Compositionality Problems and How to Solve Them
Thomas Ede Zimmermann, Frankfurt

1. Compositionality...

Generalised Principle of Compositionality
The V of a complex expression functionally depends on the Vs of its immediate parts and the way in which they are combined.

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

(1) John loves Mary

(2) The boss is asleep

(3) John or Jane and Mary

(4) John or Jane and Mary
2. Problems ...

(7) 
\[
\text{WHOLE} \\
\text{LEFT} \quad \text{RIGHT}
\]

(8) 
Type 0:
\[
\text{WHOLE} \checkmark \\
\text{LEFT} \quad \text{RIGHT}?
\]

Type 1:
\[
\text{WHOLE} \checkmark \\
\text{LEFT} \checkmark \quad \text{RIGHT}?
\]
\[
\text{WHOLE} \checkmark \\
\text{LEFT}? \quad \text{RIGHT} \checkmark
\]

Type 2:
\[
\text{WHOLE} \checkmark \\
\text{LEFT} \checkmark \quad \text{RIGHT} \checkmark
\]

(9a) \[[\text{Mary is coughing}] = \{(w,t) \mid \text{Mary is coughing in world } w \text{ at time } t\}

(b) \[[\text{The boss is laughing}] = \{(w,t) \mid \text{whoever is the boss in } w \text{ at } t, \text{ is coughing in } w \text{ at } t\}

(c) \[[\text{Sentence}] \subseteq I

(10a) \[[\text{Mary is coughing}] = \Gamma[[\text{Mary}],[[\text{is coughing}]])

(b) \[[\text{The boss is laughing}] = \Gamma[[\text{the boss}],[[\text{is laughing}]})

(c) \[[\text{Sentence}] = \Gamma([[\text{Term}],[[\text{Predicate}])}]}
(11) 

Sentence \( \sqrt{\text{Term} \ ? \ \text{Predicate} \ ?} \)

(12) 

Sentence \( \sqrt{\text{Term} \ ? \ \text{Predicate} \ ?} \)

(13) \[ [\text{Mary}] (w,t) = \text{Mary} \]

(b) \[ [\text{the boss}] (w,t) = \text{the boss in } w \text{ at } t \]

(c) \[ [\text{Term}] : I \rightarrow \wp (I) \]

(14a) \[ [\text{is coughing}] (w,t) = \{ x \mid x \text{ is coughing in } w \text{ at } t \} \]

(b) \[ [\text{is laughing}] (w,t) = \{ x \mid x \text{ is laughing in } w \text{ at } t \} \]

(c) \[ [\text{Predicate}] : I \rightarrow \wp (U) \]

(15) \[ \Gamma ([\text{Term}], [\text{Predicate}]) = \{ (w,t) \mid [\text{Term}] (w,t) \in [\text{Predicate}] (w,t) \} \]

(16) Everyone is shouting.

(17) 

Sentence \( \sqrt{\text{QuantifierPhrase} \ ? \ \text{Predicate} \ ?} \)

(18a) \[ [\text{kill}] = \{ (x,y) \in U^2 \mid x \text{ kills } y \} \]

(b) \[ [\text{introduce}] = \{ (x,y,z) \in U^3 \mid x \text{ introduces } z \text{ to } y \} \]

(c) \[ [\text{Verb}_n] \subseteq U^n \]

(19a) \[ [\text{John loves Mary}] = \Gamma ([\text{John}], [\text{loves Mary}]) = 1 \]

(b) \[ [\text{Nobody loves Mary}] = \Gamma ([\text{every boy}], [\text{loves Mary}]) = 0 \]

(c) \[ [\text{Sentence}] = \Gamma ([\text{NounPhrase}], [\text{Verb}], [\text{Verb}], [\text{Verb}]) \subseteq \{ \emptyset \} \]

(20) 

Sentence \( \sqrt{\text{NounPhrase} \ ? \ \text{Verb} \ ?} \)

(21a) \[ [\text{John}] = \{ X \subseteq U \mid \text{John } \in X \} \]

(b) \[ [\text{nothing}] = \{ \emptyset \} \]

(c) \[ [\text{NounPhrase}] \subseteq \wp (U) \]

