Compositionality Problems and How to Solve Them
Thomas Ede Zimmermann, Goethe-Universität Frankfurt
Hyderabad CogSci Colloquium, March 09

1. Compositionality ...

The V of a complex expression functionally depends on the Vs of its immediate parts and the way in which they are combined ...where V is a semantic value

Ordinary Principle of Compositionality  Montague (1970)
The meaning of a complex expression functionally depends on the meanings of its immediate parts and the way in which they are combined.

Extensional Principle of Compositionality  Frege (1892)
The extension of a complex expression functionally depends on the extensions of its immediate parts and the way in which they are combined.

Intensional Principle of Compositionality  Kaplan (1989)
The content of a complex expression functionally depends on the contents of its immediate parts and the way in which they are combined.

Parts vs. \textbf{immediate parts}

\begin{itemize}
  \item \textbf{John loves Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{John}
  \item \textbf{loves Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{loves}
  \item \textbf{Mary}
\end{itemize}

Individuation of expressions: disambiguation

\begin{itemize}
  \item \textbf{John or Jane and Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{John or Jane}
  \item \textbf{and Mary}
\end{itemize}

Simplification: binarity

... by coordination as modification

\begin{itemize}
  \item \textbf{John or Jane and Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{John}
  \item or \textbf{Jane and Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{or}
  \item \textbf{Jane and Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{Jane}
  \item and \textbf{Mary}
\end{itemize}

\begin{itemize}
  \item \textbf{and}
  \item \textbf{Mary}
\end{itemize}
... or syncategorematic coordination

Montague (1973)

\[
\begin{align*}
\text{John or Jane and Mary} \\
\text{John} & \quad \text{Jane and Mary} \\
\text{Jane} & \quad \text{Mary}
\end{align*}
\]

Level of syntactic analysis:

\[
\begin{align*}
\text{John seeks a unicorn} \\
\text{John} & \quad \text{seeks a unicorn} \\
\text{seeks} & \quad \text{a unicorn}
\end{align*}
\]

\[
\begin{align*}
\text{John or Jane and Mary} \\
\text{John} & \quad \text{seeks a unicorn} \\
\text{a unicorn} & \quad \text{John seeks } u \\
\text{seeks } u
\end{align*}
\]

Top-down strategy for determining semantic values:

- Find suitable ('cofinal') set of expressions.
- Assign values to members.
- Fill in gaps applying suitable strategies.

Frege (1884; 1892)

Hodges (2001)

2. Problems ...

3 types of problems to be encountered in analysing expressions of the form

\[
\begin{align*}
\text{WHOLE} \\
\text{LEFT} & \quad \text{RIGHT}
\end{align*}
\]

- **Type A**
  GIVEN: value of whole
  NEEDED: values of both parts
  e.g.:
  \[
  \begin{align*}
  \text{[Mary coughed]} & \quad \checkmark \\
  \text{[Mary]} & \quad \text{[coughed]}?
  \end{align*}
  \]

- **Type B**
  GIVEN: value of whole and of one part
  NEEDED: values of other part
  e.g.:
  \[
  \begin{align*}
  \text{[a unicorn coughed]} & \quad \checkmark \\
  \text{[seeks a unicorn]} & \quad \checkmark \\
  \text{[a unicorn]} & \quad \text{[coughed]}? \quad \text{or \ [seeks]}? \quad \text{[a unicorn]}?
  \end{align*}
  \]

- **Type C**
  GIVEN: value of whole and of one part
  NEEDED: combination of semantic values of parts
  e.g.:
  \[
  \begin{align*}
  \text{[read a book]} & \quad \checkmark \\
  \text{[read]} & \quad \text{[a book]}?
  \end{align*}
  \]
A compositionality problem is *solvable* just in case there is a way of replacing all ? by √ without changing any √.

*Observations*  
Type A problems are always solvable.

A Type B problem is solvable iff  
\[
\llbracket \text{RIGHT}_i \rrbracket = \llbracket \text{RIGHT}_j \rrbracket \text{ implies: } \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket
\]

[or:  
\[
\llbracket \text{LEFT}_i \rrbracket = \llbracket \text{LEFT}_j \rrbracket \text{ implies: } \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket
\]

for all \( i \) and \( j \).

