Expressive Modifiers & Mixed Expressives

Daniel Gutzmann

University of Frankfurt

CSSP 2009 – September 24, 2009

Introduction

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\mathrm{CI}}$ for dealing with *conventional implicatures* (CIs).

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\mathrm{CI}}$ for dealing with *conventional implicatures* (CIs).
- He divides CIs into *supplements* and *expressives*.

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\mathrm{CI}}$ for dealing with *conventional implicatures* (CIs).
- He divides CIs into *supplements* and *expressives*.
- In his later work (Potts 2007*b*,*c*), expressives receive a different interpretation, but the type system and the combinatorics remain the same.

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\text{CI}}$ for dealing with *conventional implicatures* (CIs).
- He divides CIs into *supplements* and *expressives*.
- In his later work (Potts 2007*b*,*c*), expressives receive a different interpretation, but the type system and the combinatorics remain the same.
- Although $\mathcal{L}_{\text{CI}}$ is a great tool for studying and analysing non-at-isue kinds of meaning, I show that it has still some problems.

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\text{CI}}$ for dealing with *conventional implicatures* (CIs).
- He divides CIs into *supplements* and *expressives*.
- In his later work (Potts 2007*b*,*c*), expressives receive a different interpretation, but the type system and the combinatorics remain the same.
- Although $\mathcal{L}_{\text{CI}}$ is a great tool for studying and analysing non-at-isue kinds of meaning, I show that it has still some problems.
- These problems are raised by *expressive modifiers* and *mixed expressives*.

- Potts (2005) develops a multidimensional logic $\mathcal{L}_{\mathrm{CI}}$ for dealing with *conventional implicatures* (CIs).
- He divides CIs into *supplements* and *expressives*.
- In his later work (Potts 2007*b*,*c*), expressives receive a different interpretation, but the type system and the combinatorics remain the same.
- Although $\mathcal{L}_{\mathrm{CI}}$ is a great tool for studying and analysing non-at-isue kinds of meaning, I show that it has still some problems.
- These problems are raised by *expressive modifiers* and *mixed expressives*.
- To overcome these problems, the type of $\mathcal{L}_{\mathrm{CI}}$ system must be extended.

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Claim (1)

(1) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Claim (1)

(1) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)

a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Claim (1)

(1) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)

a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)

b. Expressive content never applies to expressive content. (Potts 2005: § 3.5.2)

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Claim (1)

(1) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)

a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)

b. Expressive content never applies to expressive content. (Potts 2005: § 3.5.2)

Claim (2)

(2) No lexical item contributes both an at-issue and a CI-meaning. (Potts 2005: 7)

Two claims

Potts (2005, 2007*c*) makes the following two important claims:

Claim (1)

(1) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)

a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)

b. Expressive content never applies to expressive content. (Potts 2005: § 3.5.2)

Claim (2)

(2) No lexical item contributes both an at-issue and a CI-meaning. (Potts 2005: 7)

⇒ These restrictions are directly built into $\mathcal{L}_{\text{CI}}$.

- The claim (1) that expressives are only output types is challenged by expressive modifiers.

- The claim (1) that expressives are only output types is challenged by expressive modifiers.
- The claim (2) that no expression contributes to both dimensions of meaning is challenged by mixed expressives.

- The claim (1) that expressives are only output types is challenged by expressive modifiers.
- The claim (2) that no expression contributes to both dimensions of meaning is challenged by mixed expressives.
- If these challenges are valid, the system of $\mathcal{L}_{\mathrm{CI}}$ should be changed to accomodate such cases.

Outline

Potts' Logic of Conventional Implicature

The core of $\mathcal{L}_{\mathrm{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3) a. *e* and *t* are descriptive types.

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3) a. *e* and *t* are descriptive types.
b. ε is an expressive type.

The core of $\mathcal{L}_{\mathrm{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3) a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type, then $\langle \sigma, \varepsilon \rangle$ is an expressive type.

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle\sigma, \tau\rangle$ is a descriptive type.
d. If σ is a descriptive type, then $\langle\sigma, \varepsilon\rangle$ is an expressive type.
e. The set of types is the union of the descriptive and expressive types.

