Multidimensionality for use-conditional meaning*

Cross-dimensional quantification and the lexicon-syntax interface

Daniel Gutzmann

Talk @ University of Konstanz
May 23, 2013

1 Semantics vs. pragmatics

Semantics and pragmatics: 3 concepts

(1) *Truth conditions* pragmatics = meaning − truth conditions

(2) *Invariance* pragmatics = meaning − invariant meaning

(3) *Conventions* pragmatics = meaning − conventional meaning

Table 1: The "ideal" picture

Semantics	**Pragmatics**
conventional	conversational
invariant	context dependent
truth-conditional	non-truth-conditional

(4) *Indexicals* [+truth-conditional,−invariant, +conventional]

a. [adressing Paul] **You** are a linguist.
b. [adressing Pauline] **You** are a linguist.

(5) *Enrichment* [+truth-conditional, −invariant, −conventional]

a. I haven't eaten breakfast. [*today*]
b. You're not going to die. [*from this cut*]

(6) *Expressives* [−truth-conditional, −invariant, +conventional]

a. This **cur** howled the whole night.
b. This **damn** dog howled the whole night.

* This talk is based on parts of my 2012 University of Frankfurt dissertation titled *Use-conditional meaning. Studies in multidimensional semantics* (Gutzmann 2012).

Table 2: Conventionality and Truth-conditionality

	+truth-conditional	−truth-conditional
+conventional	descriptive meaning	use-conditional meaning
−conventional	pragmatic enrichment	conversational implicatures

2 Hybrid Semantics

For certain expressions of natural language, a correct Semantic Theory would state **rules of use** rather than something like a concept expressed. (Kaplan 1999: 6)

Use this perspective to *supplement* truth-conditional semantics, not to replace it.

(7) *Truth* and *use conditions*

a. "The damn dog howled" is **true** if the dog howled.

b. "The damn dog howled" is **felicitously used** if the speaker feels negatively about the dog.

Expressions with both meaning dimensions are *hybrid* expressions.

(8) *Hybrid semantics:* Two independent meaning dimensions: truth and felicity.

$\langle 1, \checkmark \rangle \quad \langle 1, \lightning \rangle \quad \langle 0, \checkmark \rangle \quad \langle 0, \lightning \rangle$

Denotations for use-conditional meaning

Truth-conditions	Use-conditions
(T) $_1$ "Snow is white" $_2$ is **true**, $_3$ iff snow is white.	(U) $_1$ "Oops!" $_2$ is **felicitously used**, $_3$ iff the speaker observed a minor mishap.

In both conditions, an expression is connected with a condition that captures its meaning. What differs is the kind of connection ("mode of expression"). These conditions can be the case or not. ➞ Standard tools available!

(9) *Tc-content: set of worlds*

a. $\|\textit{Snow is white}\|^t = \{w\text{: snow is white in } w\}$

b. $\|\textit{Snow is white}\|^t = 1$, if $w_@ \in \{w\text{: snow is white in } w\}$

(10) *Uc-content: set of contexts*

a. $\|\textit{Oops}\|^u = \{c\text{: } c_S \text{ observed a minor mishap in } c_w\}$

b. $\|\textit{Oops}\|^u = \checkmark$, if $c_@ \in \{c\text{: } c_S \text{ observed a minor mishap in } c_w\}$

Goal for this talk: Presenting a compositional, multidimensional formal framework for hybrid semantics that can deal with all kinds of use-conditional items.

3 Varieties of use-conditional meaning

Use-conditional items (UCIs) are attested at all levels of language.

(11) *Intonation*

a. Wie **KLUG** Piet ist! (How smart Piet is!) (exclamative intonation)

b. Peter **HAT** geschlafen. (Peter DID sleep) (verum focus)

(12) *Morphology*

a. Hans-**i** (diminuitive)

b. ⋆**winkwink**⋆ (wave.*stem*.wave.*stem*) (NIC)

(13) *Lexical items*

a. **Ouch**! (interjection)

b. Lessing was a **Kraut**. (ethnic slur)

c. Es regnet **ja**. ('It's raining MP') (modal particle)

d. Heino **begafft** Tina. ('Heino is gawking at Tina') (colored verb)

(14) *Syntax* Sophie, Piet likes. (Topicalization)