A Type C problem is solvable iff  
\[
\llbracket \text{RIGHT}_i \rrbracket = \llbracket \text{RIGHT}_j \rrbracket \text{ implies: } \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket
\]

and:  
\[
\llbracket \text{LEFT}_i \rrbracket = \llbracket \text{LEFT}_j \rrbracket \text{ implies: } \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket
\]

for all \( i \) and \( j \).

3. … and How to Solve Them  
*General Strategies for Unsolvable (and Solvable) Compositionality Problems*  
• Redefine syntactic input.  
*Applications:*  
– Type C (unsolvable), creating another, *solvable* Type C problem:  

  From:  
  \[
  \llbracket \text{every linguist from India} \rrbracket \checkmark
  \]

  \[
  \llbracket \text{every linguist} \rrbracket \checkmark \quad \llbracket \text{from India} \rrbracket \checkmark
  \]

  to:  
  \[
  \llbracket \text{every linguist from India} \rrbracket \checkmark
  \]

  \[
  \llbracket \text{every} \rrbracket \checkmark \quad \llbracket \text{linguist from India} \rrbracket \checkmark
  \]

  \[
  \llbracket \text{linguist} \rrbracket \checkmark \quad \llbracket \text{linguist from India} \rrbracket \checkmark
  \]

– Type C (solvable), but creating more *Type B* and *Type C* problems…  
May (1985), Heim & Kratzer (1998)  

\[
\llbracket \text{read a book} \rrbracket \checkmark
\]

\[
\llbracket \text{a book} \rrbracket \checkmark \quad \llbracket \text{read } y \rrbracket ?
\]

\[
\llbracket y \rrbracket ? \quad \llbracket \text{read } y \rrbracket ?
\]

\[
\llbracket \text{read} \rrbracket \checkmark \quad \llbracket y \rrbracket ?
\]
• Replace semantic values by more fine-grained ones:
Type B (unsolvable):
If $[[X]]$ is $X$'s extension, then:

(*)

$$[[\text{Jones believes Smith is sick}}]] \checkmark$$

$$[[\text{Jones}}]\checkmark \quad [[\text{believes Smith is sick}}]\checkmark$$

$$[[\text{believes}}]? \quad [[\text{Smith is sick}}]\checkmark$$

is unsolvable. Replacing $[[\text{Smith is sick}}]$ by the intension $[[\text{Smith is sick}}]$ renders (*) solvable.

General Strategies for Solvabe Compositionality Problems

• **Strategy A:**

  Frege (1884)

  Find covariation between one part and some other entity, and take the latter to be the former’s semantic value.

More precisely, given

(L)

$$\text{WHOLE}_1 \quad \text{WHOLE}_2 \quad \text{WHOLE}_3 \quad \ldots \quad \text{WHOLE}_i \quad \ldots$$

$$\text{LEFT}_i \quad \text{RIGHT} \quad \text{LEFT}_2 \quad \text{RIGHT} \quad \text{LEFT}_3 \quad \text{RIGHT} \quad \text{LEFT}_i \quad \text{RIGHT}_i$$

[or:] (R)

$$\text{WHOLE}_1 \quad \text{WHOLE}_2 \quad \text{WHOLE}_3 \quad \ldots$$

$$\text{LEFT} \quad \text{RIGHT}_i \quad \text{LEFT} \quad \text{RIGHT}_2 \quad \text{LEFT} \quad \text{RIGHT}_3 \quad \text{LEFT} \quad \text{RIGHT}_i$$

find objects $x_i$ such that:

$$[[\text{WHOLE}_i]] = [[\text{WHOLE}_j]] \text{ just in case } [[\text{LEFT}_i]] = [[\text{LEFT}_j]]$$

[or] $$[[\text{WHOLE}_i]] = [[\text{WHOLE}_j]] \text{ just in case } [[\text{RIGHT}_i]] = [[\text{RIGHT}_j]]$$

