Pure Quotation and General Compositionality*

Peter Pagin
Stockholm University

Dag Westerståhl
Stockholm University
and
University of Gothenburg

Abstract

Starting from the familiar observation that no straightforward treatment of pure quotation can be compositional in the standard (homomorphism) sense, we introduce *general compositionality*, which can be described as compositionality that takes linguistic context into account. A formal notion of linguistic *context type* is developed, allowing the context type of a complex expression to be distinct from those of its constituents. We formulate natural conditions under which an ordinary meaning assignment can be non-trivially extended to one that is sensitive to context types and satisfies general compositionality. As our main example we work out a Fregean treatment of pure quotation, but we also indicate that the method applies to other kinds of context, e.g. intensional contexts.

1 The Problem

A straightforward treatment of quotation cannot be compositional. Consider the simplest case: *pure* quotation in *written language* by means of *quote marks* (we shall consistently use single quotes):

(1) a. 'Cicero' has six letters
b. 'farfalla' is Italian for butterfly
c. 'str jd e' is not a sentence
d. 'א' is a Hebrew letter

*Thanks to two referees, to Ken Shan who let us see his [36] and thereby put his remarks on compositionality and quotation into the context of the present paper, and to Chris Potts for expert editing. We also wish to thank Philippe de Brabanter, Herb Clark, Kathrin Glüer-Pagin, and Francois Recanati for stimulating discussions. The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 under grant agreement no. FP7-238128. The first author's work on this paper has also received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 229 441 - CCC, and from his research fellowship at IEA-Paris, spring 2010. The second author gratefully acknowledges support from the Swedish Research Council.

Although Cicero is Tully, and thus 'Cicero' and 'Tully' have the same meaning (taking the reference of names as their meaning), (1a) is not synonymous (on any account of meaning) with

(2) 'Tully' has six letters

This argument is familiar to everyone, but the observation about compositionality doesn't depend on taking the meaning of a name to be its bearer. Much less is required. To see this, let us be precise about what we mean by 'straightforward' here: A *straightforward account of* (the use of quote marks in) *pure quotation* is one which (a) takes the *quoted phrase* to be a syntactic constituent of the the *quoting phrase* (the quoted phrase surrounded by quote marks), and (b) allows at least one case of two syntactically distinct and quotable expressions having the same semantic interpretation (meaning).[1]

Then it should be uncontroversial that no such account can be compositional: By (a), application of quote marks is a syntactic rule, on a par with other syntactic rules used for building or analyzing sentences, so compositionality requires that there be a corresponding semantic operation yielding the meaning of the quoting expression from the meaning of the quoted expression and the quote marks rule, but (b) provides a counter-example.[2]

Indeed, non-compositionality is a main motive behind the wide variety of theories attempting to *reanalyze* quotation so that the quoted phrase is not a constituent of the quoting phrase. For example, the *proper name theory* (Tarski [38], Quine [29]) treats the quoting phrase as atomic or unanalyzable, no more containing the quoted phrase than the word 'Quine' contains 'in'. Other theories have analyzed quoting phrases as *descriptions* of how to form the quoted phrase from atomic expressions (letters or words) by means of concatenation (Tarski [37], Quine [30], Geach [10], Werning [40]).[3] Especially popular among philosophers is Davidson's *demonstrative* theory in [5], which takes (the occurrence of) the quote marks to *be* a demonstration of a token, the type of which is what the sentence is about.[4] Although reference to the quoted expression

[1] A quotable expression is one that can occur within quote marks as well as without them. So (b) is the very weak requirement that one's notion of meaning allows some *non-trivial synonymies*. Note that if a semantics disallows non-trivial sameness of meaning, it is trivially compositional (see (5) in Section 2 below), *regardless of how it treats quotation*, so the issue of whether a particular account of quotation is compositional or not becomes void.

[2] At least on the plausible assumption that if you enclose two expressions with distinct surface forms within quote marks, the resulting two quoting expressions never have the same meaning. For example, *'brother'* and *'male sibling'* (quote marks included) are trivially non-synonymous on any reasonable account of meaning, regardless of whether the quoted expressions are held to have the same meaning or not. That's just because no reasonable semantics should take two (non-indexical) referring expressions that refer to distinct objects to have the same meaning.

[3] Cappelen & Lepore [3] surveys theories of quotation. As these authors note, the proper name theory and the description theory fail to do justice to essential features of quotation in natural languages (this was however not the ambition of Tarski or Quine), and are now out of fashion. Werning [40] is a recent attempt to revive the description theory. Like all description theories, Werning's account is not straightforward in our sense (requirement (a) is not satisfied).

[4] For example, (1b) is analyzed as

occurs on this analysis, it is performed by a demonstrative device — the quote marks themselves — not by an expression containing the quoted expression as a part. There is a vast literature on the demonstrative theory; among recent works, see, for example, Cappelen & Lepore [2] and Predelli [28].