Depending on how UCIs interact with other content, we can distinguish 5(-6) types of UCIs.
For illustration, I employ an informal tower notation: $\frac{\text{uc content}}{\text{tc content}}$

(15) *functional expletive UCIs*

$$\text{That damn Daniel is talking} = \frac{\text{the speaker dislikes Daniel}}{\text{Daniel is talking}}$$

(16) *Schema for functional expletive UCIs*

$$S[\ldots\varepsilon(\alpha)\ldots] = \frac{\varepsilon(\alpha)}{S[\ldots\alpha\ldots]}$$

(17) *isolated expletive UCIs*

$$\textbf{Ouch}! = \frac{\text{The speaker fells pain}}{\varnothing}$$

(18) *Schema for isolated expletive UCIs*

$$S[\ldots\varepsilon\ldots] = \frac{\varepsilon}{S}$$

(19) *functional shunting UCIs*

$$\text{NIC(wave)} = \frac{\text{The speaker performs an act of waving}}{\varnothing}$$

(20) *Schema for functional shunting UCIs*

$$S[\ldots\varepsilon(\alpha)\ldots] = \frac{\varepsilon(\alpha)}{S}$$

(21) *isolated mixed UCIs*

$$\text{Lessing is a Kraut} = \frac{\text{the speaker dislikes Germans}}{\text{Lessing is German}}$$

(22) *Schema for isolated mixed UCIs*

$$S[\ldots\varepsilon(\alpha)\ldots] = \frac{\varepsilon_u}{S[\ldots\varepsilon_t(\alpha)\ldots]}$$

(23) *functional mixed UCIs*

$$\text{Heino begafft Tina} = \frac{\text{The speaker dislikes how Heino looks at Tina}}{\text{Heino looks at Tina}}$$

(24) *Schema for unctional mixed UCIs*

$$S[\ldots\varepsilon(\alpha)\ldots] = \frac{\varepsilon_u(\alpha)}{S[\ldots\varepsilon_t(\alpha)\ldots]}$$

(25) *(Use-conditional modifiers)*

$$\text{that fucking(bastard) Daniel} = \frac{\text{the speaker really dislikes Daniel}}{\text{Daniel}}$$

These types of UCIs can be described by 3 binary features:

1. ±***functional***
   Does a UCI need an argument in order to express use-conditions?

2. ±***2-dimensional***
   Does a UCI also contribute tc-content?

3. ±***resource-sensitive***
   Does a UCI pass back its argument? (only for functional, non-mixed UCI)

Table 3: Types of use-conditional items

	f	*2d*	*rs*
isolated expletive UCIs	−	−	
isolated mixed UCIs	−	+	
functional UCIs, expletive	+	−	−
shunting	+	−	+
functional mixed UCIs	+	+	

All 6 types of UCIs should be accounted for by the framework. As we will see, the previous approaches (Potts 2005, McCready 2010, Gutzmann 2011) are not able to do this.

4 Previous approaches: $\mathcal{L}^*_{\mathsf{CI}}$

Potts's *Logic of conventional implicatures* (2005, 2007) has been highly influential. It is, however, very restrictive and cannot account for all types of UCIs. Therefore, it has been extended to account for more data (McCready 2010, Gutzmann 2011) . The resulting system is however very complex and needs many specific semantic types and combinatoric rules. It also has problems with "cross-dimensional quantification".

4.1 Potts's $\mathcal{L}_{\text{CI}}$

The basic idea of $\mathcal{L}_{\text{CI}}$ (and its successors; together referred to as $\mathcal{L}^*_{\text{CI}}$) is to have special semantic types for expressive/uc-content. These semantic types are associated with dedicated composition rules that separate the uc- from the tc-content.

(26) *Types in $\mathcal{L}_{CI}$*[1]

a. e, s, t are tc-types.
b. u is a uc-type.
c. If σ and τ are tc-types, then $\langle \sigma, \tau \rangle$ is a tc-type.
d. If σ is a tc-type and σ is a uc-type, then $\langle \sigma, \tau \rangle$ is a uc-type.

