Proceedings of SALT Volume 20, 1–15, 2010
doi: (no doi yet)

Principles of interdimensional meaning interaction [*]

Chris Barker
New York University

Raffaella Bernardi
Free University of Bozen/Bolzano

Chung-chieh Shan
Rutgers University

Keywords: Expressives, supplements, at-issue, continuations, indexicals, Lambek, Grishin

1 Introduction

Expressions that contribute side issues (including at least expressives, supplements, and epithets) challenge traditional conceptions of composition: usually, functors have full power to inspect and modify the value provided by their syntactic complements, but side issues cannot be modified by other semantic elements.

The semantic inertness of side issues combines two factors: first, that they always can take widest semantic scope. As Harris & Potts (2009) put it, "Expressives and appositives can scope out of the usual array of presupposition holes (interrogatives, negations, conditional antecedents, and modals) [..] they can remain semantically unmodified by the presupposition holes even when in their *syntactic* scope." For instance:

(1) If that bastard Kaplan was promoted, then the Regents acted foolishly.

The negative component of the epithet *bastard* is predicated of Kaplan at the level of the utterance. That is, the sentence as a whole does not imply that Kaplan was promoted, because ordinary content in an antecedent is only temporarily added to the context set during the evaluation of a conditional; but the sentence as a whole does imply that Kaplan is a bastard (more precisely, that the speaker is disposed to imply that he is). So the affective part of the epithet has escaped from the antecedent.

The second factor is that side-issues *must* take widest scope. For instance, in (1), there is no option to construe *bastard* as local to the antecedent (i.e., as contingent on Kaplan's promotion).

So if expressions that launch side-issues make contributions that take effect only at the level of the entire utterance, how are we to build a compositional semantics that guarantees the appropriate sort of non-interaction?

We draw an analogy between expressions that launch side issues and indexicals. Indexicals (typically) also find their values only at the level of the complete utterance,

[*] Authors in alphabetical order. Thanks to audiences at NYU and the Vancouver SALT.

and likewise are not affected semantically by surrounding material. Just as the non-interaction properties of indexicals motivates a multi-stage interpretation (the outer stage is the computation of a character, which, when combined with context, determines the inner stage, i.e., the propositional content), we will propose an innermost stage of at-issue evaluation followed by an outer stage at which expressive and other not-at-issue content takes effect.

More concretely, we will show how the undelimited continuations provided by Bernardi & Moortgat's (2010) Lambek Grishin calculus can provide a principled way to regulate interdimensional interaction.

2 At-issue versus side-issue entailments

Roughly put, at-issue content is what a sentence offers as its news. In a declarative assertion, it is the main propositional content proffered by the speaker for the addressee to consider adding to the common ground. Side-issues, in contrast, are entailments that are not at issue.

(2) Presupposition: John put his cat in the car.
Side-issue: John has a cat.

(3) Supplements: John, who is Italian, arrived.
Side-issue: John is Italian.

(4) Expressives: John put the damn garbage in the bin.
Side-issue: BAD [see below]

(5) Epithets: John asked the jerk to leave.
Side-issue: The person John asked to leave is a jerk.

Furthermore, the side issues of at least expressives, supplements, and epithets are speaker-oriented (Potts 2005). That is, in (3), it is the speaker who is committed to the claim that John is Italian; in (4), it is the speaker who has the negative attitude towards John's taking out the garbage, not John; and in (5), it is the speaker who is suggesting that the person who was asked to leave is a jerk. In addition, the world and time at which the speaker's attitude holds is always the world and time of the utterance situation. In other words, the content of expressives (supplements, epithets) is *non-displaceable* (Potts 2007): it always describes some aspect of the discourse situation (usually, the beliefs of the speaker).

A caveat before proceeding: the claim that expressives in particular are always speaker-oriented has been disputed.

(6) Well, in fact Monty said to me this very morning that he hates to mow the friggin lawn. [Harris & Potts 2009: 6]

Some informants report that it is possible to utter (6) intending for the negative attitude to be attributed to Monty rather than to the speaker. We follow Harris and

Potts in assuming that this a purely pragmatic effect, a kind of role-shifting on the part of the speaker in which the speaker pretends for rhetorical effect to speak as if from Monty's perspective. Semantically, then, we will assume that expressives are uniformly non-displaceable and speaker-oriented.

3 The non-interaction constraint

Although presuppositions are not at-issue, they differ from supplements, expressives, and epithets in ways that are important for the discussion below. In particular, they interact with surrounding at-issue material in intricate ways.

(7) a. If I had a cat, I would talk about my cat.
b. If I hated cats, I would talk about my damn cat.

In (7a), the at-issue content of the antecedent (that the speaker has a cat) can serve to satisfy the presupposition of the consequent, so that the sentence as a whole does not project any presupposition that the speaker owns a cat. In contrast, in (7b), the notion that the speaker has a negative attitude towards cats expressed by the antecedent has no effect on the side-issue launched by the expressive in the consequent: the sentence as a whole still implies that the speaker has negative affect towards some situation involving a cat. Simons, Tonhauser, Beaver & Roberts 2010 explain the behavior of presupposition projection in terms of conversational goals, but it is not clear to us how their account explains the contrast in (7).

To emphasize that side-issue content cannot interact with surrounding material, imagine a word *negex* that could capture and manipulate side-issue content by reversing its polarity:

(8) John **negex** read the damn book.
Side-issue: John feels good about the book.

We take it that the non-existence of words like *negex* is not accidental, and follows from the fact that side-issues must take scope only at the utterance level. Once a side-issue has been launched, its composition is ballistic, that is, is not affected by surrounding material.