Another popular approach, called the *use theory* or — misleadingly — the *identity theory*, takes quotation to be a different *use* of words than the normal one, and quote marks as *indicators* of this use. The idea goes back to the medieval distinction between *suppositio materialis* and *suppositio formalis*. In modern times it occurs with Frege (see below); more recent variants can be found with Searle [35], Washington [39], Saka [34], Recanati [31], [32], and many others. As we will see, this idea is not necessarily incompatible with approaches taking quoting expressions to name quoted expressions, but many of its proponents take it to be radically different.[5]

One may discern some recent dissatisfaction with complicated analyses of what seems to be — in the pure case — a very simple phenomenon: quote marks are a productive and transparent device for forming names of linguistic entities, names that refer to these entities in just the same way as other names refer to non-linguistic entities. For example, this eliminates any problem whatsoever with *iterated* quotation, something which at least is an issue with other approaches.[6] Intuitively, this seems correct: there is no difference between quot-

(i) farfalla. That / The expression of which this is a token is Italian for butterfly

[5] E.g. Searle [35]. Recanati [31], [32] combines a pragmatic and a referential view: Pure quotations (*closed*, in his terminology) are referential singular terms, and he explicitly raises the issue of compositionality of the corresponding semantics, as we do in the present paper. *Open* quotations (e.g. indirect or mixed cases), on the other hand, do not refer; the words have their ordinary meaning but the *speech act* is different.

Some thoroughly pragmatic accounts (Clark and Gerrig [4]) deny that pure quotation is ever done by exhibiting the quoted object within some quotational device. The idea is instead that quotation *depicts selected aspects* of the object, even in quotation of written text. For instance, one may both include and exclude the case of the letters (upper, lower) as part of what is depicted. Moreover, C&G stress that the so-called *verbatim assumption*, that the reporter commits himself to repeating the actual words spoken (p. 795, actually a quote from Leech), is not in general respected in actual quotations, partly because it conflicts with the permission to select features.

We are here dealing only with written quotation of written text, and we attempt to provide a semantics for a quotation device that operates under the verbatim assumption (a difficult enough task). It is clear that other features than verbatim wording can be selected for quoting, e.g. letter case, font size, or typeface. This *can* be accommodated in our formal model by means of encoding the relevant properties in the grammatical terms (see Section 2.1), but we shall simplify matters by leaving this out.

It should be noted, however, that allowing for selection does *not* imply that what is quoted is only *depicted* in the quoting expression, i.e. *not* a syntactic part of it. On our view, written quotation proper (as opposed to the use of typographic features such as boldface for *representing* speech features, e.g. loudness of voice), quotation is a method of referring to types *exemplified* by tokens or occurrences in the quoting. Speakers indeed *do* represent features of original utterances by means of depicting or illustrating them, but only when what is represented is actually exemplified in the representing expression itself, is it quotation proper. And therefore we need not deviate from the straightforward view that quotation is a syntactic operation.

[6] It is not obvious how Davidson's original account indicated in footnote 4 can deal with *demonstrations within demonstrations*, as in

ing expressions surrounded by quote marks and quoting any other expressions. In other words, pure quotation should be straightforward. One elegant recent account along these lines is given in Potts [27]; indeed Potts deals not only with pure quotation but also direct and indirect discourse as well as mixed cases.

But, as we just saw, with a straightforward analysis the problem of compositionality returns. On this point, the relevant literature is unclear. Semanticists accounting for quotation, including Potts, generally claim that their accounts are compositional. Sometimes, the reason such a claim doesn't conflict with our initial observation is that the account is not straightforward in our sense.[7] But the treatment in Potts [27] *is* straightforward. It could of course be the case that different notions of compositionality are at stake. In fact, Potts replied to us (*p.c.*) that he required no more than that 'the syntax and the semantics work together in tandem', a slogan taken from the Introduction to Barker and Jacobson [1]. Here, on the other hand, although we have not (yet) given a precise definition, we have taken for granted that a *necessary* condition for compositionality is that substitution of synonyms preserves meaning. That this condition is violated in straightforward accounts of quotation seems uncontroversial, as noted. But the slogan just quoted says something much weaker, and has no implications at all for substitution of synonyms.[8]

(i) "farfalla" refers to 'farfalla',

and although Cappelen & Lepore [2], Predelli [28], and others suggest mechanisms for doing this, these are certainly more complex than the straightforward approach.

[7] This holds for Davidsonian variants, for the account in Werning [40], and also for the treatment in Shan [36]. Shan mentions that feature (a) fails in his fragment, i.e. that no rule allows to you to enclose an arbitrary expression in quote marks; he also notes that this is somewhat counter-intuitive (p. XXX). In fact, his account of mixed quotation is non-standard also in the sense of not satisfying the requirement, mentioned in footnote 2, that two syntactically distinct quoting expressions cannot have the same meaning. This is because the meaning he assigns a quoting expression isn't sensitive to the quoted expression itself, but only to the quoting context, which is what allows his account to be compositional. As to Shan's account of pure quotation (his section 4.1), it is compositional for the trivial reason that there are no distinct synonymous quoted expressions (since each such expression means itself).

