Semantic pre-processing of anaphoric references

Marie Duží
VŠB-Technical University Ostrava
marie.duzi@vsb.cz

Using TIL (of course)

(S) "5 + 7 = 12, and Charles knows ***it***."

- But Charles doesn't know T. He knows that the ***procedure*** of calculating 5+7=12 yields T.
- Thus the meaning of (S) is a two-phase instruction that comes down to this:

i. ***Pre-processing*** (of the meaning of the embedded clause): execute *the substitution based on the **meaning** of the antecedent (here: 5+7=12) for the anaphoric variable (here: it);*

ii. ***Execute*** *the result (a propositional construction) again to obtain a proposition*.

Method of analysis

(S) "5 + 7 = 12, and Charles knows ***it***."

a) *Type-theoretical analysis.*

5,7,12/τ; τ is the type of real numbers.

+/(τττ); the function of adding that maps number-pairs (ττ) to a number (of type τ).

=/(οττ); the relation of identity on numbers.

∧/(οοο); conjunction: mapping couples of truth values (type (οο)) to truth values (ο).

Charles/ι; ι is the type of individuals.

Know/(οι$*_1$)$_{\tau\omega}$; the relation-in-intension (τω) of an individual (ι) to a construction (procedure of type $*_1$).

it/$*_2$→ $*_1$; the anaphoric variable.

Sub/($*_2$ $*_2$ $*_2$ $*_2$)—the substitution function: applied on constructions C_1, C_2, C_3, it yields a construction C_4 that is the result of substituting C_1 *for* C_2 *into* C_3.

Method of analysis

(S) “5 + 7 = 12, and Charles knows ***it***.”

b) <u>*Synthesis.*</u>

‘5+7=12’ → $[{}^0{=}\ [{}^0{+}\ {}^05\ {}^07]\ {}^012]$;

‘Charles knows it’ → $\lambda w\, \lambda t\, [{}^0Know_{wt}\ {}^0Charles\ it]$;

Gloss: ***Constructions*** are instructions ***(procedures)***;

Atomic constructions (supply entities that the composed constructions operate on):
Trivialisation (${}^0{=},\ {}^0{+},\ {}^05,\ {}^07,\ {}^012$), *Variables* (*x, p, it, he, …*);

Composed constructions consist of constituents:

Composition (like $[{}^0{+}\ x\ {}^01]$), *Closure* ($\lambda x\, [{}^0{+}\ x\ {}^01]$), *Double Execution* ${}^2C$;

(S) → $\lambda w\, \lambda t\, [[{}^0{=}[{}^0{+}\ {}^05\ {}^07]\ {}^012] \wedge$
$\quad {}^2[{}^0Sub\ {}^{00}[{}^0{=}[{}^0{+}\ {}^05\ {}^07]\ {}^012]\ {}^0it$
$\quad\quad {}^0[\lambda w\, \lambda t\, [{}^0Know_{wt}\ {}^0Charles\ it]]]_{wt}]$

Method of analysis

a) *Pre-processing* (1st execution):