(9) John **exonly** read the damn book.
'The only thing John feels heightened emotions concerning is the book.'

Similarly, there is no word like *exonly* that converts side-issue content into at-issue content in the way that focus-sensitive operators such as *only* can convert focus/background articulation into at-issue truth conditions.

We can summarize these observations as follows:

Principle of non-interaction: Once launched, side-issue content does not interact with at-issue content.

This generalization should not be an accidental property of the grammar, but should follow from some architectural property of the compositional system.

We hasten to note that there is an obvious way in which at-issue and other dimensions are not fully independent from each other, since ‘expressive operators can reach into the descriptive realm to find their argument’ (Potts 2007). For instance, in *that bastard Kaplan*, the epithet *bastard* is predicated of Kaplan, where the individual in question is the normal at-issue contribution of the name *Kaplan*. Thus the formation of expressive content can depend on at-issue content. Along similar lines, Gutzmann 2010 surveys cases in which expressives can be embedded under modifiers (*complete bastard*), and in which expressions contribute at-issue content and expressive content simultaneously.

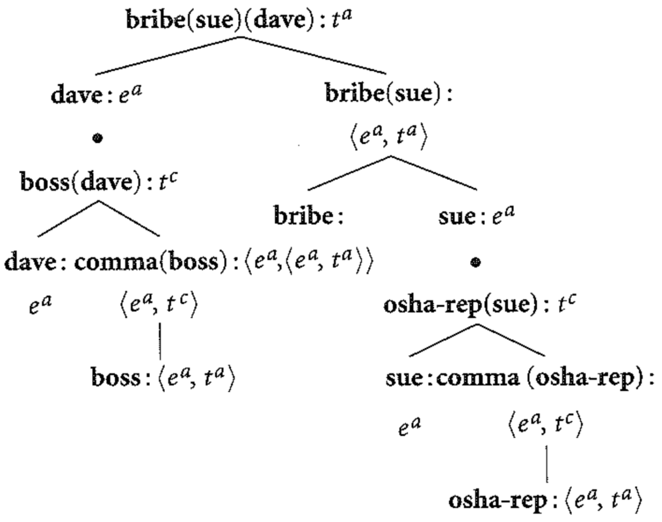
Another potential example of interaction that occurs before the side-issue has been formed: it appears that when quantifiers take scope over expressions that launch side-issues, it is possible for multiple side-issues to be launched, one for each quantificational case (see Amaral, Roberts & Smith 2007: 740).

But once the expressive proposition has been composed (e.g., that Kaplan is a bastard), it cannot be modified by or embedded under other semantic operators. We will say that once it has been formed, a side issue is launched, and proceeds ballistically to its target, which is the context within which the entire utterance is situated. It is the ballistic nature of this trajectory that we are investigating here.

4 Previous steps towards multi-dimensional composition

Potts (2007: 67) gives an explicit method for computing both the at-issue and not-at-issue dimensions of an expression’s meaning.

Dave, the boss, bribed Sue, an OSHA representative.


bribe(sue)(dave) : t^a
dave : e^a
bribe(sue) : $\langle e^a, t^a \rangle$
boss(dave) : t^c
bribe :
sue : e^a
dave : comma(boss) : $\langle e^a, \langle e^a, t^a \rangle\rangle$
e^a
$\langle e^a, t^c \rangle$
osha-rep(sue) : t^c
boss : $\langle e^a, t^a \rangle$
sue : comma (osha-rep) :
e^a
$\langle e^a, t^c \rangle$
osha-rep : $\langle e^a, t^a \rangle$

Parsetree interpretation proceeds by composing the at-issue content in the normal fashion, leaving each side issue dangling in the tree underneath an uninterpreted •. Parsetree interpretation is followed by a final step in which the side issues, no matter how deeply embedded, are gathered into a list, to which the at-issue proposition is added in a distinguished position.

Though formally precise, this method is not compositional. The reason is that the computation of the side issue content draws information from deeply embedded expressions (the supplement phrases), rather than only from the denotation of the sentence's immediate subconstituents.

As far as we know, the first strictly compositional treatment of expressive meaning is due to Kubota & Uegaki (2009).

$$\left(\begin{array}{c}\text{NP}_n\\ \textit{Taroo-ga}\\ \mathbf{t}\end{array}\;\left(\begin{array}{c}\text{NP}_d\\ \textit{Hanako-ni}\\ \mathbf{h}\end{array}\;\left(\begin{array}{c}\text{NP}_n\backslash\text{S}\\ \textit{piano-o hii-te}\\ \mathbf{playPiano}\end{array}\quad\begin{array}{c}(\text{NP}_n\backslash\text{S})\backslash\text{NP}_d\backslash\text{NP}_n\backslash\dfrac{\text{Assn}\,|\,\text{Assn}}{\text{S}}\\ \textit{morat-ta}\\ \lambda f\lambda x\lambda y.\dfrac{\mathbf{benef}(y,f(x))\wedge[\,]}{\mathbf{invlvd}(y,f(x))}\end{array}\right)\right)\right)$$

$$=\;\begin{array}{c}\dfrac{\text{Assn}\,|\,\text{Assn}}{\text{S}}\\ \textit{Taroo-ga Hanak-ni}\\ \textit{piano-o hii-te morat-ta}\\ \dfrac{\mathbf{benef}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))\wedge[\,]}{\mathbf{invlvd}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))}\end{array}\;\overset{\mathbf{A}}{\Rightarrow}\;\begin{array}{c}\dfrac{\text{Assn}\,|\,\text{Assn}}{\text{Assn}}\\ \textit{Taroo-ga Hanak-ni}\\ \textit{piano-o hii-te morat-ta}\\ \dfrac{\mathbf{benef}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))\wedge[\,]}{\mathbf{invlvd}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))}\end{array}$$

$$\overset{\text{Lower}}{\Rightarrow}\;\begin{array}{c}\text{Assn}\\ \textit{Taroo-ga Hanak-ni piano-o hii-te morat-ta}\\ \mathbf{benef}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))\wedge\mathbf{invlvd}(\mathbf{t},\mathbf{playPiano}(\mathbf{h}))\end{array}$$