[8] Potts's grammar generates triples $\mathcal{P} = \langle \Pi; \Sigma; \alpha : \sigma \rangle$, where Π is a phonological representation, Σ a syntactic category, and α a semantic representation of type σ. Letting $\mathit{SEM}(\langle \Pi; \Sigma; \alpha\!:\!\sigma \rangle) = \alpha$, the semantic interpretation function Potts (in fact) uses is

$$\mu(\mathcal{P}) = [\![\mathit{SEM}(\mathcal{P})]\!]$$

where $[\![\cdot]\!]$ gives model-theoretic denotations via a recursive truth definition. Quotation is the following syntactic rule, for $\mathcal{P} = \langle \Pi; \Sigma; \alpha\!:\!\sigma \rangle$,

$$q(\mathcal{P}) = \langle \Pi; \Sigma; \text{`}\mathcal{P}\text{'}\!:\!u \rangle$$

with the corresponding semantic interpretation rule

$$\mu(q(\mathcal{P})) = [\![\mathit{SEM}(q(\mathcal{P}))]\!] = [\![\text{`}\mathcal{P}\text{'}]\!] = \mathcal{P} \in D_u$$

where D_u is the set of grammatical triples (treated as a basic domain besides D_e and D_t). Given that the triples are the linguistic objects that get quoted, requirement (a) of a straightforward semantics is satisfied. But requirement (b) is also satisfied: Let $\mathcal{P}$ be the triple corresponding to *Cicero* and $\mathcal{Q}$ the one corresponding to *Tully*. Then $\mu(\mathcal{P}) = \mu(Q) \in D_e$ according to Potts's truth definition. But, by the above, $\mu(q(\mathcal{P})) \neq \mu(q(\mathcal{Q}))$.

Note here the distinction between *recursive* and *compositional* semantics. Potts's semantics is perfectly recursive: his rules allow you to *compute* the meanings of any well-formed triple in

This brings us to an important point. To make a meaningful claim about (non-)compositionality — one that has a chance of being true or false — two things need to be in place. *First*, you should have a particular syntax or grammar in mind, and a particular semantic interpretation. When a semanticist presents a fragment with a model-theoretic interpretation, this is usually not a problem, as long as it is clear that the claim concerns *that* syntactic analysis and *that* notion of meaning (as opposed to, say, English). The same surface strings may generate a counter-example to compositionality under one analysis (or one notion of meaning) but not under another. *Second*, you should have a clear notion of compositionality in mind. The slogan about syntax and semantics working in tandem is too vague to serve as a definition.[9]

Should we then despair of a compositional and straightforward treatment of pure quotation? With the standard notion of compositionality, the answer is Yes. In this paper we show, however, that there is an extended but still precise sense of compositionality in which the straightforward account *is* compositional. We call this *general compositionality*.

The quickest way to grasp this notion is to relate compositionality to *context dependence*. Consider first the familiar case of the extra-linguistic context dependence of indexicals and demonstratives The standard notion of compositionality applies directly to Kaplan's notion of *character*. Character is assigned to expressions, so we can ask if the character of a complex expression is determined by the characters of its immediate constituents (and the mode of composition). But this doesn't immediately apply to *content*, since content is assigned not to an expression but to an expression *and* a context. Nevertheless, no one takes this fact alone to show that content is not compositional. Indeed, Kaplan himself states explicitly in [16] that his account of content *is* compositional, and Lewis in [18] makes compositionality a necessary requirement on any

a finite number of steps. But this is not enough for compositionality, which requires that meanings of complex expressions be determined by the *meanings* of their immediate constituents (and the rule applied), whereas a recursive definition allows you to use not only those meanings but *the constituents themselves* as arguments of the composition operation. (In addition, it requires, in contrast with compositionality, that the composition operation is recursive; see Pagin & Westerståhl [23], Section 3.2.) This feature is essential for a (straightforward) formulation of the semantic rule for quotation, but, as we have been at pains to emphasize, the result is that substitution of synonyms does not preserve meaning, i.e. compositionality fails.

One might still think that since neither Potts's semantics nor ours is standardly compositional, there is no principled reason to prefer ours. However, beside the wide applicability of general compositional semantics, we can briefly point to another reason that because of space limitations we cannot here go deeply into. Semantic computational complexity, in the sense of [20], [22], does *not* increase with general compositionality compared with standard compositionality. By contrast, Potts's semantics for quotation has the effect that *iterated* quotation does increase computational complexity dramatically, because it leads to semantic values with iterated triples: triples that have other triples as their third element (and so on).

[9] We thus disagree with the view of compositionality expressed in Dowty [7], i.e. that the idea is fundamentally *vague*. We have no quarrel with Dowty's interesting discussion of how to treat various linguistic phenomena compositionally (the bulk of his paper), but we don't think compositionality is a vague notion. On the contrary, treating the grammar and the semantic interpretation as parameters, there are just a couple of notions of compositionality around, each of them precise (see our [23] for an overview).

assignment of semantic values. But, although only implicitly acknowledged by these authors, this requires an extended notion of *contextual compositionality*, one that takes the presence of the extra context argument into account.[10]

Our claim in this paper is that with a corresponding notion of *linguistic context*, the straightforward account of quotation can be made compositional, in the general, contextual sense. We shall not here discuss the intrinsic merits of compositionality. One could make an exception for quotation, and content oneself with a recursive but non-compositional semantics for a straightforward treatment of this phenomenon, as Potts *de facto* does (see footnote 8). That would be consistent with the view that quotation is a linguistic phenomenon without much interest. Such a view is not uncommon in the literature, but it is not Potts's view; on the contrary, he thinks quotation is a 'hugely important matter for linguistic theory' ([27], p. 425). Likewise, Cappelen and Lepore find it 'one of the most difficult and interesting topics in the philosophy of language' ([3], opening paragraph).