Crucially:

(i) There are no types that have a uc-type argument. (→ no uc-modifiers)
(ii) Every type is either a tc- or uc-type. (→ no mixed UCIs)

(27) *tc-application*

$$\frac{\alpha : \langle \sigma, \tau \rangle \qquad \beta : \sigma}{\alpha(\beta) : \tau}$$

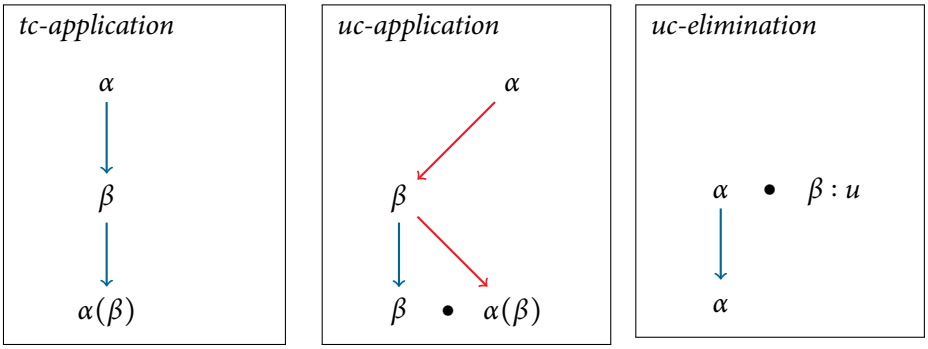
(28) *uc-application*

$$\frac{\alpha : \langle \sigma, \tau^u \rangle \qquad \beta : \sigma}{\beta : \sigma \bullet \alpha(\beta) : \tau^u}$$

(29) *uc-elimination*

$$\frac{\alpha : \sigma \bullet \beta : u}{\alpha : \sigma}$$

For better illustration, I will use an arrow-diagram notation.



(30)

$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{That damn}}{\textbf{damn} : \langle e, u \rangle}\,\text{Lx} \qquad \dfrac{\textit{Daniel}}{\textbf{daniel} : e}\,\text{Lx}}{\textbf{daniel} : e \bullet \textbf{damn}(\textbf{daniel}) : u}\,\text{uca}}{\textbf{daniel} : e}\,\text{uce} \qquad \dfrac{\textit{is talking}}{\textbf{talk} : \langle e, t \rangle}\,\text{Lx}}{\textbf{talk}(\textbf{daniel})}\,\text{tca}$$

Parsetree interpretation collects the stranded uc-proposition and interprets it together with the tc-content (=the root node).

(31) *Parsetree interpretation*

$$[\![(30)]\!] = \langle [\![\textbf{talk}(\textbf{daniel})]\!], \{[\![\textbf{damn}(\textbf{daniel})]\!]\} \rangle$$

1 This is a slight reformulation to bring it more in line with the resulting system

Since $\mathcal{L}_{\text{CI}}$ neither has types (and combination rules) for mixed UCIs nor for shunting UCIs, it can only deal with (isolated and functional) expletive UCIs. The same holds for use-conditional modifiers.

Table 4: Types of UCIs in $\mathcal{L}_{\text{CI}}$

isolated expletive UCIs	✓
isolated mixed UCIs	✗
functional UCIs, expletive	✓
shunting	✗
uc-modifiers	✗
functional mixed UCIs	✗

There are additional other constructions that are problematic for $\mathcal{L}_{\text{CI}}$. We will however first consult the extensions of $\mathcal{L}_{\text{CI}}$.

4.2 McCready's $\mathcal{L}^{+}_{\text{CI}}$ and Gutzmann's $\mathcal{L}^{+\text{EM}}_{\text{CI}}$

McCready's (2010) extension (called $\mathcal{L}^{+}_{\text{CI}}$), add new types and composition rules for shunting and mixed UCIs.

(32) *Shunting types in $\mathcal{L}^{+}_{CI}$*

a. u^s is a basic shunting type.

b. If σ is a tc- or shunting type and τ is a shunting type, then $\langle\sigma,\tau\rangle$ is a shunting type.

(33) *Mixed types in $\mathcal{L}^{+}_{CI}$*

If σ is a tc- type and τ is a shunting type, then $\sigma \times \tau$ is a mixed type.

(34) *Mixed expressions in $\mathcal{L}^{+}_{CI}$*

If α is a tc-expressions and β is a shunting expression, then $\alpha \blacklozenge \beta$ is a mixed expression.

We also need new composition rules for these new types.