Kubota and Uegaki make use of the tower notation of Barker & Shan 2008, in which syntactic categories and semantic values are each articulated into the at-issue portion (below the horizontal line) and the side-issue portion (above the horizontal line). Thus Kubota and Uegaki's analysis has the notable virtue of cleanly separating the at-issue dimension from the side-issue dimension.

Though fully compositional (indeed, even directly compositional, in the sense of Jacobson 2002), Kubota and Uegaki's analysis does not explain the non-interaction constraint. The reason is that expressives take scope over an expression with the special type "Assn" ('assertion'). Nothing prevents an expression from having a type that turns a complete assertion (a meaning representing a speech act) into, say, an ordinary verb phrase meaning. Thus the lexicon could contain the unattested *negex* operator described above, with the type $\dfrac{\text{Assn}\,|\,\text{Assn}}{\text{VP}}\Big/\dfrac{\text{Assn}\,|\,\text{Assn}}{\text{VP}}$ and the denotation $\lambda Q.\dfrac{\neg Q(\lambda P.\neg[\,])}{P}$.

Clearly, a speech act is not something that should be able to form a part of the composition of the sentence meaning. Speech acts inhabit a separate plane of existence. What we need is a way for the speech-act level of analysis of an utterance to be accessible for expressives and other not-at-issue expressions, but without giving expressions the power to turn speech acts into parts of propositional meanings. Put another way, the system should provide a ratchet mechanism that guarantees that once the propositional level has been completed and the speech act level entered, there is no going back to the propositional level.

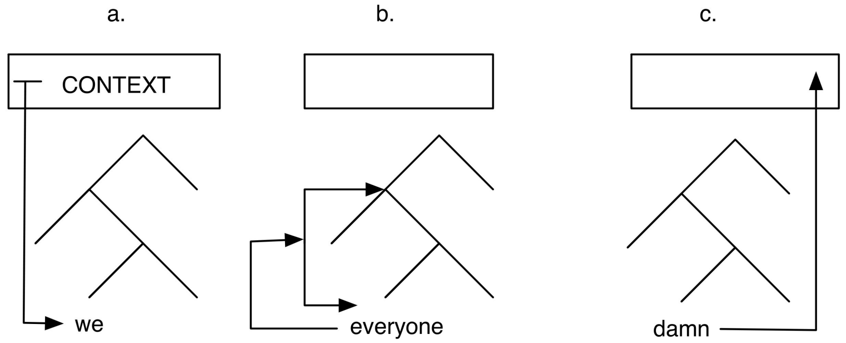
5 The analogy with indexicals: two-stage interpretation

We would like to suggest that exactly this kind of ratchet mechanism has long been familiar. There is a well-studied phenomenon that involves a semantic relationship between a deeply embedded expression and the discourse context, in which the semantic effect is never (well, almost never—see below) modulated by intervening material. The phenomenon we have in mind is the interpretation of indexicals.

(10) John believes that I am hungry.

In (10), John's belief must be about the speaker. Even if John thinks that the speaker is some other person, the indexical nevertheless still refers to the individual who is uttering the sentence. Just like the non-displaceability of expressives, the dependence of the content of indexicals on the discourse situation is insensitive to being embedded under attitude predicates, interrogatives, negation, conditional antecedents, and so on.

The difference, of course, between indexicals and expressives is that for indexicals, the indexical expression depends for its content on the discourse context, and does not change the ongoing context in any noticeable way. Expressives, on the other hand, do not depend on the discourse context in any noticeable way, but do change the ongoing context (by contributing information about the attitudes of the speaker). Indexicals and expressives are dual, then. This suggests the following typology of long-distance semantic relationships:



In (a), information flows from the context to the indexical. In (b), a quantifier might take scope over some portion of a sentence, simultaneously binding a variable in the original position of the quantifier, but without interacting with the context. In (c), information flows from an expressive directly to the utterance context, without interacting with the surrounding material.

On Kaplan's familiar analysis, the non-interaction of indexicals is captured by providing a two-stage interpretation procedure. In the first stage, context combines with the meaning of a sentence to determine a content, taken to be the proposition expressed by the sentence in that context. Thus sentences denote *characters*: functions from contexts to contents. The character of the indexical pronoun *I*, then, is a function from the discourse context to the speaker in that context, so the content of *I am hungry* is the proposition that that person is hungry. If the speaker is John, the content is the proposition that John is hungry; if the speaker is Mary, the content is the proposition that Mary is hungry. As a result, the content contains the speaker, not the indexical pronoun. That is why the indexical cannot be affected by semantic material intervening between the indexical pronoun and the root of the sentence—by the time the semantic value of the non-indexical material is evaluated, the indexical has already been replaced by its referent.

Of course, there are situations in which indexicals do interact with at-issue content (e.g., Schlenker 2003). We set such situations aside here, though it is intriguing to wonder what a language would be like that allowed similarly 'monstrous' expressives, expressions that systematically allowed expressive attitudes to be shifted in certain kinds of embedded contexts.