We agree that the way speakers 'talk about the words themselves' (see the quote from Frege below) is of significant interest, and we shall here take it for granted that a compositional account is worthwhile. And just as the presence of indexicals and demonstratives motivates an extension of the notion of compositionality to take extra-linguistic context into account, the presence of quotation motivates a generalization of compositionality that takes linguistic context into account. This can be seen already from Frege's familiar remark:

> If words are used in the ordinary way, what one intends to speak of is their reference. It can also happen, however, that one wishes to talk about the words themselves or their sense. This happens, for instance, when the words of another are quoted. One's own words then first designate words of the other speaker, and only the latter have their usual reference. We then have signs of signs. In writing, the words are in this case enclosed in quotation marks. Accordingly, a word standing between quotation marks must not be taken to have its ordinary reference. (Frege [8], pp. 58–9)

As Cappelen & LePore [3] note, this passage, which seems to be just about all Frege ever said about quotation, is probably too cryptic to be taken as a clear proposal.[11] But we shall interpret it as a *combination* of the use theory and the straightforward theory. It is straightforward in that the quoting phrase refers to the quoted phrase, but it is a use theory in that the quoted phrase itself, in the quotation context, has another meaning than it usually does: it refers to itself. So the quote marks are a syntactic operator, but they also signal a context change.

[10]Several writers are aware that the notion of compositionality needs to be extended to take context into account. Partee [26] has an extensive discussion without however arriving at explicit proposals. As we will see in Section 3 below, the formulation of context-dependent compositionality is slightly more subtle than one imagines.

[11]See Parsons [25] for a discussion of possible interpretations.

However, Frege also indicates that this sort of context dependence is not limited to quotation. He mentions talking about the *senses* of words, and we will see in Section 6 that one can construe his notion of *indirect* sense and reference as an attempt to give a general compositional semantics for intensional contexts. In fact, although in this paper we focus on quotation, there are several other applications. Moreover, we will see (Section 5.3) that the notion of general compositionality can be formulated in an equivalent way which doesn't mention contexts at all: the idea is to let a semantics consist of a *set* of meaning assignments rather than just one. This gives some indication of the robustness of the notion, and also paves the way for further applications.

We begin, in the next section, by presenting standard compositionality at a sufficiently abstract and precise level and, in the section after that, the extension of this concept to deal with non-linguistic context. Then we discuss linguistic context, and propose a formal version of this notion, which is used for the first formulation of general compositionality. We state the equivalence of this formulation to the second one (proofs of this and some other facts are relegated to Appendix 2) as well as some other facts about general compositionality. We also suggest a way to make precise the conditions under which an ordinary semantics can be said to *generalize* to a semantics which is compositional in the new sense. Finally, we show that the approach to quotation we take Frege to recommend is indeed a compositional generalization of a straightforward account of pure quotation, and in Appendix 1 we explain why an alternative approach that has been suggested would not be superior.[12]

2 Ordinary Compositionality

Compositionality is not a demand for a certain format of the grammar or the syntax-semantics interface. It is a condition that may or may not be satisfied by any semantically interpreted grammar in (almost) any format. This is clear already from the informal formulation of the compositionality principle, in either its functional or its substitutional version:

(PoC-F) The meaning of a complex expression is determined by the meanings of its immediate constituents and the mode of composition.

(PoC-S) Appropriate substitution of (not necessarily immediate) constituents with the same meaning preserves meaning.

These formulations are of course relative to given specifications of *meaning*, *constituent*, etc., but they are independent of *how* those specifications are made. For any assignment of *semantic values* to any set of *structured expressions* (expressions equipped with a notion of constituent or sub-expression), the issue of compositionality can be raised. Indeed, for (PoC-S) we don't even need to specify what the values are, only when two expressions have the *same* value. It

[12] A sketch of the ideas in this paper appeared in Pagin & Westerståhl [23] and [24].

follows that general facts about compositionality, such as the ones about quotation established in this paper, are (to a large extent) independent of particular formats chosen for syntax and semantics.

2.1 A formal framework

To make (PoC-F) and (PoC-S) precise we shall use the following framework.[13] Think of (surface) expressions as *strings*, and complex expressions as *generated* from *atomic* ones via grammar rules, which we simply take to be *partial functions* from tuples of expressions to expressions. There is no requirement on what these functions are; in particular, they are not restricted to concatenation of strings. Partiality is used to respect syntactic categories: rather than taking these as primitive, we let grammar rules be *undefined* for arguments of the wrong kind. To illustrate, rules like

NP $\longrightarrow$ Det N
S $\longrightarrow$ NP VP

correspond to binary partial functions, say α, β, such that, if *few*, *cat*, and *bark* are atoms, one derives the complex expression (sentence) *Few cats bark*, by first applying α to *few* and *cat*, and then applying β to the result of that and *bark*. These functions are necessarily partial; for example, β is undefined whenever its second argument is *cat*.

Thus, let a *grammar*

$$\mathbf{E} = (E, A, \Sigma)$$

be a partial algebra, where E is a set of expressions (strings), A its subset of atoms, and Σ is a set that, for each required $n \geq 1$, has a subset of partial functions from E^n to E, and is such that E is generated from A via Σ.