The core of $\mathcal{L}_{\mathrm{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type, then $\langle \sigma, \varepsilon \rangle$ is an expressive type.
e. The set of types is the union of the descriptive and expressive types.

Gaps in the type system

The core of $\mathcal{L}_{\text{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3) a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type, then $\langle \sigma, \varepsilon \rangle$ is an expressive type.
e. The set of types is the union of the descriptive and expressive types.

Gaps in the type system

(4) a. If σ is a descriptive type, then $\langle \varepsilon, \sigma \rangle$ …

The core of $\mathcal{L}_{\mathrm{CI}}$ is its type system where the restrictions for expressives expressions are formulated.

Types for LCI

(3)
a. e and t are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type, then $\langle \sigma, \varepsilon \rangle$ is an expressive type.
e. The set of types is the union of the descriptive and expressive types.

Gaps in the type system

(4)
a. If σ is a descriptive type, then $\langle \varepsilon, \sigma \rangle$ …
b. If σ and τ are expressive types, then $\langle \sigma, \tau \rangle$ …

Special tree-admissibilty conditions regulate how expressive and descriptive expression combine with each other.

Special tree-admissibilty conditions regulate how expressive and descriptive expression combine with each other.

CI-application

(5)

$$\beta : \sigma$$

$$\bullet$$

$$\alpha(\beta) : \varepsilon$$

$$\alpha : \langle \sigma, \varepsilon \rangle \qquad \beta : \sigma$$

Special tree-admissibilty conditions regulate how expressive and descriptive expression combine with each other.

CI-application

(5)

$$\beta : \sigma$$
$$\bullet$$
$$\alpha(\beta) : \varepsilon$$
$$\alpha : \langle \sigma, \varepsilon \rangle \quad \beta : \sigma$$

(6) That bastard burns is a zombie.

$$\textbf{zombie}(\textbf{burns}) : t$$
$$\textbf{burns} : e \qquad \textbf{zombie} : \langle e, t \rangle$$
$$\bullet$$
$$\textbf{bastard}(\textbf{burns}) : \varepsilon$$
$$\textbf{bastard} : \langle e, \varepsilon \rangle \quad \textbf{burns} : e$$

Problem 1: Expressive modifiers

Gaps in the type system

(7) a. If σ is a descriptive type, then $\langle \varepsilon, \sigma \rangle$ …
b. If σ and τ are expressive types, then $\langle \sigma, \tau \rangle$ …

Gaps in the type system

(7) a. If σ is a descriptive type, then $\langle \varepsilon, \sigma \rangle$ …
b. If σ and τ are expressive types, then $\langle \sigma, \tau \rangle$ …

Claim (1)

(8) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)
a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)
b. Expressive content never applies to expressive content. (Potts 2005: § 3.5.2)

Gaps in the type system

(7) a. If σ is a descriptive type, then $\langle \varepsilon, \sigma \rangle$ ...
    b. If σ and τ are expressive types, then $\langle \sigma, \tau \rangle$ ...

Claim (1)

(8) Expressive types are only output types, i.e.: (Potts 2007*c*: 169)
    a. At-issue content never applies to expressive content. (Potts 2005: § 3.5.1)
    b. Expressive content never applies to expressive content. (Potts 2005: § 3.5.2)

⇒ No expressive modifiers!

The type system of $\mathcal{L}_{\mathrm{CI}}$ predicts that there are no expressive modifiers, i.e expression mapping expressive content to expressive content.

Expressive modifiers

Expressive modifiers

(9) a. That fucking bastard Burns got promoted!

Expressive modifiers

(9) a. That fucking bastard Burns got promoted!

b. Holy shit, my bike tire is flat again!

Expressive modifiers

(9) a. That fucking bastard Burns got promoted!
b. Holy shit, my bike tire is flat again!
c. I feel really fucking brilliant.

Expressive modifiers

(9) a. That fucking bastard Burns got promoted!
b. Holy shit, my bike tire is flat again!
c. I feel really fucking brilliant.