6 Towards a principled compositional account of non-interaction

Adding a layer to enforce indexical-style non-interaction is straightforward. Straightforward, at least, after Kaplan showed us how to do it: simply have expressions denote (recursive, compositional) functions from contexts to contents. Enforcing expressive-style non-interaction is more challenging, yet the solution is conceptually the same: once again, we make expressions functions from contexts to contents.

A function from contexts to contents is a continuation-passing computation. A number of projects have explored the relevance of continuations for understanding long-distance semantic phenomena, notably de Groote 2001, Barker 2002, Barker & Shan 2006, and Bernardi & Moortgat 2010. Because the key feature of expressives and other not-at-issue expressions is that they manipulate context, continuations are especially well suited for reasoning about such expression types.

In the computer science and logical literature, there are two kinds of continuations: undelimited continuations (e.g., Parigot 1993), which have types like $A \to \bot$, and which characterize the computational content of classical proofs; and delimited continuations (e.g., Felleisen 1988), which have types with an explicit result type, like $A \to R$, where R is the result type. The analysis of Kubota and Uegaki mentioned above makes use of delimited continuations.

The logical property of $\bot$, absurdity, is that once $\bot$ has been reached, there is no going back. That is, there is no practical use giving an expression a type like $\bot \to A$. Thus in a classical system, $\bot$ is a ratchet. Perhaps, then, if we had a system with undelimited continuations, we could identify Kubota and Uegaki's Assn type with $\bot$, and rule out monsters such as *negex*.

There have been several proposals to use undelimited continuations to describe natural language, notably de Groote 2001, and Bernardi & Moortgat 2010. In this note, we will describe the approach of Bernardi & Moortgat 2010.

7 Sketch of the symmetric Lambek Grishin calculus: from concatenation to continuations

Lambek turned Categorial Grammar (CG) into a proper logic. In the Lambek tradition, syntactic categories are seen as sets of expressions: those expressions that belong to a given category. Asking whether an expression belongs to a certain category is turned into asking whether the categories associated with the expression in the lexicon yield the category, namely whether the Cartesian product of the corresponding sets subsume the set of the given category. The answer is obtained by applying the logical rules of the system. In the simple case of "Sara left", the proof consists of just function application, which in Gentzen Sequent style is the $(\backslash L)$ step displayed on the right below.

$$\text{sara left} \in s? \qquad dp \otimes (dp\backslash s) \vdash s? \qquad \frac{dp \vdash dp \quad s \vdash s}{dp \otimes (dp\backslash s) \vdash s}\,(\backslash L)$$

$$\underbrace{\{\text{john, sara}\}}_{dp} \underbrace{\times}_{\otimes} \underbrace{\{\text{left, teases mary}\}}_{(dp\backslash s)} \underbrace{\subseteq}_{\vdash} \underbrace{\{\text{john left, sara left, john teases mary, sara teases mary}\}}_{s}$$

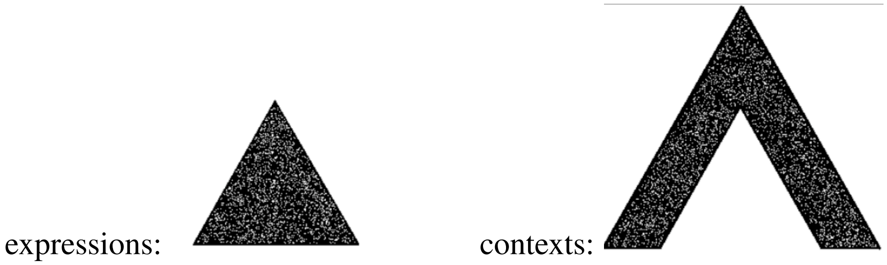
Since the number and order of occurrence of formulas (words) and their structure are important when dealing with natural language we will take the premise of the entailment to be structures. In Minimalist terms, we can say that the $\otimes$ *merges* expressions.

The switch made by Lambek from a grammar to a logic brings new features into the Categorial Grammar framework. First of all, besides the function application rules seen above, the system has rules for hypothetical reasoning too (the $(/R)$ below and the corresponding one for the $\backslash$ operator). Having both function application and hypothetical reasoning, the type lifting rule, from e to $(e \rightarrow t) \rightarrow t$, becomes a theorem of the logic. In other words, from "John" being in the category dp (considering it an entity, i.e. a term of type e) the system derives that it is also in the category of quantified dp ($s/(dp\backslash s)$ – a set of properties) as it would be necessary for, e.g. coordination of a DP and a quantificational DP.

$$\underbrace{\{\text{john}\}}_{dp}\ \underbrace{\subseteq}_{\vdash}\ \underbrace{\{\text{john, someone, everyone, some student}\ldots\}}_{s/(dp\backslash s)} \qquad \dfrac{\dfrac{dp \vdash dp \quad dp\backslash s \vdash dp\backslash s}{dp \otimes dp\backslash s \vdash s}\,(\backslash L)}{dp \vdash s/(dp\backslash s)}\,(/R)$$

Moreover, in van Benthem (1987), van Benthem showed that the logical system corresponds to (a fragment of) typed lambda calculus: each logical rule corresponds to a recipe for constructing a term in the lambda calculus; consequently to each proof corresponds a *proof-term* that encodes the sequence of rules applied. Hence, we can distinguish proof-term, that encodes the computation carried out to analyze syntactically the expression, from the meaning representation of the expression, that encodes its denotation. The latter is obtained after completing the syntactic analysis by replacing the proof-terms of the words in the structure with their lexical denotations.