One and the same expression may be generated in more than way, i.e. the grammar may allow *structural ambiguity.* Also, a semantically relevant element need not be represented in the surface expression. For these reasons, it is not the expressions in E but rather their *derivation histories* ('analysis trees'), that are assigned semantic values. These derivation histories can be represented by the *terms* in the *partial term algebra* corresponding to $\mathbf{E}$. The sentence *Few cats bark* is a string, but its derivation history is represented by the term

(3) $\quad t = \beta(\alpha(\textit{few}, \textit{cat}), \textit{bark})$

in the term algebra. *Grammatical terms* are those where all the functions involved are defined for the respective arguments. So t is grammatical but $\beta(\alpha(\textit{few}, \textit{cat}), \textit{cat})$ is not. Let $GT_{\mathbf{E}}$ be the set of grammatical terms of $\mathbf{E}$.

We shall need to distinguish between the *functions* α, β ,... taking expressions to expressions and the *names* of these functions used as operators in the term algebra. The convention here will be that $\overline{\alpha}$ is a name of α. Thus,

[13] See Hodges [15] or Westerståhl [43] for more detailed accounts, and Pagin and Westerståhl [23] and [24] for further discussion and motivation.

$$\alpha(\textit{few}, \textit{cat}) = \textit{few cats}$$

is an element of E — the string *few cats*,[14] but the corresponding *term* is a formal expression in the term algebra. So rather than (3), we shall write, from now on,

(4) $\quad t = \overline{\beta}(\overline{\alpha}(\overline{\textit{few}}, \overline{\textit{cat}}), \overline{\textit{bark}})$

In other words, the $\overline{\alpha}$ for $\alpha \in \Sigma$ are the operations in the term algebra corresponding to $\mathbf{E}$. If we obey the constraints coming from the partiality of the functions in Σ, we get the grammatical terms. Forgetting those constraints we obtain the set $T_{\mathbf{E}}$ of arbitrary *terms*, obtained by successively applying the operations $\overline{\alpha}$ to any (term) arguments.

Each grammatical term maps to a unique string in E. In other words, there is a *string value function* V from $GT_{\mathbf{E}}$ to E. For an atomic term like $\overline{\textit{few}}$, we have $V(\overline{\textit{few}}) = \textit{few}$. To handle lexical ambiguity, we need distinct atomic terms with the same value, say, $V(\overline{\textit{bank}_1}) = V(\overline{\textit{bank}_2}) = \textit{bank}$. For a complex term $\overline{\alpha}(t_1, \ldots, t_n)$ we have

$$V(\overline{\alpha}(t_1, \ldots, t_n)) = \alpha(V(t_1), \ldots, V(t_n))$$

where α is defined for the arguments $V(t_1), \ldots, V(t_n)$ precisely when the term $\overline{\alpha}(t_1, \ldots, t_n)$ is grammatical. (So V is a homomorphism from the term algebra to the expression algebra $\mathbf{E}$.) To illustrate:

$$\begin{aligned}
V(\overline{\beta}(\overline{\alpha}(\overline{\textit{few}}, \overline{\textit{cat}}), \overline{\textit{bark}})) &= \beta(V(\overline{\alpha}(\overline{\textit{few}}, \overline{\textit{cat}})), V(\overline{\textit{bark}})) \\
&= \beta(\alpha(V(\overline{\textit{few}}), V(\overline{\textit{cat}})), \textit{bark}) \\
&= \beta(\alpha(\textit{few}, \textit{cat}), \textit{bark}) \\
&= \beta(\textit{few cats}\,, \textit{bark}) \\
&= \textit{few cats bark}
\end{aligned}$$

Now we can let a *semantics* for $\mathbf{E}$ simply be a partial function μ from $GT_{\mathbf{E}}$ to some set X of semantic values ('meanings'). The domain of μ can be a proper subset of $GT_{\mathbf{E}}$ if, for example, some grammatical terms have no semantic value because no meaning of the selected kind fits them, or because none has yet been found (say, the language is still partly unknown to the theorist).

As noted, for compositionality the nature of the meanings doesn't matter, only sameness of meaning. μ induces a partial equivalence relation on E: for $u, t \in E$, define

$$u \equiv_\mu t \;\text{ iff }\; \mu(u), \mu(t) \text{ are both defined and } \mu(u) = \mu(t)$$

We sometimes call such relations *synonymies*, noting that this is a technical notion, relative to the chosen notion of meaning.

[14] More correctly, we should write the string value of $\alpha(\textit{few}, \textit{cat})$ as $\textit{few}^\frown\text{␣}^\frown\textit{cats}$, where '␣' denotes word space and '$^\frown$' concatenation, but the simplified notation used here is easier to read.

2.2 Compositionality

For a given grammar $\mathbf{E}$ and semantics μ, we render (PoC-F) as follows:

Funct(μ) For every rule $\alpha \in \Sigma$ there is a meaning operation r_α such that if $\overline{\alpha}(u_1, \ldots, u_n)$ has meaning (belongs to the domain of μ), then $\mu(\overline{\alpha}(u_1, \ldots, u_n)) = r_\alpha(\mu(u_1), \ldots, \mu(u_n))$.

Here is the formal version of (PoC-S):

Subst($\equiv_\mu$) If $s[u_1, \ldots, u_n]$ and $s[t_1, \ldots, t_n]$ are both meaningful terms, and if $u_i \equiv_\mu t_i$ for $1 \leq i \leq n$, then $s[u_1, \ldots, u_n] \equiv_\mu s[t_1, \ldots, t_n]$.