Then put:

$$[[\text{LEFT}_i]] : = x_i \text{ [or } [[\text{RIGHT}_i]] : = x_i]$$

Application:

$$[[\text{Mary coughed}}]] \checkmark$$

$$[[\text{Mary}}] = \text{Mary} \quad [[\text{coughed}}]? \quad [[\text{Mary coughed}}]] \checkmark$$

$$[[\text{Mary}}]? \quad [[\text{coughed}}] = \text{set of coughers}$$
• **Strategy B**: Frege (1892); cf. Kupffer (2008); Zimmermann (in prep.)

Determine primary occurrences of other expressions and construct their values as contributions in primary occurrences. More precisely, given

\[
\left[ \text{RIGHT}_1 \right], \left[ \text{RIGHT}_2 \right], \ldots \text{ and } \left[ \text{WHOLE}_1 \right], \left[ \text{WHOLE}_2 \right], \ldots
\]

construct:

\[
\left[ \text{RIGHT}_1 \right] \left[ \text{WHOLE}_1 \right] \quad \text{or:} \quad \left[ \text{LEFT}_1 \right] \left[ \text{WHOLE}_1 \right]
\]

\[
\left[ \text{RIGHT}_2 \right] \left[ \text{WHOLE}_2 \right] \quad \left[ \text{LEFT}_2 \right] \left[ \text{WHOLE}_2 \right]
\]

\[
\ldots \left[ \text{RIGHT}_i \right] \left[ \text{WHOLE}_i \right] \quad \ldots \left[ \text{LEFT}_i \right] \left[ \text{WHOLE}_i \right]
\]

and put \( \left[ \text{LEFT} \right] \) := \( f \) such that:

\[
f(\left[ \text{RIGHT} \right]) = \left[ \text{WHOLE} \right] \quad \text{or } f(\left[ \text{LEFT} \right]) = \left[ \text{WHOLE} \right]
\]

**Application:**

\[
\begin{array}{c}
\left[ \text{a unicorn coughed} \right] \checkmark \\
\left[ \text{a unicorn} \right] = f \quad \left[ \text{coughed} \right] \checkmark \\
\end{array}
\]

where:

\[
f(\left[ \text{coughed} \right]) = \left[ \text{a unicorn coughed} \right],
\]

\[
f(\left[ \text{neighed} \right]) = \left[ \text{a unicorn neighed} \right], \text{ etc.}
\]

If \( \left[ X \right] \) is \( X \)'s extension, it turns out that:

\[
f = \lambda P. \vdash P \cap U \neq \emptyset \vdash
\]

\[
\left[ \text{seeks a unicorn} \right] \checkmark
\]

where:

\[
f(\left[ \text{a unicorn} \right]) = \left[ \text{seeks a unicorn} \right]
\]

\[
f(\left[ \text{a horse} \right]) = \left[ \text{seeks a horse} \right], \text{ etc.}
\]

If \( \left[ X \right] \) is \( X \)'s extension, then:

\[
\left[ \text{a unicorn} \right] = \left[ \text{a ghost} \right]
\]

and so:

\[
f(\left[ \text{a unicorn} \right]) = f(\left[ \text{a ghost} \right])
\]

**BUT:**

\[
f(\left[ \text{a unicorn} \right]) = \left[ \text{seeks a unicorn} \right] = \{ x \mid x \ \text{seeks a unicorn} \}
\]

\[
\neq \{ x \mid x \ \text{seeks a ghost} \} = \left[ \text{seeks a ghost} \right] = f(\left[ \text{a ghost} \right])
\]

\[
\Rightarrow \text{ NO EXTENSIONAL SOLUTION!}
\]

If \( \left[ X \right] \) is \( X \)'s intension, then:

\[
f(\left[ \text{a unicorn} \right])(i) = \{ x \mid x \ \text{seeks a unicorn at index } i \}
\]

\[
f(\left[ \text{a horse} \right])(i) = \{ x \mid x \ \text{seeks a horse at } i \}
\]

\[
f(\left[ \text{a ghost} \right])(i) = \{ x \mid x \ \text{seeks a horse at } i \}
\]

unclear how (and even: whether) value depends on argument

\[
\Rightarrow \text{ additional theory needed}
\]

\[\text{cf. Zimmermann (2005: 2.3) [see Appendix]}\]
• **Strategy C:**

Define combination $F$ by collecting all instances:

$$F([\text{LEFT}], [\text{RIGHT}]) = [\text{WHOLE}]$$

and find pattern.