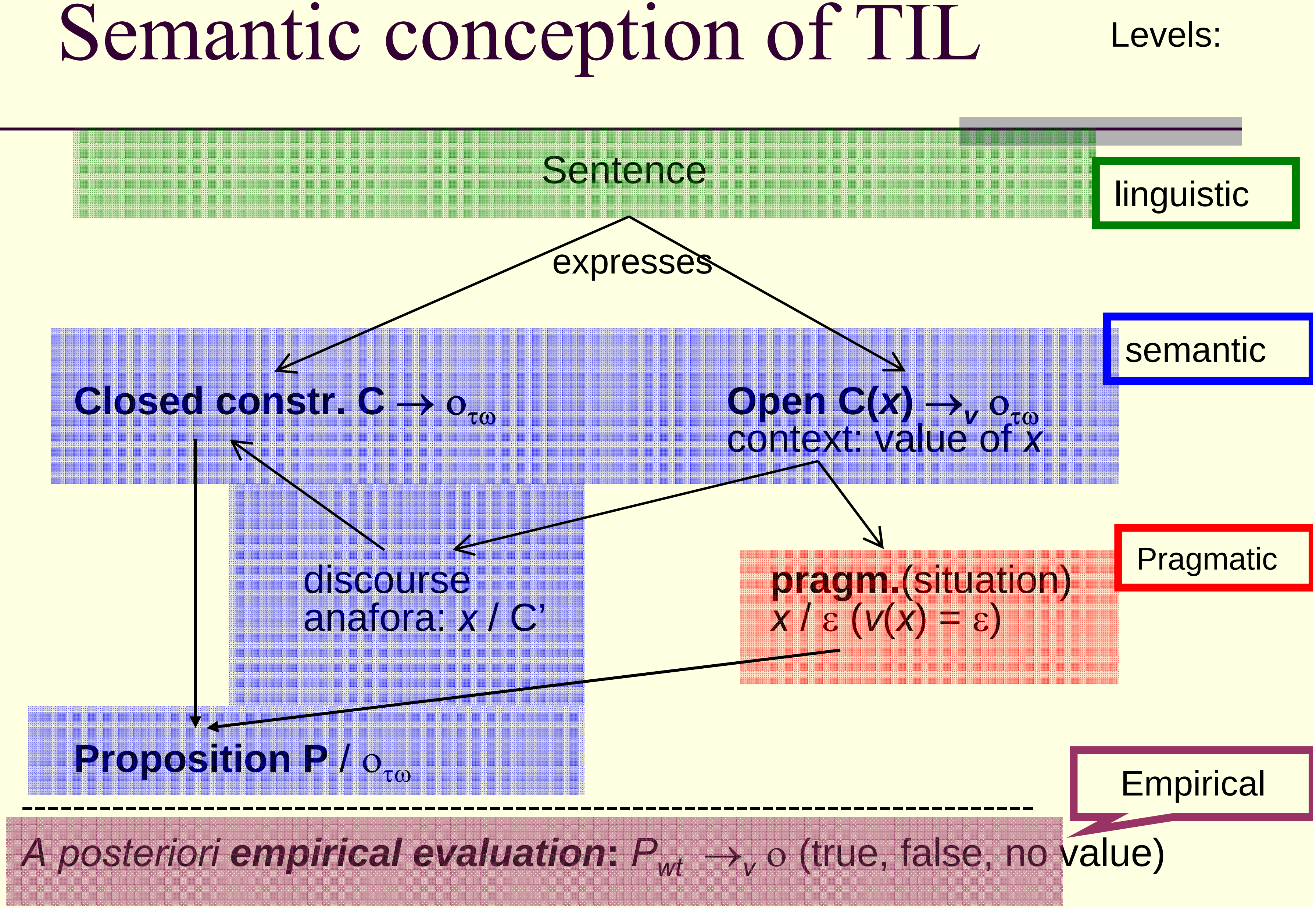
Sub(stitutes) the meaning of the antecedent $({}^0[{}^0{=}[{}^0{+}\ {}^05\ {}^07]\ {}^012])$ *for the anaphoric variable (it) into the meaning of the embedded clause* $([\lambda w\, \lambda t\, [{}^0Know_{wt}\ {}^0Charles\ it]])$;

b) *2nd Execution*:

of the adjusted (now closed) embedded clause constructs the proposition (that Charles knows …)

c) *Intensional descent*: $[[C\ w]\ t]$, $\mathbf{C_{wt}}$, *yields the truth value*

Semantic conception of TIL



de re attitudes & anaphora

"Charles is looking for the Mayor of Dunedin".

- understood as uttered in a situation where Charles knows who the Mayor is, and is striving to *locate* this individual:

"**Charles is looking for the Mayor of Dunedin, namely the location of *him***".

- $\lambda w \lambda t\ [{}^0 Look\_for_{wt}\ {}^0 Charles$ ${}^2[{}^0 Sub\ [{}^0 Tr\ [{}^0 Mayor\_of_{wt}\ {}^0 D]]\ {}^0 him\ {}^0[\lambda w \lambda t\ [{}^0 Loc_{wt}\ him]]]]$

  $\underbrace{[{}^0 Tr\ [{}^0 Mayor\_of_{wt}\ {}^0 D]]}$

  v-constructs Trivialisation of the individual (if any) who occupies the Mayor office.

Types: *Look_for*/$(o\iota\mu_{\tau\omega})_{\tau\omega}$; *Tr*/$(*_1\iota)$; *Charles*/$\iota$; *Mayor_of* /$(\iota\iota)_{\tau\omega}$; *D*(unedin)/$\iota$; *him*/$*_1 \rightarrow \iota$; *Loc*/$(\mu\iota)_{\tau\omega}$.

Donkey sentences

- **“If somebody has got a new car then *he* often washes *it*.”**
- Embedded clause: $\lambda w \lambda t\ [{}^0Wash_{wt}\ he\ it]$
  - $\lambda w \lambda t\ [{}^0Freq_t\ \lambda t'[{}^0Wash_{wt'}\ he\ it]]$
- *Problem*: how to understand the sentence?
- If somebody owns more than one new car, does he wash *all* or *some* of them?