The notation $s[u_1, \ldots, u_n]$ indicates that the term s contains — not necessarily immediate — *disjoint* occurrences of subterms among $u_1, \ldots, u_n$, and $s[t_1, \ldots, t_n]$ results from replacing each u_i by t_i.

Note that the formulation Funct(μ) presupposes the *Domain Principle* (DP): Subterms of meaningsful terms are meaningful. The following is well-known (e.g. Hodges [15]):

Fact 1
Under DP, Funct(μ) is equivalent to Subst($\equiv_\mu$).

In some cases, DP seems too strong. For example, a semantics for first-order logic assigning meanings to closed but not to open formulas doesn't satisfy DP. Likewise, a semantics for quotation that allows quotation of meaningless strings, as in (1c), may appear to violate DP. We will see, however (Sections 5 and 7), that when linguistic context is taken into account, DP is a less demanding requirement, and indeed our account handles quotation of arbitrary strings while satisfying a generalized version of DP.

Funct(μ) and Subst($\equiv_\mu$), like their informal counterparts (PoC-F) and (PoC-S), express the core meaning of compositionality. For compositionality to have practical import, one also needs to require that the meaning operations r_α are *computable* in some suitable sense. There are several ways in which basic compositionality can be strengthened. But note that (PoC-F) — by far the most common informal version of the principle in the literature — only says that the meaning of a complex phrase is *determined* by the meanings of its constituents (and the mode of composition); it says nothing about *how* it is determined. And the equivalent (PoC-S) doesn't even mention functionality or determination. As a consequence, we have (under DP):

(5) If μ is one-one, i.e. if no two distinct grammatical terms have the same meaning, then μ is compositional.

This is because under the assumption, $\equiv_\mu$ is the identity relation, so Subst($\equiv_\mu$) holds trivially.

Although Funct(μ) and Subst($\equiv_\mu$) without extra conditions are weak, they are not trivial: there are grammars and semantics for which they fail (e.g. Potts's semantics for quotation mentioned in footnote 8.) It is another matter that any semantics μ for $\mathbf{E}$ is *recoverable* from some compositional semantics μ'; for a simple example, let

(6) $\mu'(t) = \langle \mu(t), t \rangle$ (when defined)

μ' is one-one, so $\mathrm{Subst}(\equiv_{\mu'})$ holds, but this in no way makes $\mathrm{Subst}(\equiv_{\mu})$ trivial.[15]

3 Extra-linguistic Context Dependence

Context-dependence is ubiquitous in natural languages, and there is a current debate on how much of these phenomena are amenable to a systematic or compositional treatment. The simplest case is that of *basic indexicals.* A sentence φ containing such indexicals does not express a complete proposition, something that is true or false, but *utterances* or *occurrences* of φ do. The *context* of the utterance or occurrence is needed to fix the reference of the indexicals. Thus, we can distinguish the linguistic meaning of the sentence (type), and the content or proposition that various uses of it express. But what does it mean to say that this kind of linguistic meaning is or is not compositional? Without commitment to any view of what contexts are, the abstract situation is as follows.

In addition to our set $GT_{\mathbf{E}}$ of terms corresponding to structured expressions, and our set X of semantic values, there is a set C of contexts.[16] We can then think of the assignment of values in two ways:

(7) a. $\nu\colon GT_{\mathbf{E}} \longrightarrow [C \longrightarrow X]$
b. $\mu\colon GT_{\mathbf{E}} \times C \longrightarrow X$

Here $f : A \longrightarrow B$ means that f is a (partial) function from the set A to the set B, $[A \longrightarrow B]$ is the set of such functions, and $A \times B$ is the set of pairs (a, b) such that $a \in A$ and $b \in B$. So ν is a meaning assignment of the kind treated above: it assigns values directly to terms, although these values are now themselves functions. It corresponds to what Kaplan called *character.* μ, on the other hand, assigns values to 'expressions-in-context'. Is this more than a notational difference, more than two ways to think of character?

It is trivial to *define* μ in terms of ν, and vice versa, by the equation

$$\mu(t, c) = \nu(t)(c)$$

In mathematical parlance, ν is the *currying* of μ, and μ is the *uncurrying* of ν. Nevertheless, compositionality for the two functions is not the same. Indeed, the notion of compositionality from the preceding section applies only to ν. $\mathrm{Funct}(\nu)$

[15] See Westerståhl [41], and for further discussion of the alleged triviality of compositionality, Pagin and Westerståhl [24].

[16] Note that X may in turn consist of functions. For example, in standard possible-world semantics, X contains *intensions*, which are functions from possible worlds to ordinary extensions. In relativist variants, these functions have further arguments, such as times, locations, assessors, standards, etc. We then get analogous issues for the compositionality of intension, and how such compositionality is related to that discussed below. These questions are treated in Westerståhl [44], but they will not be important here.

is well-defined, but for μ we must define compositionality for semantics that take an extra context argument.

There is an obvious way to do this: think of the previous condition as *parametric* in the context. Just replace $\mu(t)$ with $\mu(t, c)$ everywhere. We obtain:[17]

C-Funct(μ) For every rule $\alpha \in \Sigma$ there is a meaning operation r_α such that for every context c, if $\overline{\alpha}(u_1, \ldots, u_n)$ and $u_1, \ldots, u_n$ have meaning in c, then $\mu(\overline{\alpha}(u_1, \ldots, u_n), c) = r_\alpha(\mu(u_1, c), \ldots, \mu(u_n, c))$.

