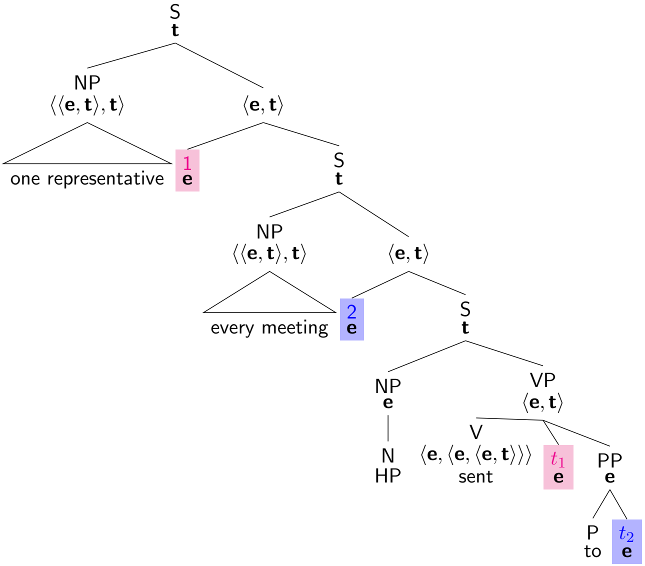
HP sent one representative to every meeting.

Double quantification under this analysis: Interpretation 1



$$\forall m(\ \exists r(\ \text{sent'}(e)\ \wedge \text{AGENT}(e, \text{hp'}) \wedge \text{THEME}(e, r) \wedge \text{RECIPIENT}(e, m)))$$

Double quantification under this analysis: Interpretation 2



$\exists r(\forall m(\text{sent'}(e) \wedge \text{AGENT}(e, \text{hp'}) \wedge \text{THEME}(e, r) \wedge \text{RECIPIENT}(e, m)))$

Interpretation under this world

- There is exactly one company, c.
- There are exactly two representatives, r_1 and r_2.
- There are exactly three meetings, m_1, m_2 and m_3.
- c sent r_1 to m_1, r_2 to both m_2 and m_3, and nobody else to anything else.

Which truth-value is assigned to the two interpretations on the previous pages under this world?

In-situ analysis vs. Movement analysis

- What we have just seen here is the movement analysis favoured by many Chomskyan Generative Linguists
- There is also an "in-situ" analysis
- In-situ means that the quantified NPs stay in their place
- The solution then involves two different types for quantified subject and object NPs
- CCG chose this solution
- MRS solves the problem with underspecification
- Contentious issue in Computational Linguistics
- Advantages and disadvantages for either

Presupposition and Quantifiers

Presupposition

(5) a. All American kings lived in New York.
    b. The vice-president is in the house.
    c. The twenty-five cats are in the kitchen.

Presupposition

(5) a. All American kings lived in New York.
    b. The vice-president is in the house.
    c. The twenty-five cats are in the kitchen.

Observation: Presupposition failure for a) in all cases, and for b) and c) if there aren't exactly one (salient) vice-president or twenty-five cats exactly.

So which of the following definitions of the semantics of "every" is correct?

- $\mathsf{F}_{\mathsf{every}} = \lambda \langle A, B \rangle : A \subseteq B$ (Theory 1)
- $\mathsf{F}_{\mathsf{every}} = \lambda \langle A, B \rangle : A \neq \emptyset \wedge A \subseteq B$ (Theory 2)

Presuppositional Hypothesis

Presuppositional hypothesis (H&K, page 163)

In natural languages, a lexical item δ with a denotation of type $\langle\langle \mathbf{e}, \mathbf{t}\rangle, \langle\langle \mathbf{e}, \mathbf{t}\rangle, \mathbf{t}\rangle\rangle$ is presuppositional

iff $\forall A \subseteq D, B \subseteq D$: if $A = \emptyset$, then $\langle A, B\rangle \notin dom(F_\delta)$

(6) All American kings lived in New York

This means that presupposition failure occurs if $A = \emptyset$ (there are no American kings)

Some doubt about Presuppositional Hypothesis

Speaker intuitions about the following sentences:

(7) a. No American king lived in New York.

b. Two American kings lived in New York.

and more problems:

(8) a. Every unicorn has exactly one horn.

b. All trespassers will be prosecuted.

Negative Scope

Negative scope

(9) a. You cannot not do this.
    b. You must/should do it

- Double negation is logically equivalent to positive statement
- Modulo focus effects; modulo presuppositions
- In some langugages, what looks like double negation is in fact a circumflex morpheme for single negation:

(10) a. I ain't seen no gun around here. (BAE)
     b. Je ne regrette rien (French)

Triple negation

Grand Designs, Episode "The Whirral 2016"

Types of Scope

Types of scope

We have so far seen quantifier scope and negative scope. Other kinds:

- modal scope
- "only" scope
- comparative scope
- contrastive scope (*rather than*)
- hypothetical scope
- attributive scope (*she said that...*)
- quotation scope (*so-called...*)
- ...