Finally, the move from a grammar to a logic made it possible to introduce into the framework the concept of duality. In CG, as in many other formal grammars, the resources composed by the system are expressions. Taking a logic view highlights the fact that one can also compose the dual of expressions, namely *contexts*. An expression of type B is a structure, whereas a context of type B is a structure with a B-type hole:



Traditional Lambek calculus deals with expressions, its dual deals with contexts. The idea of a dual Lambek calculus was introduced already in Grishin (1983), and interactions between the two calculi had been studied by Grishin in Grishin (1983). Bernardi & Moortgat 2010 proposed to exploit his results for linguistics applications and called the system consisting of the Lambek calculus, its dual and the interactions between the two, Lambek-Grishin (LG).

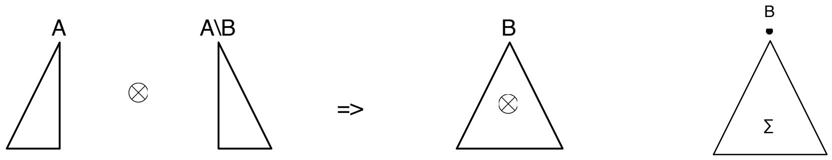
7.1 A Logic of Expressions, Contexts and their interaction: LG

LG has also been called Symmetric CG; let's see why. As we have seen, CG and the Lambek Calculus use the operators $\backslash, \otimes, /$ to merge expressions. The Dual Lambek Calculus composes contexts using their dual operators, $\oslash, \oplus, \obslash$, which are symmetric to the former with respect to the sequent turnstile ($\vdash$):

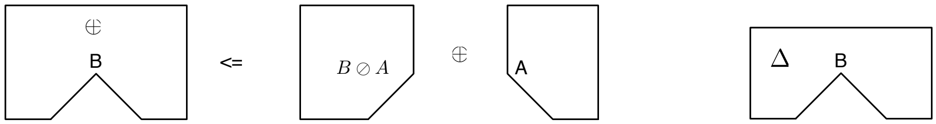
$$A \otimes A\backslash B \vdash B \qquad \text{symmetrically} \qquad B \vdash B \oslash A \oplus A$$

The function $(A\backslash B)$ turns an expression A into an expression B. The dual-function $(B \oslash A)$ turns the context of an A (the need of an A) into the context of a B (the need of a B). In other words, $\otimes$ assembles *expressions* to produce an expression (Merge), whereas $\oplus$ assembles contexts to produce a context (Co-merge). Hence, in the picture below Σ is a $\otimes$-structure which is proved to be of category B ($\Sigma \vdash B$) via the Lambek Calculus rules, and Δ is a $\oplus$-structure which is proved to be a B-context via the Dual Lambek Calculus rules ($B \vdash \Delta$) – we can read this as saying that the $\vdash$ is read right-to-left, viz. using a contrapositive reasoning.

Merging of Expressions

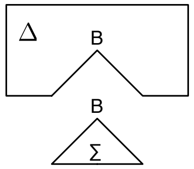


Co-merging of Contexts



When one has built both an expression of category B and a B-context the computation is over since the former fills in the hole of the latter. The rules performing this action is the “cut rule”.

$$\frac{\Sigma \vdash B \text{ and } B \vdash \Delta}{\Sigma \vdash \Delta}\ \text{(Cut)}$$



Grishin interaction postulates, the Mixed Associativity and Mixed 3-argument Commutativity below and their symmetric version (see Bernardi (2010) for further details) allow the Lambek and dual Lambek language to interact: a $\obslash$-category of the dual Lambek calculus can be merged with categories of the Lambek calculus. If embedded into the $\otimes$-structure, the Grishin postulates will allow the $\obslash$-category to reach the outermost position, from which it can be co-abstracted and be added into the $\oplus$-structure. Without the Grishin postulates, the co-abstraction could happen only when the $\obslash$-category occur in the outermost position.

Mixed Associativity	$(B\obslash C)\otimes A \vdash B\obslash(C\otimes A)$	$(\otimes$ MA)
Mixed 3-arg Commutativity	$A\otimes(B\obslash C) \vdash B\obslash(A\otimes C)$	$(\otimes$ MC)

$$\begin{array}{l} \dfrac{\Sigma[B] \vdash A \oplus \Delta}{A \obslash \Sigma[B] \vdash \Delta}\ \text{Co-Abs} \\ \quad\vdots\quad \text{Grishin postulates} \\ \Sigma[A \obslash B] \vdash \Delta \end{array}$$

similarly, for abstraction within a $\oplus$-structure.

$$\frac{B \vdash A \oplus \Delta}{A \obslash B \vdash \Delta}\ \obslash\text{L} \qquad \text{symmetrically} \qquad \frac{\Delta \otimes A \vdash B}{\Delta \vdash B/A}\ /\text{R}$$

As we have seen before, abstraction says that if Δ together with A produces a B, then Δ alone produces B provided an A is given. Co-abstraction says that if by adding up the A and Δ contexts (debits), we obtain the B contexts, then from the Δ context (debit), we obtain a B context (debit) without the A context.

Words that exhibit a mismatch between their syntactic and semantic behavior, like Quantifier Phrases (QPs) and other scope-taking expressions, require both bottom-up composition on the left of the turnstile and top-down composition on the right of the turnstile. Their syntactic categories are built with the dual operators that needs to be merged with the other expressions. For instance, the QP category introduced in Bernardi & Moortgat (2010) consists of a syntactic component (an expression, dp) that works locally—in a bottom up regime—and also a scope component (a context, $(s \oslash s)$) that works long-distance (not locally)—in a top-down regime: $(s \oslash s) \obslash dp$. Grishin interaction postulates make the two views work together.

