

# Copredication, dynamic generalized quantification and lexical innovation by coercion

Robin Cooper  
Göteborg University

The Department of Linguistics

COMPUTER SCIENCE AND ENGINEERING



GÖTEBORG UNIVERSITY  
Faculty of Arts



# Contents

<b>1 Copredication and dot types</b>	<b>3</b>
<b>2 Nondistributive predication and dot types</b>	<b>12</b>
<b>3 Records and record types</b>	<b>16</b>
<b>4 Dynamic generalized quantifiers</b>	<b>22</b>
<b>5 Treating copredication</b>	<b>26</b>
<b>6 Treating lexical innovation</b>	<b>32</b>
<b>7 Conclusion</b>	<b>42</b>
⇐ contents	2/43

# 1. Copredication and dot types

## Dot types in the Generative Lexicon

- compositions of two types which nevertheless allow the two individual types to be recovered
- used by Pustejovsky in *The Generative Lexicon*
- type theoretical approach proposed by Asher and Pustejovsky (2005)

## Lunch as food or an event

- (1) a. The lunch was delicious
- b. The lunch took forever

*delicious* – normally a predicate of food but not events

- (2) a. The blancmange was delicious
- b. <sup>i</sup>IFK Göteborg's last game was delicious

- Generative Lexicon – coercion
- Grammatical Framework (Aarne Ranta) – resource grammars (libraries)
- “English as a formal language” → “English as a toolbox for constructing formal languages”

## *take forever* – a predicate of events, not of food

- (3) a. <sup>i</sup>The blancmange took forever
- b. IFK's last game took forever

## Is *lunch* polysemous?

lunch<sub>1</sub> : *FoodPred*

lunch<sub>2</sub> : *EventPred*

## Copredication

- (4) a. The lunch was delicious but took forever
- b. <sup>!</sup>The bank specializes in IPO's and is being quickly eroded by the river

## Coerced joins of alternative readings

- (5) <sup>i</sup>Sam went to the wrong bank
- (6) Kim told me about the wrong lunch
- (7) Kim told me the wrong things about the lunch. (I was interested in the food, not the conversation.)

## 2. Nondistributive predication and dot types

## The dot-type analysis

- (8) a. A lunch was delicious
- b.  $\exists x : Food \exists v : Food \cdot Event [lunch(v) \wedge$   
 $O\text{-Elab}(x, v) \wedge delicious(x)]$

*Alternative:*

- A lunch is both food *and* an event in virtue of having parts which are (purely) food and (purely) an event.
- Complex objects inherit properties from their parts (or at least we talk as if they do).

*Consequence:* We can use *meet* (conjunction) instead of *dot* and *O-elab*.

## Predications of objects in virtue of predications of parts

- (9) a. The house is locked
  - b. The door/lock is locked
  
- (10) a. The choir/singers sang the Hallelujah Chorus. Mildred had a tickle in her throat and didn't sing a note.
  - b. The (players of the) Gothenburg Symphony Orchestra played Mahler's second and a Mozart symphony. There were many more people playing for the Mahler.

# Genericity, mereology and inexact predicates

*Boston drivers are bad* (Greg Carlson)

- (11) The GSO played Elgar's Intro and Allegro for Strings
- (12) The GSO played a solo violin partita by Bach

## 3. Records and record types

# Records and record types

If  $a_1 : T_1, a_2 : T_2(a_1), \dots, a_n : T_n(a_1, a_2, \dots, a_{n-1})$ ,  
 the record:

$$\left[ \begin{array}{l} l_1 = a_1 \\ l_2 = a_2 \\ \dots \\ l_n = a_n \\ \dots \end{array} \right]$$

is of type:

$$\left[ \begin{array}{l} l_1 : T_1 \\ l_2 : T_2(l_1) \\ \dots \\ l_n : T_n(l_1, l_2, \dots, l_{n-1}) \end{array} \right]$$

*a man owns a donkey*

Record type:

$$\left[ \begin{array}{l} x : Ind \\ c_1 : \text{man}(x) \\ y : Ind \\ c_2 : \text{donkey}(y) \\ c_3 : \text{own}(x,y) \end{array} \right]$$

Record:

$$\left[ \begin{array}{l} x = a \\ c_1 = p_1 \\ y = b \\ c_2 = p_2 \\ c_3 = p_3 \end{array} \right]$$

where

$a, b$  are of type *Ind*, individuals

$p_1$  is a proof of  $\text{man}(a)$

$p_2$  is a proof of  $\text{donkey}(b)$

$p_3$  is a proof of  $\text{own}(a, b)$

[⇐ contents](#)

## Records are recursive

$$r = \left[ \begin{array}{l} f = \left[ \begin{array}{l} f = \left[ \begin{array}{l} ff = a \\ gg = b \end{array} \right] \\ g = c \end{array} \right] \\ g = \left[ \begin{array}{l} h = \left[ \begin{array}{l} g = a \\ h = d \end{array} \right] \end{array} \right] \end{array} \right]$$

is of type

$$R = \left[ \begin{array}{l} f : \left[ \begin{array}{l} f : \left[ \begin{array}{l} ff : T_1 \\ gg : T_2 \end{array} \right] \\ g : T_3 \end{array} \right] \\ g : \left[ \begin{array}{l} h : \left[ \begin{array}{l} g : T_1 \\ h : T_4 \end{array} \right] \end{array} \right] \end{array} \right]$$

given that  $a : T_1$ ,  $b : T_2$ ,  $c : T_3$  and  $d : T_4$ .