Now, one might think that this is simply equivalent to Funct(ν). But it isn't. *First*, it is easy to give examples where Funct(ν) holds but C-Funct(μ) fails.. *Second*, there is no corresponding substitution version of C-Funct(μ). There is one for *immediate* subterms: C-Funct(μ) is equivalent to

(8) If $\overline{\alpha}(u_1, \ldots, u_n)$ and $\overline{\alpha}(t_1, \ldots, t_n)$ are both meaningful, and if $\mu(u_i, c_1) = \mu(t_i, c_2)$ for $1 \leq i \leq n$, then $\mu(\overline{\alpha}(u_1, \ldots, u_n), c_1) = \mu(\overline{\alpha}(t_1, \ldots, t_n), c_2)$.

But this does not extend to subterms that are more deeply embedded: that would require the subterms of $s[u_1, \ldots, u_n]$ that are *not* replaced to have the same meaning in c_1 and c_2, and there is no guarantee for that.

Third, there is a way in which C-Funct(μ) can fail *merely by changing the context*, without replacing any expressions at all, namely, if each of $u_1, \ldots, u_n$ has the same meaning in c_1 and c_2 but $\overline{\alpha}(u_1, \ldots, u_n)$ doesn't. This is called *context shift failure* in Pagin [19], where possible examples are given. This kind of failure of compositionality is not available for ν.

Fourth, there is a weaker variant of C-Funct(μ), which also seems quite natural:

C-Funct(μ)$_w$ For every $\alpha \in \Sigma$ there is a meaning operation r_α such that for every context c, if $\overline{\alpha}(u_1, \ldots, u_n)$ and $u_1, \ldots, u_n$ have meaning in c, then $\mu(\overline{\alpha}(u_1, \ldots, u_n), c) = r_\alpha(\mu(u_1, c), \ldots, \mu(u_n, c), c)$.

The only difference here is that the meaning operation r_α takes c as an extra argument. Again, this might seem insignificant, but it isn't. Context shift failure cannot occur for this version. Various semantic theories for which C-Funct(μ)$_w$ holds but C-Funct(μ) fails have been proposed. Furthermore, this time *there is* a simple substitution version, for arbitrary subterms:

C-Subst($\equiv_\mu$)$_w$ If $s[u_1, \ldots, u_n]$ and $s[t_1, \ldots, t_n]$ are both meaningful terms, and if $\mu(u_i, c) = \mu(t_i, c)$ for $1 \leq i \leq n$ (i.e. both sides are defined and

[17] We make the simplifying assumption that the non-linguistic context c is the same for all parts of a complex term. This is sometimes violated; for example, when the sergeant, pointing in turn at three of the recruits, shouts

(i) You, you, and you are volunteers!

We shall not pursue this here. When we come to linguistic context, however, it is a crucial feature that the context of the immediate subterms of a term can be different from the context of the term itself in a way that is semantically significant (see Section 4).

have the same value), then $\mu(s[u_1, \ldots, u_n], c) = \mu(s[t_1, \ldots, t_n], c)$.

Moreover, we shall see that it is this version of contextual compositionality that lends itself to adaptation for linguistic contexts.

The following result explains the logical relations between the notions of compositionality discussed so far, with μ and ν as in (7) (see Westerståhl [44] for a proof and discussion):

Fact 2
C-Funct(μ) implies C-Funct$(\mu)_w$ (equivalently, C-Subst$(\equiv_\mu)_w$), which in turn implies Funct(ν) (equivalently, Subst($\equiv_\nu$)), but none of these implications can be reversed.

4 Linguistic Context

Often the interpretation of a linguistic expression on an occasion depends on other expressions that are used (or just have been used) on the same occasion. This can happen in two main ways: by *linguistic* context-dependence proper, and also by way of the influence of linguistic expressions (used on the occasion) on the *extra-linguistic* context. We may call the latter 'linguistic contextual impact'. Consider

(9) That is a big mouse.

Here it may be, or it may fail to be, the case that a standard or comparison class for size is given in the extra-linguistic context. If it is, e.g. the comparison class of animals in some particular laboratory, and it applies in the case of the utterance of (9), then *big* applies intersectively with respect to *mouse*. But the occurrence of *mouse* may also set a *new standard* of size in the context, whether or not there was a standard in the immediately preceding context. In that case the complex noun *big mouse* will apply to anything that (within some relevant domain) is big for a mouse. 'big' still applies intersectively, but the standards have changed. This phenomenon is worthy of further study (see e.g. Recanati [33] for a similar view, and Kennedy [17] for more on gradable adjectives), but is mentioned here only as a contrast to the present topic, that of linguistic context dependence proper.

4.1 Sentential contexts

What is a linguistic context? Here we restrict attention to the context of an expression within a larger expression, notably a sentence. The basic idea is simple enough: take a well-formed complex term $s[u]$ and knock out the constituent u. What remains, $s[\ldots]$, is the *linguistic context* of that occurrence of u in $s[u]$, the environment of the argument place. This simple idea needs to be refined, and we shall see how.