Intuitive structure (cf. e.g. Geurts 2007)

(10) That (fucking(bastard))(Burns) got promoted!

Expressive modifiers

(9) a. That fucking bastard Burns got promoted!
    b. Holy shit, my bike tire is flat again!
    c. I feel really fucking brilliant.

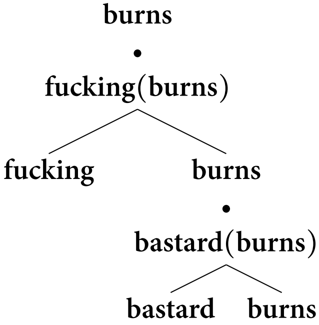
Intuitive structure (cf. e.g. Geurts 2007)

(10) That (fucking(bastard))(Burns) got promoted!

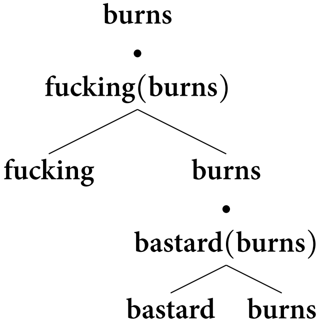
Structure assigned by $\mathcal{L}_{\mathrm{CI}}$

(11) That fucking(Burns)•bastard(Burns) got promoted!

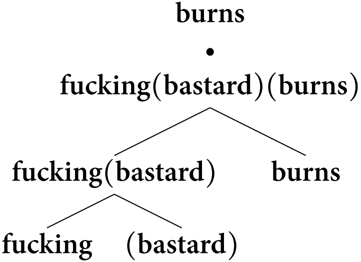
(12) Structure assigned by $\mathcal{L}_{\text{CI}}$



(12) Structure assigned by $\mathcal{L}_{\text{CI}}$



(13) Intuitive structure



- Potts (2007*a*) presents a revised approach that ensures that *that fucking bastard Burns* is expressively stronger than *that bastard Burns* while keeping the structure in (12).

- Potts (2007*a*) presents a revised approach that ensures that *that fucking bastard Burns* is expressively stronger than *that bastard Burns* while keeping the structure in (12).
- But this cannot work for examples like (14a) in which the first expression is clearly modifying the expressive and *not* the argument of the expressive.

- Potts (2007*a*) presents a revised approach that ensures that *that fucking bastard Burns* is expressively stronger than *that bastard Burns* while keeping the structure in (12).
- But this cannot work for examples like (14a) in which the first expression is clearly modifying the expressive and *not* the argument of the expressive.

(14) a. Holy shit, my bike tire is flat again!

- Potts (2007*a*) presents a revised approach that ensures that *that fucking bastard Burns* is expressively stronger than *that bastard Burns* while keeping the structure in (12).
- But this cannot work for examples like (14a) in which the first expression is clearly modifying the expressive and *not* the argument of the expressive.

(14) a. Holy shit, my bike tire is flat again!
b. Shit, my bike tire is flat again!

- Potts (2007*a*) presents a revised approach that ensures that *that fucking bastard Burns* is expressively stronger than *that bastard Burns* while keeping the structure in (12).
- But this cannot work for examples like (14a) in which the first expression is clearly modifying the expressive and *not* the argument of the expressive.

(14) a. Holy shit, my bike tire is flat again!
b. Shit, my bike tire is flat again!
c. *Holy, my bike tire is flat again!

- The same holds for *intensifiers* (Schwager & McCready 2009) modifying an expressive.

- The same holds for *intensifiers* (Schwager & McCready 2009) modifying an expressive.

(15) a. Dieser voll beschissene Idiot Peter ist zu spät.
»This totally shitty idiot Peter is too late«.

- The same holds for *intensifiers* (Schwager & McCready 2009) modifying an expressive.

(15) a. Dieser voll beschissene Idiot Peter ist zu spät.
»This totally shitty idiot Peter is too late«.
b. *Dieser voll Peter ist zu spät.

- The same holds for *intensifiers* (Schwager & McCready 2009) modifying an expressive.