For a full explanation of the analysis of quantification in LG the reader is referred to Bernardi & Moortgat (2010); Bernardi (2010). Recently, in Moortgat 2010 an alternative proposal has been made still based on the idea of contexts marked by unary operators instead of dual binary ones.

As mentioned above, the Logic of expressions, the Lambek Calculus, is in correspondence with the lambda-calculus and that their denotation can be represented by lambda-terms using a Direct interpretation. In the following, we are going to look at the interpretation of contexts and at their role in the assembly of meaning representation.

8 LG and Continuation Semantics

In the type-logical grammar tradition, syntactic derivations take the form of a logical proof. One advantage of this strategy is that proofs have computational content; this can make the syntax/semantics interface particularly straightforward. For instance, as mentioned, the Curry-Howard isomorphism maps proofs in intuitionistic logic into terms in the lambda calculus. Bernardi and Moortgat map their logic onto the lambda calculus in two steps: first, they map proofs onto Curien & Herbelin's (2000) $\bar{\lambda}\mu\widetilde{\mu}$-calculus, and then map $\bar{\lambda}\mu\widetilde{\mu}$-expressions via a Continuation Passing Transform (CPS) into the simply-typed linear lambda calculus. The CPS step is a standard technique for implementing a continuation-based semantics.

One point of interest for present purposes is that the LG system provides a special type R, the result type, standing for the result of a complete derivation. The continuation of an expression of type A in this system, then, is a function from objects of type A into R. The composition of expressions is recast in terms of continuations in the following way: instead of thinking of a simple functional expression like a VP of type $dp\backslash s$ as a function from individuals to truth values, we now think of it as a function from s continuations to dp continuations. That is, a verb phrase is the kind of expression that tells you how to turn a context that needs a sentence into a context that only needs an DP in order to be complete.

(11)
$$\begin{array}{ll}
\mathtt{Type}(s) & = t \\
\mathtt{Type}(dp) & = e \\
\mathtt{Type}(N) & = e \rightarrow t \\
\mathtt{Type}(A/B) = \mathtt{Type}(B\backslash A) & = (\mathtt{Type}(A) \rightarrow R) \rightarrow (\mathtt{Type}(B) \rightarrow R) \\
\mathtt{Type}(B \oslash A) = \mathtt{Type}(A \obslash B) & = ((\mathtt{Type}(A) \rightarrow R) \rightarrow (\mathtt{Type}(B) \rightarrow R)) \rightarrow R
\end{array}$$

Note the relation between e.g. $\oslash$ and \ $\mathtt{Type}(B \oslash A) = \mathtt{Type}(B\backslash A) \rightarrow R$.

9 Separating dimensions of meaning

So the Lambek-Grishin system provides undelimited continuations of semantic type $A \rightarrow R$. (Here R plays the semantic counterpart of $\bot$ in classical logic.) In the application of Lambek-Grishin to modeling the scope of quantifiers studied in Bernardi & Moortgat 2010 and Bernardi 2010, it is necessary to identify (i.e., equate) R with the semantic type of sentences, type $\mathtt{t}$. This identification occurs only in the semantics, at the level of the output of the CPS translation. Within the syntactic logic, expressions have no knowledge that $R = \mathtt{t}$, and it is only in their lexical denotations that quantifiers rely on the knowledge that continuations return results of semantic type $\mathtt{t}$. Furthermore, lexical denotations that are not quantificational do not rely on this knowledge; they have the correct semantic type according to (11) no matter what R turns out to be.

We propose to model non-interaction of expressives with at-issue content using the same mechanism of undelimited continuations. This application of Lambek-Grishin requires identifying R with the semantic type of speech acts, which, as discussed below, is not $\mathtt{t}$, but rather some kind of context update function.

Of course, if $R \neq \mathtt{t}$, then we cannot identify the same R with the type $\mathtt{t}$ to model quantifier scope. To accommodate both applications at once (quantification and side-issues), it would be nice to generalize Lambek-Grishin from a single result type to a hierarchy of result types: at least one result type for quantification and other at-issue content, and additional result types for not at issue content. We leave this intriguing possibility for future research.

9.1 At-issue versus side-issue update

To say what *damn* denotes, we must be more explicit about the result type R of speech acts. By their very nature, at-issue entailments are negotiable in a way that side-issue entailments are not. Therefore, we should not treat an assertion with both at-issue and side-issue content as just the conjunction of the two propositions. Rather, we should leave room for at-issue content and side-issue content to contribute information in different ways. Also, we should provide referents for indexicals, so it must have access to the identity of the discourse participants.

Building on the proposal of Murray 2010, we can model the side-issue and at-issue dimensions as different kinds of context update. Roughly put, side-issue entailments directly modify the context set, reflecting the fact that side-issue update is obligatory (non-negotiable). In contrast, at-issue content only *proposes* an update, which other discourse participants can accept, reject, or modify as they see fit. In Murray's implementation, side-issue content removes alternatives from the context set, but at-issue entailments merely impose an ordering on the updated context set, in such a way that the worlds that satisfy the at-issue entailments dominate the worlds that don't. A cooperative addressee will accept a proposed at-issue update by keeping the worlds that are highly ranked by the speaker and discarding the rest.

We will simplify Murray's proposal slightly for expository simplicity. Instead of imposing an ordering, a context update function will return a pair of propositions, where a proposition has type $\mathtt{s} \to \mathtt{t}$ (the characteristic function of a set of worlds). The first element of the pair of propositions will be interpreted as the set of worlds that satisfy background assumptions updated with all non-negotiable content. The second element of the pair will give the set of worlds that satisfy the at-issue entailments (in Murray's system, the highly-ranked worlds). Then an addressee can accept proffered at-issue content by simply taking the intersection of the two sets of worlds. An addressee rejects at-issue content by adopting the first set of worlds and discarding the second. Other conversational moves involve more complex functions.