(35) *shunting application*

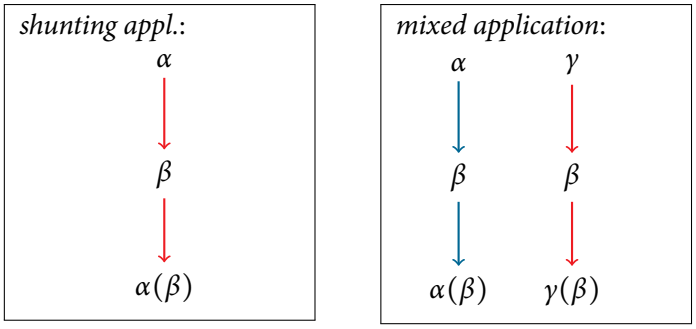
$$\frac{\alpha : \langle\sigma,\tau\rangle^s \qquad \beta : \sigma}{\alpha(\beta) : \sigma}$$

(36) *mixed application*

$$\frac{\alpha \blacklozenge \gamma : \langle\sigma,\tau\rangle \times \langle\sigma,\rho\rangle \qquad \beta : \sigma}{\alpha(\beta) \blacklozenge \gamma(\beta) : \tau \times \rho}$$

(37) *shunting elimination*

$$\frac{\alpha \blacklozenge \beta : \sigma \times u^s}{\alpha : \sigma \bullet \beta : u^c}$$



(38) Heino beglotzt Tina ("Heino gawks at Tina")

$$\dfrac{\dfrac{\mathbf{heino} : e \qquad \dfrac{\lambda x \lambda y.\mathbf{look\text{-}at}(x)(y) \blacklozenge \lambda x \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(x)(y)) : \langle e, \langle e, t\rangle\rangle \times \langle e, \langle e, u^s\rangle\rangle \qquad \mathbf{tina} : e}{\lambda y.\mathbf{look\text{-}at}(\mathbf{tina})(y) \blacklozenge \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(y)) : \langle e, t\rangle \times \langle e, u^s\rangle}}{\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) \blacklozenge \mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heine})) : t \times u^s}}{\dfrac{\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) : t \bullet \mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heine})) : u}{\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) : t}}$$

(39) *wink* ("*wave*")

$$\dfrac{\mathbf{wave} : \langle e, t\rangle \qquad \mathbf{NIC} : \langle \langle e, t\rangle, u^s\rangle}{\mathbf{NIC}(\mathbf{wave}) : u^s}$$

$\mathcal{L}^{+}_{\mathrm{CI}}$ needs a second rule of parsetree interpretation for cases in which there is no tc-content. It employs a trivial tc-item T.

(40) *Parsetree interpretation "without" tc-content*

$$[\![(39)]\!] = \langle [\![T]\!], [\![\mathbf{wave})]\!]\rangle$$

Even $\mathcal{L}^{+}_{\mathrm{CI}}$ can still not deal with use-conditional modification. Therefore, Gutzmann (2011) adds another set of types and composition rules.

(41) *Pure uc-types*
If σ and τ are pure or ordinary uc-types, then $\langle \sigma, \tau\rangle$ is a pure uc-type.

(42) *Uc-modification*

$$\dfrac{\alpha : \langle \sigma^u, \tau^u\rangle \qquad \beta : \sigma^u}{\alpha(\beta) : \tau^u}$$

Taken together, $\mathcal{L}^{*}_{\mathrm{CI}}$ then has 4 different types and (at least) 7 composition rules. However, it still has problem with some quantificational constructions.

4.3 Cross-dimensional quantification

(43) Alle beglotzen Tina. (Quantification with mixed UCIs)

$$\dfrac{\mathbf{everybody} : \langle\langle e, t\rangle, t\rangle \qquad \dfrac{\lambda x \lambda y.\mathbf{look\text{-}at}(x)(y) \blacklozenge \lambda x \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(x)(y)) : \langle e, \langle e, t\rangle\rangle \times \langle e, \langle e, u^s\rangle\rangle \qquad \mathbf{tina} : e}{\lambda y.\mathbf{look\text{-}at}(\mathbf{tina})(y) \blacklozenge \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(y)) : \langle e, t\rangle \times \langle e, u^s\rangle}}{\lightning}$$