(15) a. Dieser voll beschissene Idiot Peter ist zu spät.
»This totally shitty idiot Peter is too late«.
b. *Dieser voll Peter ist zu spät.

(16) a. That absolutely fucking bastard Burns got promoted.

- The same holds for *intensifiers* (Schwager & McCready 2009) modifying an expressive.

(15) a. Dieser voll beschissene Idiot Peter ist zu spät.
»This totally shitty idiot Peter is too late«.
b. *Dieser voll Peter ist zu spät.

(16) a. That absolutely fucking bastard Burns got promoted.
b. *That absolutely bastard Burns got promoted.

- That it is the expressive that is modified is also suggested by case marking in languages like German.

- That it is the expressive that is modified is also suggested by case marking in languages like German.

(17) a. Verdammt-e Scheiße, mein Fahrrad hat wieder einen Platten!
damn.FEM shit.FEM my bike has again a flat

- That it is the expressive that is modified is also suggested by case marking in languages like German.

(17)
a. Verdammt-e Scheiße, mein Fahrrad hat wieder einen Platten!
damn.FEM shit.FEM my bike has again a flat

b. Verdammt-er Mist, mein Fahrrad hat wieder einen Platten!
damn.MASC shit.MASC my bike has again a flat

»Damn shit, my bike tire is flat again!«

- That it is the expressive that is modified is also suggested by case marking in languages like German.

(17) a.

Verdammt-e	Scheiße,	mein	Fahrrad	hat	wieder	einen	Platten!
damn.FEM	*shit.FEM*	*my*	*bike*	*has*	*again*	*a*	*flat*

b.

Verdammt-er	Mist,	mein	Fahrrad	hat	wieder	einen	Platten!
damn.MASC	*shit.MASC*	*my*	*bike*	*has*	*again*	*a*	*flat*

»Damn shit, my bike tire is flat again!«

(18) a.

Das	verdammt-e	Arschloch	Peter	hat	mich	abgezockt.
the	*damn.NEUT*	*asshole.NEUT*	*Peter.MASC*	*has*	*me*	*off-ripped*

»That damn asshole Peter ripped me off!«

- That it is the expressive that is modified is also suggested by case marking in languages like German.

(17) a. Verdammt-e Scheiße, mein Fahrrad hat wieder einen Platten!
damn.FEM shit.FEM my bike has again a flat

b. Verdammt-er Mist, mein Fahrrad hat wieder einen Platten!
damn.MASC shit.MASC my bike has again a flat
»Damn shit, my bike tire is flat again!«

(18) a. Das verdammt-e Arschloch Peter hat mich abgezockt.
the damn.NEUT asshole.NEUT Peter.MASC has me off-ripped
»That damn asshole Peter ripped me off!«

b. *Das verdammt-er Arschloch Peter hat mich abgezockt.
the damn.MASC asshole.NEUT Peter.MASC has me off-ripped

Problem 2: Mixed expressives

Claim (2)

(19) No lexical item contributes both an at-issue and a CI-meaning. (Potts 2005: 7)

Claim (2)

(19) No lexical item contributes both an at-issue and a CI-meaning. (Potts 2005: 7)

⇒ No mixed expressives!

There are no expressions that contribute to both dimension of meaning.

But: Some of most clearest cases of expressives contribute to both dimensions!

But: Some of most clearest cases of expressives contribute to both dimensions!

Rascist swear words

(20) a. Lessing was a Boche. (Williamson 2009)
b. Hitler was a Kraut. (Saka 2007)

But: Some of most clearest cases of expressives contribute to both dimensions!

Rascist swear words

(20) a. Lessing was a Boche. (Williamson 2009)
b. Hitler was a Kraut. (Saka 2007)

Expressive content

The speaker has a negative attitude towards Germans.

But: Some of most clearest cases of expressives contribute to both dimensions!

Rascist swear words

(20) a. Lessing was a Boche. (Williamson 2009)
b. Hitler was a Kraut. (Saka 2007)

Expressive content

The speaker has a negative attitude towards Germans.

Descriptive content

Lessing/Hitler was a German.

- More generally, every expression with an expressive connotation is a mixed expressive.