(22) \[ \Gamma ([\text{NounPhrase}], [\text{Predicate}]) = \begin{cases} 1, & \text{if } [\text{NounPhrase}] \in [\text{Predicate}] \\ 0, & \text{if } [\text{NounPhrase}] \notin [\text{Predicate}] \end{cases} \]
(23) \[\llbracket \text{every semanticist} \rrbracket = \Gamma(\llbracket \text{every} \rrbracket, \llbracket \text{semanticist} \rrbracket)\]

(24) \[\text{QuantifierPhrase} \checkmark\]

\[\begin{align*}
\text{Determiner} & \quad ? \\
\text{Noun} & \quad ?
\end{align*}\]

(25a) John says it is raining.
(b) Most experts believe Mary will win the election.

(26) \[\begin{align*}
\text{Verb} & \quad \checkmark \\
\text{Verb}_\text{cc} & \quad ? \\
\text{Sentence} & \quad \checkmark
\end{align*}\]

(27a) \[\llbracket \text{says it is raining} \rrbracket = \Gamma(\llbracket \text{says} \rrbracket, \llbracket \text{says it is raining} \rrbracket)\]
(b) \[\llbracket \text{believe Mary will win} \rrbracket = \Gamma(\llbracket \text{believe} \rrbracket, \llbracket \text{Mary will win} \rrbracket)\]
(c) \[\llbracket \text{Verb}_1 \rrbracket = \Gamma(\llbracket \text{Verb}_\text{cc} \rrbracket, \llbracket \text{Sentence} \rrbracket)\]

(28) no linguist from India

(29) \[\begin{align*}
\text{NounPhrase} & \quad \checkmark \\
\text{NounPhrase}^\prime & \quad \checkmark \\
\text{PrepositionalPhrase} & \quad ?
\end{align*}\]

(30a) \[\llbracket \text{no linguist from India} \rrbracket = \Gamma(\llbracket \text{no linguist} \rrbracket, \llbracket \text{from India} \rrbracket)\]
(b) \[\llbracket \text{every pope from India} \rrbracket = \Gamma(\llbracket \text{every pope} \rrbracket, \llbracket \text{from India} \rrbracket)\]
(c) \[\llbracket \text{NounPhrase} \rrbracket = \Gamma(\llbracket \text{NounPhrase}^\prime \rrbracket, \llbracket \text{PrepositionalPhrase} \rrbracket)\]

(31) \[\llbracket \text{every pope} \rrbracket = \{X \subseteq U \mid \{p\} \subseteq X\} = \{X \subseteq U \mid \{p\} \cap X \neq \emptyset\} = \llbracket \text{some pope} \rrbracket\]

(32) \[\begin{align*}
\text{Verb} & \quad \checkmark \\
\text{Verb}_2 & \quad \checkmark \\
\text{NounPhrase} & \quad \checkmark
\end{align*}\]

(33) \[\Gamma(\llbracket \text{Verb}_2 \rrbracket, \llbracket \text{NounPhrase} \rrbracket)\]
\[= \{x \in U \mid \exists y \in U \mid (x, y) \in \llbracket \text{Verb}_2 \rrbracket \} \in \llbracket \text{NounPhrase} \rrbracket\]

(34a) Jones is looking for a sweater.
(b) Jones painted a unicorn.

(34a) If a farmer owns a donkey, he beats it.
(b) Every farmer who owns a donkey beats it.
3. ... and How to Solve Them
A compositionality problem is solvable just in case there is a way of replacing all \( ? \) by \( \sqrt{\_} \) without changing any \( \sqrt{\_} \).

**Observations**

Type 0 problems are always solvable.

A Type 1 problem is solvable iff

\[
\llbracket \text{RIGHT}_i \rrbracket = \llbracket \text{RIGHT}_j \rrbracket \implies \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket,
\]

[or:

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

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

A Type 2 problem is solvable iff

\[
\llbracket \text{RIGHT}_i \rrbracket = \llbracket \text{RIGHT}_j \rrbracket \implies \llbracket \text{WHOLE}_i \rrbracket = \llbracket \text{WHOLE}_j \rrbracket
\]

and:

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

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

**General Strategies for Unsolvable (and Solvable) Compositionality Problems**

- **Syntactic Solution**: Redefine input.