**Application:**

$$F([\text{read}], [\text{a book}]) = [\text{read a book}]$$

$$F([\text{read}], [\text{every book}]) = [\text{read every book}]$$

$$F([\text{buy}], [\text{a book}]) = [\text{buy a book}]$$

etc.

If $[X]$ is $X$'s extension, we have:

$$F([(x,y) \mid x \text{ reads } y], \lambda P. \vdash P \cap \emptyset \neq 0 \rightarrow) = \{x \mid \{y \mid x \text{ reads } y\} \cap \emptyset \neq 0\}$$

$$F([(x,y) \mid x \text{ reads } y], \lambda P. \vdash P \subseteq \emptyset \neq 0 \rightarrow) = \{x \mid \emptyset \subseteq \{y \mid x \text{ reads } y\}\}$$

$$F([(x,y) \mid x \text{ buys } y], \lambda P. \vdash P \cap \emptyset \neq 0 \rightarrow) = \{x \mid \{y \mid x \text{ buys } y\} \cap \emptyset \neq 0\}$$

etc.

– the pattern being:

$$F([\text{LEFT}], [\text{RIGHT}]) = [\text{RIGHT}]([x \mid \{y \mid (x,y) \in [\text{LEFT}]\}])$$

**References**


2.3. California. Summer of love. Montague, using techniques of higher-order modal logic, turned Quine’s account of opacity into a surfacecompositional analysis\(^4\), the starting point of which is a possible worlds adaptation of Quine’s paraphrase formulated in intensional type theory; under the assumption that try-for-it-to-be-the-case-that denotes a binary relation between individuals and propositions, seek a unicorn receives the following logical analysis:\(^5\)

\[
(5) \quad \lambda x \mathbf{T}(x, (\exists y)[U(y) \land F(x, y)])
\]

Surface compositionality, i.e. a word-by-word analysis, is then achieved by applying the Fregean strategy of meaning subtraction\(^6\), obtaining the meaning of seek by separating the quantifier expressed by a unicorn from the property denoted by seek a unicorn. Such a separation can be carried out using a series of lambda-abstractions:

\[
(5) \quad \equiv \quad \lambda x \mathbf{T}(x, [\lambda Q (\exists y)[U(y) \land Q(y)]][\lambda y F(x, y)])
\]

\[
\equiv \quad [\lambda \mathcal{Q} \lambda x \mathcal{T}(x, \mathcal{Q} (\lambda y F(x, y)))](\lambda Q (\exists y)[U(y) \land Q(y)])
\]

The first step isolates the (underlined) meaning of the indefinite a unicorn as contributing to the (varying) implicit attitude object denoted by find a unicorn; the second step separates that meaning from the rest, which in turn may serve as an analysis of seek. Given the types in (5), it turns out that seek denotes a relation between an individual and a quantifier, i.e. its meaning is of type \(((\text{et})t)(\text{et})\), viz.:

\[
(6) \quad \lambda \mathcal{Q} \lambda x \mathbf{T}(x, (\mathcal{Q} y)F(x, y))
\]

where “(\(\mathcal{Q} y\))\(\phi\)” abbreviates “\(\mathcal{Q}(\lambda y \phi)\)”. Since the resulting type is independent of the paraphrase, Montague’s analysis is more general than Quine’s:

As far as “seeks” and “owes” are concerned, circumlocution involving infinitives is possible. It is not, however, in the case of all English verbs sharing the logical peculiarities of “seeks” and “owes” […]

Montague (1969: 177)