Donkey sentences

Peter **Geach** (*NC* new car):

$\forall x \forall y\ ((NC(y) \wedge Has(x, y)) \rightarrow Wash(x, y))$.

Bertrand **Russell** ('a new car' is an *indefinite description*:

$\forall x\ (\exists y\ (NC(y) \wedge Has(x, y)) \rightarrow Wash(x, y))$.

But the last occurrence of the variable *y* (marked in red) is free in this formula—out of the scope of the existential quantifier supposed to bind it.

Neale (*restricted quantifiers*):

[every *x*: man *x* and [a *y*: new-car *y*](*x* owns *y*)]([whe *z*: car *z* and *x* owns *z*] (*x* often washes *z*)).

Neale takes into account that the sentence is true even if a man owns *more than one* new car: his abbreviation 'whe *F*' stands for 'the *F* or the *F*s'.

Donkey sentences

(D1) "**Anybody who owns some new cars often washes *all of them*** [each of the new cars he owns]."

(D2) "**Anybody who owns some new cars often washes *some of them*** [some of the new cars he owns]."

- *Types*: *Own*/$(o\iota\iota)_{\tau\omega}$; *Wash*/$(o\iota\iota)_{\tau\omega}$; *NC* (being a new car)/$(o\iota)_{\tau\omega}$; *x*, *y*, *he, it* $\rightarrow \iota$.
- We need another type of quantifiers: *Some, All*

Some, All quantifiers

- *Some*, *All* *of type* ((ο(οι))(οι)).
- *Some* is a function that associates the argument—a set S—with the set of all those sets which have a non-empty intersection with S.
- *All* is a function that associates the argument—a set S—with the set of all those sets which contain S as a subset.
- $\lambda w \lambda t\, [[{}^0Some\ {}^0Student_{wt}]\ {}^0Happy_{wt}]$.

Donkey sentences

(D1'')

$$\lambda w \lambda t\ [^0\forall \lambda x\ [[[^0Man_{wt}\ x] \wedge$$
$$[^0\exists \lambda y\ [[^0NC_{wt}\ y] \wedge [^0Own_{wt}\ x\ y]]]] \supset$$
$$^2[^0Sub\ ^0[\lambda y\ [[^0NC_{wt}\ y] \wedge [^0Own_{wt}\ x\ y]]]\ ^0them$$
$$[^0Sub\ ^0x\ ^0he$$
$$^0[\lambda w' \lambda t'\ [[^0All\ them]\ \lambda it\ [^0Wash_{w't'}\ he\ it]]]]]_{wt}]].$$

- Gloss: "For every man, if the man owns some new cars then all of them [i.e., the new cars owned] are washed by him [the man x]."

Donkey sentences

(D2'')

$\lambda w \lambda t\ [{}^0\forall \lambda x\ [[[{}^0Man_{wt}\ x] \wedge$

$[{}^0\exists \lambda y\ [[{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]] \supset$

${}^2[{}^0Sub\ {}^0[\lambda y\ [[{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]\ {}^0them$

$[{}^0Sub\ {}^0x\ {}^0he$

${}^0[\lambda w' \lambda t'\ [[{}^0Some\ them]\ \lambda it\ [{}^0Wash_{w't'}\ he\ it]]]]]_{wt}]].$

- Gloss: "For every man, if the man owns some new cars then some of them [i.e., the new cars owned] are washed by him [the man x]."

Compositional analysis

Gabriel Sandu formulates (1997):

- there is a one-to-one mapping of the surface structure of a sentence of (a fragment of) English into its logical form which preserves the left-to-right ordering of the logical constants
- the mapping preserves the nature of the lexical properties of the logical constants, in the sense that ***an indefinite is translated by an existential quantifier***, etc.

Compositional analysis (of D)

In the interest of disambiguation, we actually analysed two variants of the original sentence (a man vs. every man).

The analysis of the antecedent clause "***A man has a new car***" should be as follows:

(NC) $\lambda w \lambda t\ [{}^0\exists \lambda xy\ [[{}^0Man_{wt}\ x] \wedge [{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]$.

- Additional type: $\exists/(o(o\iota\iota))$.

The consequent of (D) expresses that ***all* the couples <*he*, *it*> are such that *he Washes it.*** Using a variable *couples*/$*_1 \rightarrow (o\iota\iota)$, we have:

(AC) $\lambda w \lambda t\ [[{}^0All\ couples]\ \lambda he\ it\ [{}^0Wash_{wt}\ he\ it]]$.