- More generally, every expression with an expressive connotation is a mixed expressive.

(21) a. *Köter* $\rightsquigarrow$ dog + negative attitude

- More generally, every expression with an expressive connotation is a mixed expressive.

(21) a. *Köter* $\rightsquigarrow$ dog + negative attitude
b. *Bulle* $\rightsquigarrow$ policeman + negative attitude

- More generally, every expression with an expressive connotation is a mixed expressive.

(21)
a. *Köter* $\rightsquigarrow$ dog + negative attitude
b. *Bulle* $\rightsquigarrow$ policeman + negative attitude
c. *Sie* $\rightsquigarrow$ the hearer + expressing a formal relationship

- More generally, every expression with an expressive connotation is a mixed expressive.

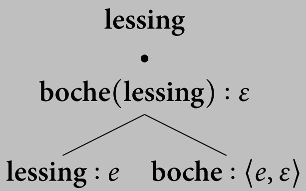
(21) a. *Köter* $\rightsquigarrow$ dog + negative attitude
b. *Bulle* $\rightsquigarrow$ policeman + negative attitude
c. *Sie* $\rightsquigarrow$ the hearer + expressing a formal relationship

- For a lot of examples of mixed expressives in Japanese, cf. McCready (2009).

- Furthermore, if *Boche* is treated as a simple expressive, $\mathcal{L}_{\mathrm{CI}}$ makes the wrong predictions about the descriptive meaning.

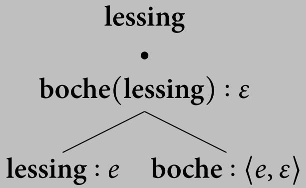
- Furthermore, if *Boche* is treated as a simple expressive, $\mathcal{L}_{\mathrm{CI}}$ makes the wrong predictions about the descriptive meaning.

(22)



- Furthermore, if *Boche* is treated as a simple expressive, $\mathcal{L}_{\mathrm{CI}}$ makes the wrong predictions about the descriptive meaning.

(22)



- A further argument is that the contribution of the past tense in *Lessing was a Boche* has to apply to the descriptive component since the expressive component cannot be shifted to the past.

Extending the system

- In order to overcome this problems, I suggest to extent the type system of $\mathcal{L}_{\text{CI}}$.

- In order to overcome this problems, I suggest to extent the type system of $\mathcal{L}_{\mathrm{CI}}$.
- To allow for expressive modifiers, I fill one of the gaps in the type definition.

- In order to overcome this problems, I suggest to extent the type system of $\mathcal{L}_{\text{CI}}$.
- To allow for expressive modifiers, I fill one of the gaps in the type definition.

Types for $\mathcal{L}_{\text{CI +EM}}$

(23) a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.

- In order to overcome this problems, I suggest to extent the type system of $\mathcal{L}_{\text{CI}}$.
- To allow for expressive modifiers, I fill one of the gaps in the type definition.

Types for $\mathcal{L}_{\text{CI +EM}}$

(23)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle\sigma, \tau\rangle$ is a descriptive type.
d. If σ is a descriptive type and τ is a (hybrid or pure) expressive type, then $\langle\sigma, \tau\rangle$ is a hybrid expressive type.

- In order to overcome this problems, I suggest to extent the type system of $\mathcal{L}_{\text{CI}}$.
- To allow for expressive modifiers, I fill one of the gaps in the type definition.

Types for $\mathcal{L}_{\text{CI +EM}}$

(23)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle\sigma, \tau\rangle$ is a descriptive type.
d. If σ is a descriptive type and τ is a (hybrid or pure) expressive type, then $\langle\sigma, \tau\rangle$ is a hybrid expressive type.
e. If σ and τ are (hybrid or pure) expressive types, then $\langle\sigma, \tau\rangle$ is a pure expressive type.