(44) Heino beglotzt jedes Mädchen. (QR with mixed UCIs)
[jedes Mädchen$_1$ [1 [Heino [beglotzt t_1]]]]

$$\dfrac{\mathbf{every}(girl) : \langle\langle e, t\rangle, t\rangle \qquad \dfrac{\lambda z \qquad \dfrac{\dfrac{\dfrac{\mathbf{heino} : e \qquad \dfrac{\lambda x \lambda y.\mathbf{look\text{-}at}(x)(y) \blacklozenge \lambda x \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(x)(y)) : \langle e, \langle e, t\rangle\rangle \times \langle e, \langle e, u^s\rangle\rangle \quad z : e}{\lambda y.\mathbf{look\text{-}at}(z)(y) \blacklozenge \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(z)(y)) : \langle e, t\rangle \times \langle e, u^s\rangle}}{\mathbf{look\text{-}at}(z)(\mathbf{heino}) \blacklozenge \mathbf{bad}(\mathbf{look\text{-}at}(z)(\mathbf{heino})) : t \times u^s}}{\mathbf{look\text{-}at}(z)(\mathbf{heino}) : t \bullet \mathbf{bad}(\mathbf{look\text{-}at}(z)(\mathbf{heino})) : u^s}}{\mathbf{look\text{-}at}(z)(\mathbf{heino}) : t}}{\lambda z \mathbf{look\text{-}at}(z)(\mathbf{heino}) : \langle e, t\rangle}}{\mathbf{every}(girl)(\lambda z \mathbf{look\text{-}at}(z)(\mathbf{heino})) : t}$$

The variable z remains unbound in the uc-dimension, therefore leading to wrong use-conditions.

5 $\mathcal{L}_{\mathrm{TU}}$ – a framework for hybrid semantics

The basic idea "True multidimensionality": Each natural language expression corresponds to a 3-dimensional logical expression in a semantic parsetree.

(45) *3-dimensional expressions*
$A \rightsquigarrow \alpha_1 \blacklozenge \alpha_2 \bullet \alpha_3$

t-dimension:	tc-content
s-dimension:	content relevant for the calculation of uc-content
u-dimension:	store for saturated uc-content

Each dimension is represented by an expression of the logic $\mathcal{L}_{\mathrm{TU}}$. The distinction between tc- and uc-content is again built on a semantic type distinction.

(46) *Types for $\mathcal{L}_{TU}$*

a. e, t are basic truth-conditional types for $\mathcal{L}_{\mathrm{TU}}$.
b. u is a basic use-conditional type for $\mathcal{L}_{\mathrm{TU}}$.
c. If τ is a truth-conditional type for $\mathcal{L}_{\mathrm{TU}}$,
then $\langle s, \tau\rangle$ is a truth-conditional type for $\mathcal{L}_{\mathrm{TU}}$.
d. If σ and τ are truth-conditional types for $\mathcal{L}_{\mathrm{TU}}$,
then $\langle \sigma, \tau\rangle$ is a truth-conditional type for $\mathcal{L}_{\mathrm{TU}}$.
e. If σ is a type for $\mathcal{L}_{\mathrm{TU}}$ and τ is a use-conditional type for $\mathcal{L}_{\mathrm{TU}}$,
then $\langle \sigma, \tau\rangle$ is a use-conditional type for $\mathcal{L}_{\mathrm{TU}}$.
f. The set of all types for $\mathcal{L}_{\mathrm{TU}}$ is the union of all truth-conditional and use-conditional types.

Expressions of type u denote set of contexts ("use-conditional propositions").

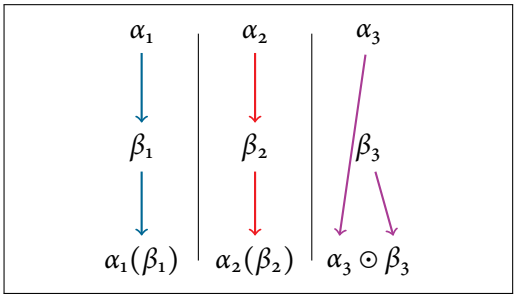
These 3-dimensional expressions are inserted into the compositional system, where they are combined according to two composition rules.

