

Dispensing with unwanted polysemy: Deriving verbal idioms by coercive typing

1 The Problem. A puzzling problem of polysemy concerning idioms is their relationship with lexical items (*LIs*) that appear within them: what are formal properties of *show* and *the door* in the idiom *show someone the door*? On the one hand, there are arguments suggesting that idioms are derived from “standard” LIs (cf. Larson (2017)). On the other hand, idioms show effects suggesting the opposite. Apart from the fact that their meaning is not straightforwardly composable from such “standard” LIs, idioms tend to behave like atoms. First, they do not allow questions targeting proper idiomatic items, blocking *wh*-movement, as in (1b). Second, idiomatic items pose problems for co-indexing when taken as antecedents for pronouns, as in (2b).

- (1) a. What_{*i*} did they keep *t_i* at bay?
b. #Where_{*i*} did they keep the police *t_i*? [# stands for the lack of idiomatic reading]
- (2) a. Yesterday she showed Peter the door_{*i*}, but today she hired him_{*i*} again.
b. #Yesterday she showed Peter the door_{*i*}, but today I couldn’t find it_{*i*}.

So, taking the weakest assumption that idioms are composed of LIs which behave differently when enclosed within an idiom, we can itemize two general strategies present in the recent literature:

Strategy 1: LIs are polysemous embracing standard and idiomatic reading (Williams (2007)).
Consequence: We need a substantially enriched semantics selecting the right meaning (idiomatic or not) for a given phrase (cf. Mateu & Espinal (2007) and Gehrke & McNally (2017)).

Strategy 2: LIs are not polysemous; they are subject to separate computation rules, either pre-syntactic (Goldberg (2006)), syntactic (Chomsky (1980)), or post-syntactic (Chomsky (1995)).

Consequence: We need an even more enriched semantics, slicing semantic computation into layers of representation, each defined in terms of specially dedicated composition principles.

2 A middle path. While having its own pros, each of the two strategies poses new problems connected with increasing complexity of semantics. I show that combining Minimalist syntax with coercive typing opens up a middle path capitalizing on strong points of the two. It dispenses with polysemy (Strategy 2) without imposing ancillary layers of computation (Strategy 1).

The conception I shall make use of is that of roots, as originally proposed by Marantz (1997). Leaving aside numerous variations, the crucial idea is that LIs are uncategorised roots. To become interpretable, they must be merged with functional nodes. To illustrate with a toy example:

- (3) $v^0 : \lambda y_e. \lambda x_e. email(x, y) = 1$ iff x emails y ; $n^0 : \lambda x_e. email(x) = 1$ iff x is an email
- $$\begin{array}{c} \diagup \quad \diagdown \\ v^0 \quad \sqrt{email} \end{array}$$

$$\begin{array}{c} \diagup \quad \diagdown \\ n^0 \quad \sqrt{email} \end{array}$$

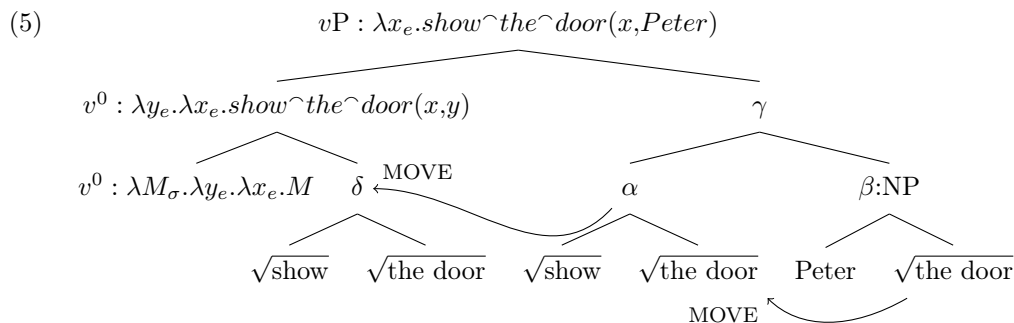
An LI *email* is an underspecified root which requires a functional node determining its interpretation. In this regard, functional nodes stand for a mechanism creating meaningful items.