Thus, we identify R with the type $\mathtt{e} \to (\mathtt{s} \to \mathtt{t}) \to ((\mathtt{s} \to \mathtt{t}) \times (\mathtt{s} \to \mathtt{t}))$: a function from the speaker and an input context set to a pair of propositions.

An indexical such as *my* will have the syntactic type dp/N, so its denotation should have the semantic type $(\mathtt{e} \to R) \to ((\mathtt{e} \to \mathtt{t}) \to R)$ (a function from DP continuations to nominal continuations). The denotation is

$$\lambda \kappa P s.\, \kappa(\iota x.\, Px \wedge \mathbf{of}\, s\, x) s. \tag{12}$$

This denotation receives a nominal meaning P, retrieves the speaker s of the utterance, and computes the individual x that satisfies the nominal property P and belongs to s. It then passes x to the continuation of the DP, without modifying the speaker s.

An expressive such as *damn* will have the syntactic type N/N, so its denotation should have the semantic type $((\mathtt{e} \to \mathtt{t}) \to R) \to ((\mathtt{e} \to \mathtt{t}) \to R)$. The denotation is

$$\lambda \kappa P s c.\, \langle (\lambda w.\, \mathbf{BAD}\, w \wedge (\pi_1(\kappa P s c) w)), \pi_2(\kappa P s c) \rangle. \tag{13}$$

Here, π_1 is the projection function returning the first element of an ordered pair; likewise π_2 returns the second element. This denotation receives a nominal meaning P, computes the context pair κPsc that would result from passing on P (and the current speaker s and the background context set c) unchanged, then restricts the first element of the resulting pair to reflect the information that the speaker has a negative attitude towards some aspect of the situation (the proposition **BAD**).

9.2 Characterizing non-interaction

We return to the main discussion, namely, accounting for non-interaction in a principled way. An at-issue expression is an expression whose denotation is *polymorphic* in the result type. For example, the ordinary (non-expressive) adjective *beautiful* has the same syntactic type (N/N) as *damn*, so its denotation will also have the semantic type $((\mathtt{e} \to \mathtt{t}) \to R) \to ((\mathtt{e} \to \mathtt{t}) \to R)$. Unlike *damn*, however, the denotation of *beautiful*

(14) $$\lambda\kappa P.\, \kappa(\mathbf{beautiful}\, P)$$

has that semantic type no matter what choice we make for R. The reason is that it does not make any assumptions about the result type of κ; whatever type of object κ chooses to return will be what (4) as a whole returns.

What damage to the side-issue update can be done by a denotation so polymorphic? Not much. The denotation receives a continuation κ and an argument P, then must return something of semantic type R. Without relying on what R is, the only way to manufacture any such result is to apply κ to something. Knowing that the argument type of κ is $\mathtt{e} \to \mathtt{t}$, the denotation can apply κ to a variety of arguments—not only $\mathbf{beautiful}\, P$ but also P, $\mathbf{ugly}\, P$, etc.—but it cannot tamper with the result returned by κ because it does not know the result type.

Formally, we have the following *free theorem* (Wadler 1989): if f is an adjective whose denotation is polymorphic in R, we have a guarantee that f will not interfere with side-issue content. In particular, we can be sure that the following holds:

(15) $$g(f\kappa P) = f(g \circ \kappa)P$$

for all g, κ, P.[1]

Here's how this equation expresses a version of the non-interaction constraint. Think of f as an ordinary adjective such as *beautiful*, which depends on the nominal meaning P. Then κ is the context surrounding the ⌜*beautiful* + nominal⌝ construction, and let g be a function of type $R \to R$, expressing whatever context update function is waiting for the result of the utterance. In the current implementation, g might

[1] (Function application is left associative). Note that f is used on the two sides of this equation in ways that may potentially deliver *different* result types. If this polymorphism seems distressing, consider the special case in which g has type $R \to R$.

be the function that makes a conversational move of accepting a newly-proffered content (or of rejecting it, etc.). Then if f is innocent in the relevant sense, we know that applying f to its argument and its continuation first and then applying g will produce the same result as composing g with the continuation before applying f. The fact that f is polymorphic in the type of its continuation argument means that it does not—indeed, it cannot—adjust the result delivered by surrounding context, so any side effects pass through unmolested.

If, in contrast, f is evil (in the way that *negex* is evil), the guarantee in (15) does not hold. For instance, let g be the rejection function (it keeps the first component of an update pair, but replaces the at-issue component with the maximally uninformative proposition, i.e., the constant function that maps every world to true). Furthermore, let f have some side effect; for simplicity, let f be factive: it will take any at-issue content and apply it as a filter on the background component (i.e., will presuppose it). Then applying f first and then g will result in the at-issue component being added to the background component, exactly as if the at-issue proposition had been accepted after all. But if we compose g with κ first, the at-issue component will be discarded before f can fold it into the background proposition, so the equation in (15) does not hold.

Now imagine writing out what f does: in part, it takes a continuation and intersects the two propositions. In order to accomplish this much, it must know something about the internal structure of the result type R. It must know that R has two components, merely in order to be able to extract the first component or the second component. Therefore this f will not be polymorphic with respect to R, and this is what leads to its failure to satisfy equation (15).

The bottom line: the reason at-issue expressions do not interact with side-issues is that at-issue meanings are polymorphic in a way that guarantees (15).