Comparison between $\mathcal{L}_{\text{CI}}$ and $\mathcal{L}_{\text{CI}\ +\text{EM}}$:

	T	[redacted]	E'
T	T	[redacted]	E'
[redacted]			
s	T	[redacted]	E'

Table: The type system of $\mathcal{L}_{\text{CI}}$

Comparison between $\mathcal{L}_{\text{CI}}$ and $\mathcal{L}_{\text{CI +EM}}$:

	T	[redacted]	E'
T	T	[redacted]	E'
[redacted]	[redacted]	[redacted]	[redacted]
s	T	[redacted]	E'

Table: The type system of $\mathcal{L}_{\text{CI}}$

	T	E	E'
T	T	E'	E'
E	[redacted]	E	E
E'	[redacted]	E	E
s	T	E	E'

Table: The type system of $\mathcal{L}_{\text{CI+EM}}$

Comparison between $\mathcal{L}_{\text{CI}}$ and $\mathcal{L}_{\text{CI +EM}}$:

	T		E'
T	T		E'
s	T		E'

Table: The type system of $\mathcal{L}_{\text{CI}}$

	T	E	E'
T	T	E'	E'
E		E	E
E'		E	E
s	T	E	E'

Table: The type system of $\mathcal{L}_{\text{CI+EM}}$

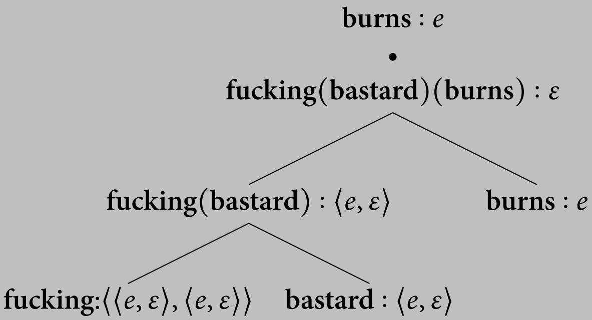
A new tree-admissibility condition

(24)

$$\alpha(\beta) : \tau^{\varepsilon}$$

$$\alpha(\sigma^{\varepsilon}, \tau^{\varepsilon}) \qquad \beta : \tau^{\varepsilon}$$

- Equipped with these rules and types, we can provide an intuitive semantics for expressve modifiers.



- To account for mixed expressives, I adopt a type definition from McCready (2009):

- To account for mixed expressives, I adopt a type definition from McCready (2009):

Types for $\mathcal{L}_{\text{CI}\ +\text{EM}+\text{ME}}$

(25)
a. e and t are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type and τ is a (hybrid or pure) expressive type, then $\langle \sigma, \tau \rangle$ is a hybrid expressive type.
e. If σ and τ are (hybrid or pure) expressive types, then $\langle \sigma, \tau \rangle$ is a pure expressive type.

- To account for mixed expressives, I adopt a type definition from McCready (2009):

Types for $\mathcal{L}_{CI\ +EM+ME}$

(25)
a. *e* and *t* are descriptive types.
b. ε is an expressive type.
c. If σ and τ are descriptive types, then $\langle \sigma, \tau \rangle$ is a descriptive type.
d. If σ is a descriptive type and τ is a (hybrid or pure) expressive type, then $\langle \sigma, \tau \rangle$ is a hybrid expressive type.
e. If σ and τ are (hybrid or pure) expressive types, then $\langle \sigma, \tau \rangle$ is a pure expressive type.
f. If σ and τ are descriptive type and υ is a pure expressive type, then $\langle \sigma, \tau \rangle \diamond \langle \sigma, \upsilon \rangle$ is a mixed type.

- A corresponding tree-admissibility condition distributes one argument to both parts of a mixed expressive, isolates the expressive content, and passes the descriptive content up the tree.

(26)

$$\alpha(\gamma) : \tau$$

$$\bullet$$

$$\beta(\gamma) : \upsilon$$

$$\alpha : \langle \sigma, \tau \rangle \diamond \beta : \langle \sigma, \upsilon \rangle \qquad \gamma : \sigma$$

- Equipped with these rules and types, we can provide an adequate semantics for mixed expressives.