This approach is potentially attractive. Roots are not polysemous but uncategorised. Still, there are no additional levels of computation – just the syntactic specification of roots, given a uniform semantic computation. Viewed from that angle, idioms could be conceived of as arising from basically the same mechanism as standard coercion rooted in predication, as in *start/throw a book*. The difference would lie in the coercing predicate: idioms would arise as an effect of coercion provided by functional nodes, not lexical predicates. This could naturally account for the unclear source of idiomatic reading: rather than depending on a particular predicate, it arises from enclosing a structure of LIs within the idiomatic context. The crux of the problem is how to define coercion in a way accounting not only for the emergence of idiomatic reading, but also for the atomic behaviour exemplified in (1)-(2). To show how this might work in type-theoretic

terms, I draw on the account of coercion given in Hindley & Seldin (2008) to propose the following principle, adjusted to strictly linguistic analysis:

- (4) **(Coercive typing):** If x^0 is a functional node and \sqrt{R} a root such that $\llbracket \sqrt{R} \rrbracket$ is of type σ , then $\llbracket x^0 \rrbracket(\llbracket \sqrt{R} \rrbracket) = \langle \lambda M_{\sigma} . \lambda y_a \dots M \rangle(\llbracket \sqrt{R} \rrbracket) \triangleright_{\beta} \lambda y_a \dots R(\dots, y)$, where:
- i. $(\lambda y_a \dots R(\dots, y))$ is of type $a \rightarrow \dots \rightarrow t$; a, \dots, t are type constants, $t \in \{0, 1\}$;
 - ii. $\ulcorner R \urcorner$ is a predicate constant represented by the phonological form of \sqrt{R} ; R takes as arguments all and only those variables that are bound in $(\lambda y_a \dots R(\dots, y))$.

The above predicate-building mechanism has an important consequence. If roots are not interpretable, then the lexical information that must be stored is that concerning the result of coercive typing. Thus if we assume that idioms are stored like other LIs being only phonologically complex (cf. Bruening (2015)), then they get a natural interpretation as a structure of roots merged with one functional node. To check how this works, take an especially challenging example as (2a) where an idiom is punctured by a non-idiomatic argument (irrelevant details omitted).



The idiom is derived from two roots ($\sqrt{\text{show}}$, $\sqrt{\text{the door}}$) and one argument *Peter*, specified in a lower part of derivation. After providing the right order *show-the-door*, $\sqrt{\text{the door}}$ moves to $\sqrt{\text{show}}$. To become interpretable, α moves to δ . v^0 coerces the whole δ , generating the transitive verb *show the door*. Its idiomatic reading is stored exactly like that of any other LI, without additional levels of computation. Moreover, coercion as defined in (4) allows to explain effects in (1)-(2): they naturally follow from the fact that *show the door* is a single predicate constant.

3. Conclusion. The proposed account has five consequences. (i) it dispenses with polysemy assuming standard and idiomatic readings of LIIs. (ii) it sets up a relation between lexicon and predicates letting the former be a set of phonological representations (roots), coerced to build predicates. (iii) it shows how coercion allows to tinker with terms, but economise both lexicon and semantics. (iv) it unearths a new dimension of coercion based on a different sort of coercing predicates. (v) it opens up new paths for exploring coercion as a word-formation mechanism.

References

- ## References
- Bruening, Benjamin. 2015. Idioms, Anaphora, and Movement Diagnostics. *MS*.
- Chomsky, Noam. 1980. *Rules and Representations*.
— 1995. *The Minimalist Program*.
- Gehrke, Berit & Louise McNally. 2017. Idioms and the syntax/semantics interface of descriptive content vs. reference. *MS*.
- Goldberg, Adele E. 2006. *Constructions at Work: The Nature of Generalization in Language*.
- Hindley, Roger J. & Jonathan P. Seldin. 2008. *Lambda-Calculus and Combinators: an Introduction*.
- Larson, Richard K. 2017. On “Dative Idioms” in English. *Linguistic Inquiry* 48 (3):389–426.
- Marantz, Alec. 1997. No Escape from Syntax: Don’t try Morphological Analysis in the Privacy of Your Own Lexicon. *University of Pennsylvania Working Papers in Linguistics* 4 (2):201–225.
- Mateu, Jaume & M Teresa Espinal. 2007. Argument structure and compositionality in idiomatic constructions. *The Linguistic Review* 24 (1):33–59.
- Williams, Edwin. 2007. Dumping Lexicalism. In Gillian Ramchand & Reiss Charles (Eds.), *The Oxford Handbook of Linguistic Interfaces*.