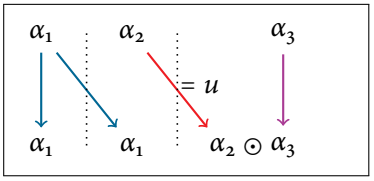
(47) *multidimensional application*

$$\frac{\alpha_1 \blacklozenge \alpha_2 \blacklozenge \alpha_3 \qquad \beta_1 \blacklozenge \beta_2 \blacklozenge \beta_3}{\alpha_1(\beta_1) \blacklozenge \alpha_2(\beta_2) \bullet \alpha_3 \odot \beta_3}$$

(48) *uc-elimination*

$$\frac{\alpha_1 \blacklozenge \alpha_2 : u \bullet \alpha_3}{\alpha_1 \blacklozenge \alpha_1 \bullet \alpha_3 \odot \alpha_2}$$





Even if every expression needs to be 3-dimensional for the compositional system to work, the lexical entries do not need to be. It is sufficient to know one or two dimensions, the rest can be derived on a regular basis. For this, I make use of so-called *lexical extension rules*, (LERs) that expand the 1- or 2-dimensional lexical entries into 3-dimensional expressions that can be used by the compositional system.

(49) *Lexical extension of functional expletive UCIs*

bastard : $\langle e, u\rangle \Rightarrow I_e \blacklozenge$ **bastard** : $\langle e, u\rangle \bullet U$

This does not only help to keep the lexicon simple, but also allows for the implementation of combinatoric restrictions. Since the LERs are assumed to be part of the lexicon(-syntax interface), they may very cross-linguistically (which may be desirable).

In order for dealing with cross-dimensional quantification, we need the following rule to "hybridize" quantifiers.

(50) *Hybrid quantifiers*

$$\frac{Q : \langle\langle e, t\rangle, t\rangle}{\star Q : \langle\langle e, u\rangle, u\rangle}$$

where $[\![\star Q]\!]$ = the function $f \in D_{\langle\langle e,u\rangle,u\rangle}$ such that for every $E \in D_{\langle e,u\rangle}$, $f(E) = \{c \in C : \{x \in D_e : E(x)(c) = 1\} \in [\![Q]\!]^{\mathcal{M},g}\}$.

Abstraction crucially just works on the first two dimensions and only over tc-material.

(51) *Multidimensional abstraction*

$$\frac{\langle \alpha : \tau, \beta : \rho, \gamma\rangle}{\langle \lambda x.\alpha : \langle \sigma, \tau\rangle, \lambda x.\beta : \langle \sigma, \rho\rangle, \gamma\rangle}\ \lambda x_\sigma$$

where σ is a truth-conditional type

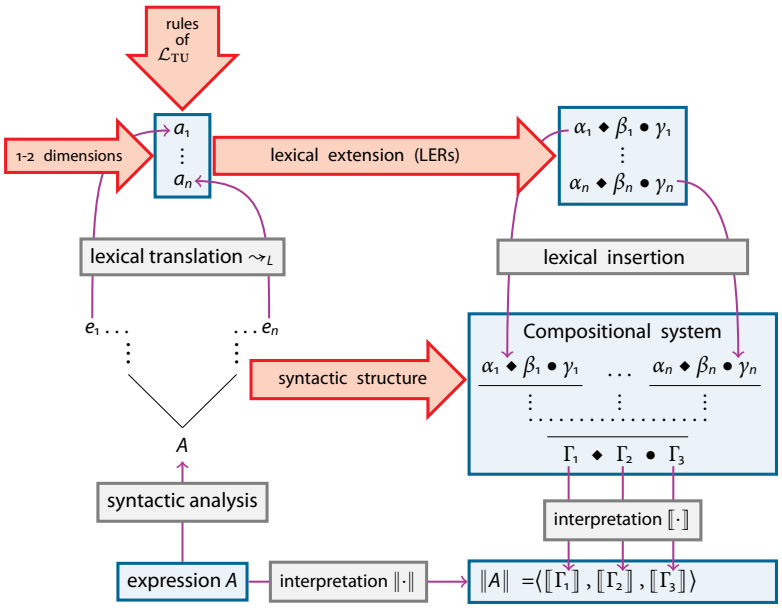


Figure 1 Architecture of the framework

5.1 Some Sample derivations

(52) *Functional expletive* UCIs — That damn Daniel is talking.