Compositional analysis (of D)

- Composing (NC) with (AC), we substitute the set of couples … for the variable *couples*:

(D’) $\lambda w \lambda t\ [[{}^0\exists \lambda xy\ [[{}^0Man_{wt}\ x] \wedge [{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]] \supset$
${}^2[{}^0Sub\ {}^0[\lambda xy\ [[{}^0Man_{wt}\ x] \wedge [{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]\ \ {}^0couples$
${}^0[\lambda w \lambda t\ [[{}^0All\ couples]\ \lambda he\ it\ [{}^0Wash_{wt}\ he\ it]]]]_{wt}].$

The Dynamic Predicate Logic (DPL): passing on binding.

(D’) – the semantics of this mechanism.

Variables he and *it* are bound in (D’), but not directly bound by the existential quantifier.

Technically, they are bound by Trivialization; semantically, they are bound by the condition that the pairs of individuals they *v*-construct have to belong to the set mentioned by the antecedent clause.

Discourse Representation Theory (Kamp & Reyle)

- dynamic interpretation of natural language, where each sentence is interpreted within a certain discourse, which is a sequence of sentences uttered by the same speaker.
- the problem of anaphoric links crossing the sentence boundary.
- *Pressing question*: to determine the respective antecedent to which the anaphoric pronoun refers.
- first-order theory; only expressions denoting individuals introduce the so-called *discourse referents*, i.e., free variables that are updated when interpreting the discourse.

TIL and Discourse Representation

- TIL: higher-order, procedural
  - not only individuals, but ***entities of any type***, like *properties*, *propositions*, *relations-in-intension*, and even *constructions* can be ***linked to anaphoric variables***.
  - Moreover, strong typing makes it possible to ***determine*** the respective ***type-appropriate antecedent(s)***.

Outline of Implementation method (first proposed by J. Křetínský)

- For each type (ι, (οι)τω, οτω, (οι(οι)τω)τω, (οιι)τω, ∗*n*, …) we create the ***list of free discourse variables.*** *They are dynamic (program-like) variables.*
- The method substitutes their content for anaphoric (logical) variables to complete the meaning of anaphoric clauses.
- Each closed constituent of a resulting construction becomes an updated value of the respective (type-appropriate) free discourse-referent variable.
- In this way the discourse variables are gradually updated.

Example: dialogue between *Adam*, *Berta* and *Cecil*.

- *Adam to Cecil*: "Berta is coming. ***She*** is looking for a parking".

'Inform' message content:

$\lambda w \lambda t\ [[{}^0Coming_{wt}\ {}^0Berta]$.

- (Relevant) discourse variables updates:

  ind*:=⁰*Berta; *pred*:=^{0}*Coming*; *prop*:= $\lambda w \lambda t\ [[{}^0Coming_{wt}\ {}^0Berta]$;

$\lambda w \lambda t\ {}^2[{}^0Sub\ \mathbf{ind}\ \mathbf{{}^0she}\ {}^0[{}^0Looking\_for_{wt}\ she\ {}^0Parking]] \Rightarrow$ (is transformed into)

$\lambda w \lambda t\ [{}^0Looking\_for_{wt}\ {}^0Berta\ {}^0Parking]$.

- (Relevant) discourse variables updates:

  *rel*1:= ^{0}*Looking_for*; *pred*:=^{0}*Parking*;

  prop:= $\lambda w \lambda t\ [{}^0Looking\_for_{wt}\ {}^0Berta\ {}^0Parking]$;

  prof:= $\lambda w \lambda t\ \lambda x\ [{}^0Looking\_for_{wt}\ x\ {}^0Parking]$;

Example: dialogue between *Adam*, *Berta* and *Cecil*.

- *Cecil to Adam*: "***So*** am I."

$\lambda w \lambda t\ {}^2[{}^0Sub\ \mathbf{prof}\ {}^0\mathbf{so}\ {}^0[so_{wt}\ {}^0Cecil]] \Rightarrow$
$\lambda w \lambda t\ [{}^0Looking\_for_{wt}\ {}^0Cecil\ {}^0Parking]$.

- (Relevant) discourse variables updates:

$ind := {}^0Cecil$; $rel_1 := {}^0Looking\_for$; $pred := {}^0Parking$;

- *Adam to both*: "There is a free parking at p_1".