To be sure, there is much work left to do. All we have accomplished is to guarantee that if an expression contributes only at-issue content, it cannot interact with side-issues. But what about side-issue content itself? The lexical entry we have proposed for *damn* satisfies the equation in (15) for some choices of g but not others. In particular, *damn* obeys (15) when g is the accept update function or the reject update function, but not for arbitrary update functions. Because set intersection is commutative, a similar generalization will be true for any side-issue whose update effect is to impose intersective restrictions on the background context set.

We should also acknowledge that order of update still makes a difference for discourse anaphora and presupposition projection, important topics that we have not addressed except in passing here.

The next step, then, is to develop a theory of natural context update functions, and to integrate that theory with a general approach to presupposition and presupposition projection.

10 Conclusions

Some deeply embedded expressions launch side issues that can and must take scope at the level of a complete utterance. Once launched, side-issue content does not interact with at-issue content.

Kubota & Uegaki's (2009) analysis in terms of delimited continuations provides the first compositional treatment of multi-dimensional meaning, but does not capture the non-interaction constraint. The problem is that because the type system makes no formal distinction among types, nothing prevents operators from interacting with side-issue content. Thus it appears that delimited continuations alone are not sufficient for describing natural language, contra a conjecture in Barker 2004.

Associating side-issue content instead with undelimited continuations, as in Bernardi & Moortgat 2010, allows side issue content to take scope over expressions of type $\perp$ (corresponding to R in the semantics), a type that is safely beyond the reach of at-issue content. An expressive or a supplement can launch a side-issue that travels ballistically to the context update phase of processing, without any chance of interacting with intervening quantifiers or any other at-issue content. The result is a compositional system in which indexicals, quantifiers, and expressives all interact (or not!) in a principled way.

References

Amaral, P., C. Roberts & E.A. Smith. 2007. Review of The Logic of Conventional Implicatures by Chris Potts. *Linguistics and Philosophy* 30(6). 707–749.

Barker, C. 2002. Continuations and the nature of quantification. *Natural Language Semantics* 10. 211–242.

Barker, C. 2004. Continuations in natural language. In Hayo Thielecke (ed.), *Fourth ACM-SIGPLAN continuation workshop*, School of Computer Science, University of Birmingham.

Barker, C. & C. Shan. 2006. Types as graphs: Continuations in type logical grammar. *Journal of Logic, Language and Information* 15(4). 331–370.

Barker, C. & C. Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1(1). 1–46. doi:10.3765/sp.1.1.

Bernardi, R. 2010. Scope ambiguities through the mirror. In Everaert, Lentz, De Mulder, Nilsen & Zondervan (eds.), *The linguistics enterprise: from knowledge of language to knowledge in linguistics*, vol. 150, John Benjamins.

Bernardi, R. & M. Moortgat. 2010. Continuation semantics for the Lambek-Grishin calculus. *Information and Computation* 208(5). 397–416.

Curien, P.L. & H. Herbelin. 2000. The duality of computation. In *Proceedings of* *the fifth ACM SIGPLAN international conference on functional programming*, 233–243. ACM Press.

Felleisen, M. 1988. The theory and practice of first-class prompts. In *Proceedings of the 15th ACM SIGPLAN-SIGACT symposium on principles of programming languages*, 180–190. ACM Press.

Grishin, V.N. 1983. On a generalization of the Ajdukiewicz-Lambek system. In Mikhailov (ed.), *Studies in nonclassical logics and formal systems*, 315–334. Moscow: Nauka. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

de Groote, P. 2001. Type raising, continuations, and classical logic. In R. van Rooy & M. Stokhof (eds.), *Proceedings of the thirteenth Amsterdam Colloquium*, 97–101. ILLC, Universiteit van Amsterdam.

Gutzmann, Daniel. 2010. Expressive modifiers and mixed expressives. In O. Bonami & P. Cabredo Hofherr (eds.), *Empirical issues in syntax and semantics 8*, .

Harris, J.A. & C. Potts. 2009. Perspective-shifting with appositives and expressives. *Linguistics and Philosophy* 32(6). 523–552.

Jacobson, P. 2002. The (dis)organization of the grammar: 25 years. *Linguistics and Philosophy* 25(5). 601–626.

Kubota, Y. & W. Uegaki. 2009. Continuation-based semantics for conventional implicatures: The case of Japanese benefactives. In Satoshi Ito & Ed Cormany (eds.), *Semantics and linguistic theory XIX*, CLC Publications.

Moortgat, M. 2010. Symmetric categorial grammar: residuation and galois connections. *CoRR* abs/1008.0170.

Murray, S. 2010. A Hamblin semantics for evidentials. In Satoshi Ito & Ed Cormany (eds.), *Semantics and linguistic theory XIX*, CLC Publications.

Parigot, M. 1993. Classical proofs as programs. *Computational logic and proof theory* 263–276.

Potts, C. 2005. *The logic of conventional implicatures*. Oxford University Press, USA.

Potts, C. 2007. The expressive dimension. *Theoretical Linguistics* 33(2). 165–198.

Schlenker, P. 2003. A plea for monsters. *Linguistics and Philosophy* 26(1). 29–120.

Simons, Mandy, Judith Tonhauser, David Beaver & Craige Roberts. 2010. What projects and why. In D. Lutz & N. Li (eds.), *Semantics and linguistic theory 20*, CLC Publications.

van Benthem, J. 1987. Categorial Grammar and Lambda Calculus. In D. Skordev (ed.), *Mathematical Logic and its Applications*, 39–60. Plenum, New York.

Wadler, Philip L. 1989. Theorems for free! In *4th international conference on functional programming languages and computer architecture*, 347–359. ACM Press.