**Applications**:

- **Type 1** (unsolvable), creating another, solvable Type 1 problem:

  From:

  \[
  \llbracket \text{every linguist from India} \rrbracket \sqrt{\_}
  \]

  \[
  \llbracket \text{every linguist} \rrbracket \sqrt{\_} \quad \llbracket \text{from India} \rrbracket ?
  \]

  to:

  \[
  \llbracket \text{every linguist from India} \rrbracket \sqrt{\_}
  \]

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

  \[
  \llbracket \text{linguist} \rrbracket \sqrt{\_} \quad \llbracket \text{from India} \rrbracket ?
  \]

- **Type 2** (solvable), but creating more Type 0 and Type 1 problems...

  May (1985), Heim & Kratzer (1998)
• **Ontological Solution**: Replace semantic values by more fine-grained ones.  

**Applications**:

- **Type 1** (unsolvable):  
  From:  
  
  \[
  \begin{array}{c}
  \text{believes Smith is sick} \text{ } \checkmark \\
  \text{believes} \text{ } ? \text{ } \text{Smith is sick} \text{ } \checkmark \\
  \end{array}
  \]

  (where \([X]\) is \(X\)'s extension) to the solvable Type 1 problem:
  
  \[
  \begin{array}{c}
  \text{believes Smith is sick} \text{ } \checkmark \\
  \text{believes} \text{ } ? \text{ } \text{Smith is sick} \text{ } \checkmark \\
  \end{array}
  \]

  (where \([X]\) is suitably fine-grained: *sense, intension, …*).

- **Type 1** (unsolvable):  
  From:  
  
  \[
  \begin{array}{c}
  \text{If a farmer owns a donkey, he beats it} \text{ } \checkmark \\
  \text{a farmer owns a donkey} \text{ } \checkmark \text{ } \text{he beats it} \text{ } ? \\
  \end{array}
  \]

  (where \([X]\) is \(X\)'s extension) to the solvable Type 2 problem:
  
  \[
  \begin{array}{c}
  \text{If a farmer owns a donkey, he beats it} \text{ } \checkmark \\
  \text{a farmer owns a donkey} \text{ } \checkmark \text{ } \text{he beats it} \text{ } \checkmark \\
  \end{array}
  \]

  (where \([X]\) is suitably fine-grained: *relation, context change potential, …*).

**General Strategies for Solvable Compositionality Problems**

• **Strategy 0**:  
  Find covariation between one part and some other entity, and take the latter to be the former’s semantic value.

More precisely, given

(L)

\[
\begin{array}{cccccccc}
\text{WHOLE}_1 & \text{WHOLE}_2 & \text{WHOLE}_3 & \ldots & \text{WHOLE}_i & \ldots \\
\text{LEFT}_1 & \text{RIGHT} & \text{LEFT}_2 & \text{RIGHT} & \text{LEFT}_3 & \text{RIGHT} \\
\end{array}
\]

[or:

(R)

\[
\begin{array}{cccccccc}
\text{WHOLE}_1 & \text{WHOLE}_2 & \text{WHOLE}_3 & \ldots \\
\text{LEFT} & \text{RIGHT}_1 & \text{LEFT} & \text{RIGHT}_2 & \text{LEFT} & \text{RIGHT}_3 \\
\end{array}
\]

find objects \(x_i\) such that:
In case \( \text{LEFT}_i = \text{LEFT}_j \) just in case \( \text{LEFT}_i = \text{LEFT}_j \)

\[ \text{or} \quad \text{WHOLE}_i = \text{WHOLE}_j \]  just in case \( \text{RIGHT}_i = \text{RIGHT}_j \)

Then put:
\[
\text{LEFT}_i := x_i \quad \text{or} \quad \text{RIGHT}_i := x_i
\]

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

\[\text{or:} \]
\[
\text{Mary coughed} \quad \checkmark
\]
\[
\text{Mary} \quad ? \quad \text{coughed} = \text{set of coughers}
\]
\[
\text{every} \quad ? \quad \text{book} = \text{set of books}
\]