(27)

$$\textbf{german}(\textbf{lessing}) : t$$
$$\bullet$$
$$\textbf{neg-att}(\textbf{lessing}) : \varepsilon$$

$$\textbf{german} : \langle e, t\rangle \diamond \textbf{neg-att} : \langle e, \varepsilon\rangle \qquad \textbf{lessing} : e$$

A remaining problem

2-place expressives

- Even $\mathcal{L}_{\text{CI+EM+ME}}$ doesn't allow for 2-place expressives.

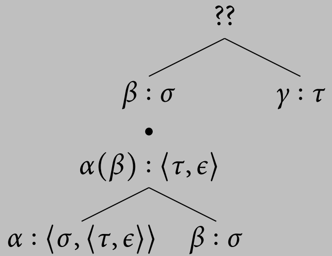
2-place expressives

- Even $\mathcal{L}_{\text{CI+EM+ME}}$ doesn't allow for 2-place expressives.
- Although the type for such expressions is defined in $\mathcal{L}_{\text{CI}}$, they cannot be computed.

2-place expressives

- Even $\mathcal{L}_{\text{CI+EM+ME}}$ doesn't allow for 2-place expressives.
- Although the type for such expressions is defined in $\mathcal{L}_{\text{CI}}$, they cannot be computed.

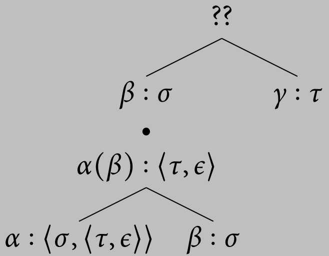
(28)



2-place expressives

- Even $\mathcal{L}_{\text{CI+EM+ME}}$ doesn't allow for 2-place expressives.
- Although the type for such expressions is defined in $\mathcal{L}_{\text{CI}}$, they cannot be computed.

(28)



- However, I am not sure at the moment whether 2-place expressives are actually needed.

Summary

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:
  - Expressives are never arguments.

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:
  - Expressives are never arguments.
  - No expression contributes to both dimensions of meaning.

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:
  - Expressives are never arguments.
  - No expression contributes to both dimensions of meaning.
- I have challenged both claims.

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:
  - Expressives are never arguments.
  - No expression contributes to both dimensions of meaning.
- I have challenged both claims.
- To account for expressive modifiers and mixed expressives, I added new types and tree-admissibility to $\mathcal{L}_{\mathrm{CI}}$.

- In his influential work on expressives, Potts (2005, 2007*b*,*c*) makes two claims:
  - Expressives are never arguments.
  - No expression contributes to both dimensions of meaning.
- I have challenged both claims.
- To account for expressive modifiers and mixed expressives, I added new types and tree-admissibility to $\mathcal{L}_{\text{CI}}$.
- A remaining problem(?): Even $\mathcal{L}_{\text{CI+EM+ME}}$ still predicts that there are no 2-place expressives.

Thank you for your attention

References

Geurts, Bart (2007): »Really fucking brilliant«. In: *Theoretical Linguistics* 33.2, 209–214. 10.1515/TL.2007.013.

McCready, Eric (2009): »Varieties of conventional implicature. Evidence from Japanese«. Ms. Tokyo: Aoyama Gakuin University. URL: http://semanticsarchive.net/Archive/DdmMWQyM/.

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

Potts, Christopher (2007*a*): *Expressives (Issue of Theoretical Linguistics 33)*. Berlin and New York: de Gruyter.

Potts, Christopher (2007*b*): »The centrality of expressive indices«. In: *Theoretical Linguistics* 33.2, 255–268. 10.1515/TL.2007.019. URL: http://people.umass.edu/potts/papers/potts-tl-reply.pdf.

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

Saka, Paul (2007): *How to Think about Meaning*. (Philosophical Studies Series 109). Dordrecht: Springer.

Schwager, Magdalena & Eric McCready (2009): »Intensifiers«. In: *DGfS 31. Workshop in Expressives and other Kinds of Non-truth-conditional Meaning*. University of Osnabrück.

Williamson, Timothy (2009): »Reference, inference and the semantics of pejoratives«. In: Almog, Joseph & Paolo Leonardi, eds. (2009): *The Philosophy of David Kaplan*. Oxford: Oxford University Press, 137–158.