$$\dfrac{\dfrac{\dfrac{\mathbf{damn} : \langle e, u\rangle}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{damn} : \langle e, u\rangle\\ I_e\end{array}}\,\text{LER} \qquad \dfrac{\mathbf{daniel}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{daniel} : e\\ \mathbf{daniel} : e\end{array}}\,\text{LER}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{damn}(\mathbf{daniel}) : u\\ \mathbf{daniel} : e\end{array}}\,\text{MdA}}{\begin{array}{c}\bullet\, \mathbf{damn}(\mathbf{daniel}) : u\\ \blacklozenge\, \mathbf{daniel} : e\\ \mathbf{daniel} : e\end{array}}\,\text{UCE} \qquad \dfrac{\mathbf{talk}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{talk} : \langle e, t\rangle\\ \mathbf{talk} : \langle e, t\rangle\end{array}}\,\text{LER}}{\begin{array}{c}\bullet\, \mathbf{damn}(\mathbf{daniel}) : u\\ \blacklozenge\, \mathbf{talk}(\mathbf{daniel}) : t\\ \mathbf{talk}(\mathbf{daniel}) : t\end{array}}\,\text{MdA}$$

(53) *Isolated expletive UCIs* — Ouch!

$$\dfrac{\mathbf{ouch} : u}{\top \blacklozenge \top \bullet \mathbf{ouch} : u}\,\text{LER}$$

(54) *Isolated mixed UCIs* Lessing is a Kraut.

$$\dfrac{\dfrac{\begin{array}{c}\mathbf{lessing} : e\\ \hline \bullet\, U\\ \blacklozenge\, \mathbf{lessing} : e\\ \mathbf{lessing} : e\end{array}}{}\,\text{LER} \qquad \dfrac{\langle \mathbf{german} : \langle e, t\rangle, \mathbf{bad}({}^{\cap}\mathbf{german}) : u\rangle}{\begin{array}{c}\bullet\, \mathbf{bad}({}^{\cap}\mathbf{german}) : u\\ \blacklozenge\, \mathbf{german} : \langle e, t\rangle\\ \mathbf{german} : \langle e, t\rangle\end{array}}\,\text{LER}}{\mathbf{german}(\mathbf{lessing}) : t \blacklozenge \mathbf{german}(\mathbf{lessing}) : t \bullet \mathbf{bad}({}^{\cap}\mathbf{german}) : u}\,\text{MdA}$$

(55) *Functional shunting UCIs* *winkwink*

$$\dfrac{\dfrac{\dfrac{\lambda P_{\langle e,t\rangle}.T \blacklozenge NIC : \langle\langle e, t\rangle, u\rangle}{\lambda P_{\langle e,t\rangle}.\top \blacklozenge NIC : \langle\langle e, t\rangle, u\rangle \bullet U}\,\text{LER} \qquad \dfrac{\mathbf{wave} : \langle e, t\rangle}{\mathbf{wave} : \langle e, t\rangle \blacklozenge \mathbf{wave} : \langle e, t\rangle \bullet U}}{\top \blacklozenge NIC(wave) : u \bullet U}\,\text{MdA}}{\top \blacklozenge \top \bullet NIC(wave) : u}\,\text{UCE}$$

(56) *Functional mixed expressives* Heino begafft Tina.

$$\dfrac{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{heino} : e\\ \mathbf{heino} : e\end{array} \qquad \dfrac{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \lambda x \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(x)(y)) : \langle e, \langle e, u\rangle\rangle\\ \mathbf{look\text{-}at} : \langle e, \langle e, t\rangle\rangle\end{array} \quad \begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{tina} : e\\ \mathbf{tina} : e\end{array}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(y)) : \langle e, u\rangle\\ \mathbf{look\text{-}at}(\mathbf{tina}) : \langle e, t\rangle\end{array}}\,\text{MdA}}{\dfrac{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino})) : u\\ \mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) : t\end{array}}{\begin{array}{c}\bullet\, \mathbf{bad}(\mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino})) : u\\ \blacklozenge\, \mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) : t\\ \mathbf{look\text{-}at}(\mathbf{tina})(\mathbf{heino}) : t\end{array}}\,\text{UCE}}\,\text{MdA}$$