$\lambda w \lambda t\ \exists x\ [[[{}^0Free\ {}^0Parking]_{wt}\ x] \wedge [{}^0At_{wt}\ x\ {}^0p_1]]$

- (Relevant) discourse variables updates:

$loc := {}^0p_1$; $pred := [{}^0Free\ {}^0Parking]$;

$prop := \lambda w \lambda t\ \exists x\ [[[{}^0Free\ {}^0Parking]_{wt}\ x] \wedge [{}^0At_{wt}\ x\ {}^0p_1]]$.

Example: dialogue between *Adam*, *Berta* and *Cecil*.

- *Berta to Adam*: “What do you mean by free parking?”

‘***Query***’ message content:

$$\lambda w \lambda t\ [{}^0Refine_{wt}\ {}^0[{}^0Free\ {}^0Parking]]$$

- (Relevant) discourse variables updates:

  $constr := {}^0[{}^0Free\ {}^0Parking]$

- *Adam to Berta*: “Free parking is a parking and some parts of it are not occupied”.

‘***Reply***’ message content:

$$[{}^0Free\ {}^0Parking]] =$$

$$[\lambda w \lambda t\ \lambda x\ [[{}^0Parking_{wt}\ x] \wedge$$

$$\exists y\ [[{}^0Part\_of_{wt}\ y\ x] \wedge \neg[{}^0Occupied_{wt}\ y]]]]$$

Example: dialogue between *Adam*, *Berta* and *Cecil*.

- *Berta to Adam*: “I don’t believe ***it***. I have just been ***there***”.

‘Inform’ message content (first sentence):

$\lambda w \lambda t\ [^2[^0Sub\ \mathbf{prop}\ {}^0\mathbf{it}\ {}^0[\neg[^0Believe_{wt}\ {}^0Berta\ it]]] \Rightarrow$

$\lambda w \lambda t\ \neg[^0Believe_{wt}\ {}^0Berta\ [\lambda w \lambda t\ \exists x\ [[[^0Free\ {}^0Parking]_{wt}\ x]$
$\wedge\ [^0At_{wt}\ x\ {}^0p_1]]].$

‘Inform’ message content (second sentence):

$\lambda w \lambda t\ \exists t'[[t' \leq t] \wedge$

$^2[^0Sub\ \mathbf{loc}\ {}^0\mathbf{there}\ {}^0[^0Been\_at_{wt'}\ {}^0Berta\ there]]] \Rightarrow$

$\lambda w \lambda t\ \exists t'[[t' \leq t] \wedge [^0Been\_at_{wt'}\ {}^0Berta\ {}^0p_1]].$

Concluding remarks

- Due to the procedural semantics, our ***agents can learn new concepts*** by asking the other agents.
- In our example, after receiving Adam's reply Berta learns the refined meaning of the 'free parking' predicate, i.e., she updates her knowledge base by the respective composed construction.
- Moreover, though our approach is as fine-grained as the syntactic approach of standard FIPA languages like KIF, the ***content of agent's knowledge is not a piece of syntax***, but its ***meaning*** (i.e., TIL construction).
- And since the respective construction is what synonymous expressions (even of different languages) have in common, the ***agents*** *(should)* ***behave in the same way independently of the language*** in which their knowledge and ontology is expressed.
- For instance, if we switch to Czech, the underlying constructions are ***identical***: $^0[^0\mathit{Free}\ ^0\mathit{Parking}] = {}^0[^0\mathit{Volné}\ ^0\mathit{Parkoviště}]$.

Concluding remarks

- Of course, improvements of the above method are straightforward.
- For instance, in the example we were substituting the last type-appropriate entity that received mention;
- If we wanted to take into account ambiguities of anaphoric references, we might store into the discourse-representation file more than one variable for each type,
- together with the other characteristics or prerequisites of entities (e.g., gender, or implied properties), so as to be able to generate more meanings of an ambiguous sentence.

Concluding remarks

- The problem of an anaphoric reference to a previously used expression is a well-known hard nut of *linguistic* analysis, because the antecedent of the anaphoric reference is often not unambiguously determined.
- Thus it is often said that anaphora constitutes a *pragmatic* problem rather than a problem of logical semantics.
- We agree that *logical* analysis cannot disambiguate any sentence, because it presupposes understanding and full linguistic competence.
- Yet our method of logical analysis *contributes to solving the problem of disambiguation* in at least two respects:
  - (a) the type-theoretical analysis often unambiguously determines which of the possible meanings of a homonymous expression is used in a sentence, and
  - (b) if there are two or more possible readings of a sentence, the logical analysis should make all of them explicit. This often concerns the distinction between *de dicto* and *de re* readings.
- In this sense anaphora *is* a *semantic* problem.

Thank you for your attention

- Questions, answers, comments, …