## Path names

We can use *path-names* in records and record types to designate values in particular fields, e.g.

$$r.f = \left[ \begin{array}{l} f = \left[ \begin{array}{l} ff = a \\ gg = b \end{array} \right] \\ g = c \end{array} \right]$$

$$R.f.f.ff = T_1$$

## Functions ( $\lambda$ -calculus)

*donkey*

$\lambda r: [x:Ind] ([c:donkey(r.x)])$

:

$([x:Ind])RecType$

*cf.* Montague's  $\langle e, t \rangle$

## 4. Dynamic generalized quantifiers

## Generalized quantifiers

*the donkey runs*

(13)  $[ c : \text{the}(\lambda r:[x:Ind])([c_1:\text{donkey}(r.x)]), \lambda r:[x:Ind]([c_2:\text{run}(r.x)]) ) ]$

# Polymorphic generalized quantifiers

Used in the treatment of dynamic generalized quantifiers.

If  $q$  is a quantifier predicate then

1.  $\text{arity}(q) = \langle (X \sqsubseteq [x:\text{Ind}])\text{RecType}, (X \sqsubseteq [x:\text{Ind}])\text{RecType} \rangle$
2.  $q$  is associated with a relation between sets  $q^*$  (the relation between sets from classical generalized quantifier theory) such that

$$p : q(f_1 : (T_1)\text{RecType}, f_2 : (T_2)\text{RecType})$$

iff

$$p = \langle \{a \mid \exists r : T_1[f_1(r) \text{ is non-empty} \wedge a = r.x]\}, \\ \{a \mid \exists r : T_2[f_2(r) \text{ is non-empty} \wedge a = r.x]\} \rangle$$

and  $q^*$  holds between  $p_1$  and  $p_2$

## Dynamic generalized quantifiers

$$(14) \quad q(f_1 : (T) \text{RecType}, f_2 : (\mathcal{F}(f_1)) \text{RecType})$$

where:  $\mathcal{F}(f)$  is the *fixed-point type* of  $f$ , i.e.  $a : \mathcal{F}(f) \rightarrow a : f(a)$

$$(15) \quad \left[ c : \text{the}(\lambda r : [x : \text{Ind}]([c_1 : \text{donkey}(r.x)])), \lambda r : \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{donkey}(x) \end{array} \right] ([c_2 : \text{run}(r.x)]) \right]$$

*Conservativity*: The donkey runs  $\leftrightarrow$  The donkey is a donkey which runs

## 5. Treating copredication

## Additional constraints on arguments

*take forever*

$$(16) \quad \lambda r: \left[ \begin{array}{l} x : Ind \\ c_1: \text{event}(x) \end{array} \right] ([c_2: \text{take\_forever}(r.x)])$$

*be delicious*

$$(17) \quad \lambda r: \left[ \begin{array}{l} x : Ind \\ c_1: \text{food}(x) \end{array} \right] ([c_2: \text{be\_delicious}(r.x)])$$

## Predicate conjunction

*be delicious and take forever*

$$(18) \quad \lambda r: \left[ \begin{array}{l} x : Ind \\ c_1:food(x) \\ c_2:event(x) \end{array} \right] \left( \left[ \begin{array}{l} c_3:be\_delicious(r.x) \\ c_4:take\_forever(r.x) \end{array} \right] \right)$$

## Additional constraints on nouns

(19) a.  $\lambda r: \left[ \begin{array}{l} x : Ind \\ c_1: \text{food}(x) \end{array} \right] ([c_2: \text{blancmange}(r.x)])$

b.  $\lambda r: \left[ \begin{array}{l} x : Ind \\ c_1: \text{event}(x) \end{array} \right] ([c_2: \text{game}(r.x)])$

c.  $\lambda r: \left[ \begin{array}{l} x : Ind \\ c_1: \text{food}(x) \\ c_2: \text{event}(x) \end{array} \right] ([c_3: \text{lunch}(r.x)])$

## Dynamic quantification

*the game took forever*

$$(20) \quad \text{the}(\lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{event}(x) \end{array} \right] ([c_2 : \text{game}(r.x)]), \\ \lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{event}(x) \\ c_2 : \text{game}(x) \end{array} \right] ([c_3 : \text{took\_forever}(r.x)]))$$