**Strategy 1:**
Frege (1892); cf. Kupffer (2008); Zimmermann (in prep.)
Determine primary occurrences of valueless expressions and construct their values as contributions in primary occurrences. More precisely, given
\[
\text{right}_1, \text{right}_2, \ldots \quad \text{and} \quad \text{whole}_1, \text{whole}_2, \ldots \quad \text{construct:}
\]

\[
\text{right}_1 \quad \text{whole}_1 \\
\text{right}_2 \quad \text{whole}_2 \\
\ldots \\
\ldots \\
\text{right}_1 \quad \text{whole}_i \\
\ldots \\
\text{right}_2 \quad \text{whole}_2 \\
\ldots \\
\text{left}_i \quad \text{whole}_i \\
\ldots \\
\text{left}_i \quad \text{whole}_i \\
\ldots \\
\text{and put} \quad \text{left}_i := f \quad \text{such that:}
\]

\[
f(\text{right}_i) = \text{whole}_i \\
\text{or} \quad f(\text{left}_i) = \text{whole}_i
\]

Application:
\[
\text{a unicorn coughed} \quad \checkmark
\]
\[
\text{a unicorn} = f \quad \text{coughed} \quad \checkmark
\]

where:
\[
f(\text{coughed}) = \text{a unicorn coughed},
\]
\[
f(\text{neighed}) = \text{a unicorn neighed}, \text{etc.}
\]

**Strategy 2:**
Define combination \( \Gamma \) by collecting all instances:

\[
\Gamma(\text{left}_i, \text{right}_i) = \text{whole}_i
\]

and find pattern.
Applications:
– quantified objects of transparent verbs:
  \[ \Gamma([\text{\textbf{read}}], [\text{a book}]) = [\text{read a book}] \]
  \[ \Gamma([\text{\textbf{read}}], [\text{every book}]) = [\text{read every book}] \]
  \[ \Gamma([\text{\textbf{buy}}], [\text{a book}]) = [\text{buy a book}] \]
  etc.
If \[ [X] \] is \(X\)'s extension, we have:
  \[ \Gamma((x,y) \mid x \text{ reads } y), \lambda P. \top P \cap B \neq \emptyset = \{ x \mid \{ y \mid x \text{ reads } y \} \cap B \neq \emptyset \} \]
  \[ \Gamma((x,y) \mid x \text{ reads } y), \lambda P. \top \subseteq P \subseteq \emptyset = \{ x \mid \{ y \mid x \text{ reads } y \} \} \]
  \[ \Gamma((x,y) \mid x \text{ buys } y), \lambda P. \top \cap B \subseteq \emptyset = \{ x \mid \{ y \mid x \text{ buys } y \} \cap B \subseteq \emptyset \} \]
  etc.
  – the pattern being:
    \[ \Gamma([\text{\textbf{LEFT}}], [\text{\textbf{RIGHT}}]) = [\text{\textbf{RIGHT}}](\{ x \mid \{ y \mid (x,y) \in \{ \text{\textbf{LEFT}} \} \}) \]
– quantified objects of opaque verbs:
    \[ [\text{seeks a unicorn}] \]
    \[ [\text{seeks}] = f \quad [\text{a unicorn}] \]
where:
(*) \[ f([\text{a unicorn}]) = [\text{seeks a unicorn}] \]
  \[ f([\text{a horse}]) = [\text{seeks a horse}] \]
If \[ [X] \] is \(X\)'s extension, then:
  \[ [\text{a unicorn}] = [\text{a ghost}] \]
and so:
  \[ f([\text{a unicorn}]) = f([\text{a ghost}]) \]
BUT: \[ f([\text{a unicorn}]) = [\text{seeks a unicorn}] = \{ x \mid x \text{ seeks a unicorn} \} \]
\[ \neq \quad \{ x \mid x \text{ seeks a ghost} \} = [\text{seeks a ghost}] = f([\text{a ghost}]) \]
\[ \Rightarrow \quad \text{NO EXTENSIONAL SOLUTION!} \]
If \[ [X] \] is \(X\)'s intension, then:
  \[ f([\text{a unicorn}]) = \{ x \mid x \text{ seeks a unicorn at [index] } i \} \]
  \[ f([\text{a horse}]) = \{ x \mid x \text{ seeks a horse at } i \} \]
  \[ f([\text{a ghost}]) = \{ x \mid x \text{ seeks a horse at } i \} \]
unclear how (and even: whether) value depends on argument
\[ \Rightarrow \quad \text{additional strategy needed:} \]

*Reduction by paraphrase*
… also works for (34b)

References
(on request)