(57) *UC-modifiers* That fucking bastard Daniel

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathbf{fucking} : \langle\langle e, u\rangle, \langle e, u\rangle\rangle\\ \hline I_{\langle e,e\rangle} \blacklozenge \mathbf{fucking} : \langle\langle e, u\rangle, \langle e, u\rangle\rangle \bullet U\end{array}}{}\,\text{LER} \qquad \dfrac{\begin{array}{c}\mathbf{bastard} : \langle e, u\rangle\\ \hline I_e \blacklozenge \mathbf{bastard} : \langle e, u\rangle \bullet U\end{array}}{}\,\text{LER}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{fucking}(\mathbf{bastard}) : \langle e, u\rangle\\ I_e\end{array}}\,\text{MdA} \qquad \begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{daniel} : e\\ \mathbf{daniel} : e\end{array}}{\dfrac{\begin{array}{c}\bullet\, U\\ \blacklozenge\, \mathbf{fucking}(\mathbf{bastard})(\mathbf{daniel}) : u\\ \mathbf{daniel} : e\end{array}}{\mathbf{daniel} : e \blacklozenge \mathbf{daniel} : e \bullet \mathbf{fucking}(\mathbf{bastard})(\mathbf{daniel}) : u}\,\text{UCE}}\,\text{MdA}$$

(58) *Cross-dimensional quantification*

$$
\begin{array}{c}
\dfrac{\dfrac{\dfrac{\textit{Jeder}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\ \star\,\mathbf{everyone} : \langle\langle e,u\rangle,u\rangle\\ \mathbf{everyone} : \langle\langle e,t\rangle,t\rangle\end{array}}\ \text{LxL} \qquad \dfrac{\dfrac{\textit{begafft}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\ \lambda x\lambda y.\mathbf{bad}(\mathbf{look\text{-}at}_i(x)(y)) : \langle e,\langle e,u\rangle\rangle\\ \lambda x\lambda y.\mathbf{look\text{-}at}_i(x)(y) : \langle e,\langle e,t\rangle\rangle\end{array}}\ \text{LxL} \quad \dfrac{\textit{Natalie}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\ \mathbf{n} : e\\ \mathbf{n} : e\end{array}}\ \text{LxL}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\ \lambda y.\mathbf{bad}(\mathbf{look\text{-}at}_i(\mathbf{n})(y)) : \langle e,u\rangle\\ \lambda y.\mathbf{look\text{-}at}_i(\mathbf{n})(y) : \langle e,t\rangle\end{array}}\ \text{Md}}}{\begin{array}{c}\bullet\, U\\ \blacklozenge\ \star\,\mathbf{everyone}(\lambda y.\mathbf{bad}(\mathbf{look\text{-}at}_i(\mathbf{n})(y))) : u\\ \mathbf{everyone}(\lambda y.\mathbf{look\text{-}at}_i(\mathbf{n})(y)) : t\end{array}}\ \text{Md}}{\begin{array}{c}\bullet\ \star\,\mathbf{everyone}(\lambda y.\mathbf{bad}(\mathbf{look\text{-}at}_i(\mathbf{n})(y))) : u\\ \blacklozenge\,\mathbf{everyone}(\lambda y.\mathbf{look\text{-}at}_i(\mathbf{n})(y)) : t\\ \mathbf{everyone}(\lambda y.\mathbf{look\text{-}at}_i(\mathbf{n})(y)) : t\end{array}}\ \text{Ue}
\end{array}
$$

$$\boxed{\|\text{Thank you!}\|^u = \{c : c_S \text{ thanks } c_A \text{ in } c_W\}}$$

References

Gutzmann, Daniel (2011): "Expressive modifiers & mixed expressives". In: Olivier Bonami & Patricia Cabredo-Hofherr, eds.: *Empirical Issues in Syntax and Semantics 8*, 123–141. URL: http://www.cssp.cnrs.fr/eiss8/gutzmann-eiss8.pdf.

Gutzmann, Daniel (2012): "Use-conditional meaning. Studies in multidimensional semantics". PhD thesis. Frankfurt: Goethe-Universität.

Kaplan, David (1999): "The meaning of *ouch* and *oops*. Explorations in the theory of meaning as use". 2004 version. Ms. Los Angeles.

McCready, Eric (2010): "Varieties of conventional implicature. Evidence from Japanese". *Semantics & Pragmatics* 3.8, 1–57. DOI: 10.3765/sp.3.8.

Potts, Christopher (2005): *The Logic of Conventional Implicature*. Oxford Studies in Theoretical Linguistics 7. Oxford: Oxford University Press.

Potts, Christopher (2007): "The expressive dimension". *Theoretical Linguistics* 33.2, 165–197. DOI: 10.1515/TL.2007.011. URL: http://people.umass.edu/potts/papers/potts-expressives06.pdf.