Problems with negation and modal scope

"Du musst nicht weinen." (= you needn't cry)

Informing of lack of need to cry?

Problems with negation and modal scope

English:

(11) a. you mustn't cry
must (not (cry))

b. you needn't cry
not (must (cry))

German:

(12) a. du musst nicht weinen
not (must (cry))

b. du darfst nicht weinen
must (not (cry))

Problems with negation and modal scope

English:

(11) a. you mustn't cry
must (not (cry))

b. you needn't cry
not (must (cry))

German:

(12) a. du musst nicht weinen
not (must (cry))

b. du darfst nicht weinen
must (not (cry))

A simple mistranslation

Somebody misheard something

(13) I could care less

What does that possibly mean?

Somebody misheard something

(13) I could care less

What does that possibly mean?

Comparative, modal and negation scope

“Only” scope

(14) a. Kim loved her cats.
b. Only Kim loved her cats.
c. Kim only loved her cats.
d. Kim loved only her cats.

- Comparison on some scope is involved
- “only” picks out the smaller situation
- Unless the scope is numerical or “big-small”, we need to infer the comparison ground.
- “only”’s smaller cousin is called “just”
- Writing tip

Trouble with "only" scope

(15) a. If the notice had **only** said 'mine-field' in Welsh as well as in English, . . .

b. **If only** the notice had said 'mine-field' in Welsh as well as in English, . . .

c. If the notice had only said (rather than signalled in Morse-code) 'mine-field' in Welsh as well as in English, . . .

Only meets not

(16) a. He didn't only see the cow, but also the bull

b. He only saw the cow, and not the bull

c. It is not the case that he saw only the cow and not the bull

d. He saw the cow and the bull

- We are told explicitly that it is not the case that the cow-seeing alone is in "only" scope.
- The bull-seeing also happened.
- This type of scope is closely related to the concept of **focus** (discourse lecture)
- " You might be thinking that it's more likely to see the cow, but hey, the bull was also seen."
- There is a "not" in the sentence, but neither the cow-seeing nor the bull-seeing are negated.

Contrastive scope

(17) a. **Instead of** using biaffine parse selection in subordinate structures, my system uses simple black magic.

b. In our interpretation of possible worlds, fictional characters are treated as semi-translucent slime, **rather than** as micron-thin gold plate, as Millovski (2013) does.

Part of the effect of contrastive scope is negation.

Writing tip

!

Avoid scope ambiguity when negative (or partially negative) scope is involved.

1. Recognise scopal properties of lexical items you want to use.
2. Move clauses which are under scope into positions where the scope is naturally bounded.

(18) I do X, rather than Y, which causes Z to happen.

Writing tip

> **!** Avoid scope ambiguity when negative (or partially negative) scope is involved.
>
> 1. Recognise scopal properties of lexical items you want to use.
> 2. Move clauses which are under scope into positions where the scope is naturally bounded.

(18) I do X, rather than Y, which causes Z to happen.

Whoa. Who did Z, you or the people who do Y?

Reformulations:

(19) a. Rather than doing Y, which would cause Z, I do X.
b. Rather than doing Y, I do X, which then causes Z to happen.
c. In order to avoid Z, I do X, rather than doing Y.
d. In order to achieve Z, I do X, rather than doing Y.

Special forcus on “careless i.e.”

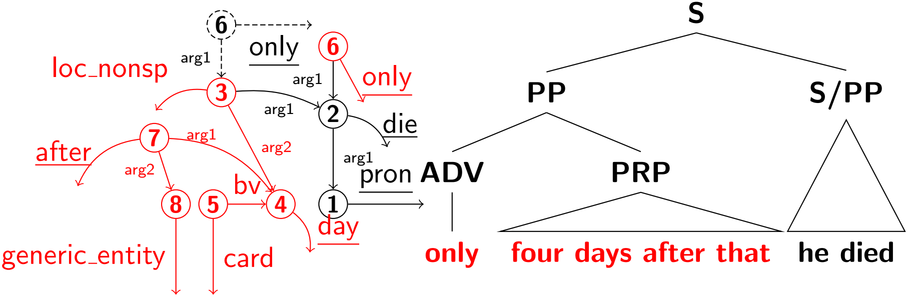
(20) I wouldn’t do X, i.e., do Y.

Whoa. Is Y negated or not?

Avoid careless “i.e.”

Representing Scope in CS

Scope


6
only
6
only
loc_nonsp
arg1
3
arg1
arg1
2
die
7
arg1
arg2
after
arg2
arg1
bv
pron
8
5
4
1
day
generic_entity
card
S
PP
S/PP
ADV
PRP
only
four days after that
he died

Reading

- Heim and Kratzer (1999):
    - Chapter 6 and 7 for quantifiers and scope
    - Chapter 5 for traces and Predicate Abstraction
- Reading for next time: Arcs of Coherence; chapter 5 from Pinker (2014)