## Dynamic quantification and *lunch*

- (21) a.  $\text{the}(\lambda r: \begin{bmatrix} x : \textit{Ind} \\ c_1 : \text{food}(x) \\ c_2 : \text{event}(x) \end{bmatrix} ([c_3 : \text{lunch}(r.x)])),$
- $\lambda r: \begin{bmatrix} x : \textit{Ind} \\ c_1 : \text{food}(x) \\ c_2 : \text{event}(x) \\ c_3 : \text{lunch}(x) \end{bmatrix} ([c_4 : \text{took\_forever}(r.x)]))$
- b.  $\text{the}(\lambda r: \begin{bmatrix} x : \textit{Ind} \\ c_1 : \text{food}(x) \\ c_2 : \text{event}(x) \end{bmatrix} ([c_3 : \text{lunch}(r.x)])),$
- $\lambda r: \begin{bmatrix} x : \textit{Ind} \\ c_1 : \text{food}(x) \\ c_2 : \text{event}(x) \\ c_3 : \text{lunch}(x) \end{bmatrix} ([c_4 : \text{be\_delicious}(r.x)]))$

## 6. Treating lexical innovation

## The game was delicious

$$(22) \quad \text{the}(\lambda r: \begin{bmatrix} x : \text{Ind} \\ c_1 : \text{event}(x) \end{bmatrix}) (\begin{bmatrix} c_2 : \text{game}(r.x) \end{bmatrix}),$$

$$\lambda r: \begin{bmatrix} x : \text{Ind} \\ c_1 : \text{event}(x) \\ c_2 : \text{game}(x) \end{bmatrix} (\begin{bmatrix} c_3 : \text{be\_delicious}(r.x) \end{bmatrix}))$$

Lexical resource for *be delicious*:

$$\lambda r: \begin{bmatrix} x : \text{Ind} \\ c_1 : \text{food}(x) \end{bmatrix} (\begin{bmatrix} c_2 : \text{be\_delicious}(r.x) \end{bmatrix})$$

$$\begin{bmatrix} x : \text{Ind} \\ c_1 : \text{event}(x) \\ c_2 : \text{game}(x) \end{bmatrix} \not\sqsubseteq \begin{bmatrix} x : \text{Ind} \\ c_1 : \text{food}(x) \end{bmatrix}$$

The use of *be delicious* is *innovative* with respect to this resource.

## Coercion and dynamic quantification

- coercion in *the game was delicious* results from the general treatment of dynamic quantification
- but what would an appropriate proof object be?

## Proof objects for non-mathematical types

- Martin-Löf – Proof objects for mathematical propositions (as types)
- Ranta – events as proof objects for non-mathematical propositions (as types), e.g. a proof of *Amundsen flew over the North Pole* is a (Davidsonian) event of flying by Amundsen over the North Pole
- Ranta’s “problem” – natural language predicates like *man* and *tree* do not correspond to sets in the constructive mathematical sense
- the “problem” seems greater with predicates like *delicious* which is both vague and subjective

## Ranta's three strategies

- work with *types* rather than sets
- develop techniques of *approximation*
- study delimited *models* of language use

It seems to me that all three of these should be developed in an adequate type-theoretical approach to natural language.

In particular, ...

## Inexact types

- It seems important that we recognize that human beings reason with *inexact* types for which they do not have (or indeed there may not even exist) clearly defined procedures for determining whether objects belong to those types or not.
- Such types may be refined by successive approximations as a result of being exposed to natural language utterances in a variety of situations.
- We have the ability to “tie down” certain expressions in a given restricted domain, “for the purposes of discussion” or at least to know what certain complex expressions would mean under the assumption that exact interpretations were assigned to their constituent parts.

## Inexact types in innovative uses

- $be\_delicious(g)$  for some game  $g$
- the type itself may be well-defined as an object
- but not the objects (proof-objects) which belong to the type
- we may accept as being sufficient information for present purposes
- or we may clarify: *what do you mean “delicious”?*

# Importance of inexact types for innovation

- inexactitude plays an important role in allowing innovation
- *? the game was divisible by three*

## Resource based coercions

*The blancmange took forever*

(23) *exploiting dynamic quantification*

$$\begin{array}{l}
 ?\text{the}(\lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{food}(x) \end{array} \right] ([c_2 : \text{blancmange}(r.x)]), \\
 \lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{food}(x) \\ c_2 : \text{blancmange}(x) \end{array} \right] ([c_3 : \text{took\_forever}(r.x)]))
 \end{array}$$

(24) *exploiting qualia in the resource lexicon*

$$\begin{array}{l}
 \text{the}(\lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{event}(x) \end{array} \right] ([c_2 : \text{preparation\_of\_blancmange}(r.x)]), \\
 \lambda r: \left[ \begin{array}{l} x : \text{Ind} \\ c_1 : \text{event}(x) \\ c_2 : \text{preparation\_of\_blancmange}(x) \end{array} \right] ([c_3 : \text{took\_forever}(r.x)]))
 \end{array}$$

## Preferred readings and degrees of innovation

Exploiting resources yields a preferred reading with a robust intuition that it is “less innovative” than non-resource based coercions.

## 7. Conclusion

## Record types and lexical resources

- record types as an alternative to dot types in the treatment of copredication
- connection between dynamic generalized quantification and the coercion involved in innovative uses of linguistic predicates
- GL and GF: lexical resources in natural languages viewed as toolboxes for constructing formal languages
- degrees of innovation
- variability and dialogue management