Standard examples of linguistic contexts that are relevant to semantics include *belief contexts*. In a belief sentence like

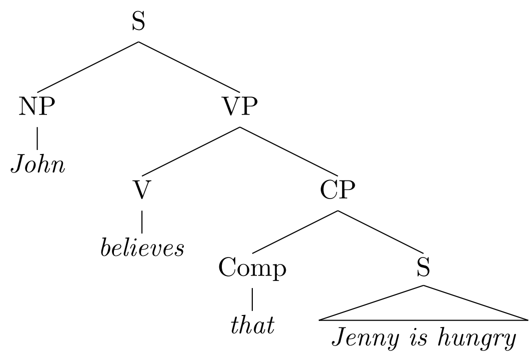

Figure 1: A phrase structure tree for (10).

(10) John believes that Jenny is hungry

the embedded sentence

(11) Jenny is hungry

occurs in the belief context

(12) John believes that ...

According to Frege [8], that (11) occurs in a belief context has an effect on its meaning in that context: it has *indirect reference* (*ungerade Bedeutung*).

This view gives us a first reason for revising the simple idea: the occurrence of the subject *John* does not participate in defining the belief context. What matters for being in a belief context is that the expression occurs in the second argument place of the verb *believes*. What occurs in the first argument place, *John*, matters to the truth conditions of the sentence as a whole, but has no effect on the semantics for the embedded sentence itself.

For definiteness, let the construction of belief VPs be given by two one-place rules: α_{bel}, where the argument is the sentence term for the *that*-clause (Complementizer Phrase) argument, and α_{cp}, which takes the embedded sentence as argument and gives the *that*-clause as value (various other formats are possible). Figure 1 gives a simplified tree representation.

The belief context is the context of the position of the dominated S node in the tree, where the embedded sentence goes. In the term notation, this is $\overline{\alpha}_{\mathrm{bel}}(\overline{\alpha}_{\mathrm{cp}}(\ldots))$. The example illustrates that we should not rule out that what matters to determining the linguistic context can extend beyond the immediate syntactic rule. In this case, two rules are required.

There is a further important complication. On a standard conception, if a complex expression occurs in a belief context, then the constituents of that expression also occur in the belief context. For instance, the name *Jenny* occurs in the belief context just like the entire sentence (11). So on this conception, being in a particular linguistic context is like being in the scope of an operator.

But this is not quite right, or at least highly misleading, as can be seen by considering iterated belief contexts. In

(13) Emma believes that John believes that Jenny is hungry

the name *John* occurs in a simple belief context, but *Jenny* occurs in an iterated, and more precisely second degree, belief context. This can matter to semantics. On a common interpretation of Frege, expressions in second degree belief contexts have a doubly indirect reference and a doubly indirect sense.[18] This shows that in general we need to take into account several constituent levels in the syntactic term, and for each level the relevant argument place. Therefore it is appropriate to let linguistic context be defined in terms of operators and argument positions down to the target argument place. We now give a precise version of this idea.

4.2 A formal notion of linguistic context

The notion of linguistic context applies to *occurrences* of (sub)terms. We let o (with subscripts) range over occurrences.[19] For any term s, and any occurrence o of a subterm of s, we define inductively *the* (linguistic) *context of o in s*, $cxt(o, s)$, as a finite sequence $\xi = \langle(\alpha_1, i_1), \ldots, (\alpha_n, i_n)\rangle$ of pairs consisting of a grammar rule and a natural number. ($^\frown$ is now concatenation of sequences.)

($\mathrm{L}_{\mathrm{cxt}}$) (i) $cxt(s, s) = \langle\rangle$ (the *null context*).

(ii) If $cxt(\overline{\alpha}(o_1, \ldots, o_k), s) = \xi$ then $cxt(o_i, s) = \xi^\frown\langle(\alpha, i)\rangle$, $1 \leq i \leq k$.

(iii) $CXT_{\mathbf{E}} = range(cxt)$ (the set of *contexts* in $\mathbf{E}$).

So the context of o in s encodes the *path* in the tree corresponding to the term s that starts with the node s and ends with the node o. Clearly, every subterm occurrence in s is in a unique context in s, and distinct subterm occurrences are in distinct contexts in s. The same context ξ can be a context in many different terms,[20] but for any grammatical term s, ξ determines at most one subterm occurrence. If $cxt(o, s) = \xi$, and we replace o by another term, we obtain a term s' (which may or may not be grammatical), differing from s in having the new term in the *same* context as o, i.e. ξ.

[18] There is some textual evidence that this, at least at some point, was Frege's own view. The following is a quote from a letter to Russell 28 December 1902 (thanks here to Sara Packalén):

> Man kann sagen, dass wir im doppelt Unterstrichenen die ungerade Bedeutung zweiten Grades haben, in dem einmal Unterstrichenen dagegen die ungerade Bedeutung ersten Grades ([9], p. 236).

Here Frege explicitly refers to "an indirect reference of second degree".

[19] We use an informal notion of term occurrence here. It could be redefined later in terms of the function *cxt* introduced below.

[20] For example, $\xi = \langle(\alpha, 1), (\beta, 2)\rangle$ is a context in each of the terms $\overline{\alpha}(\overline{\beta}(t, u), \overline{\gamma}(t))$, $\overline{\alpha}(\overline{\beta}(t, u), t')$, and $\overline{\alpha}(\overline{\beta}(t, \overline{\delta}(u)), \overline{\gamma}(t))$. The subterm occurring at context ξ is u in the first two cases, and $\overline{\delta}(u)$ in the third case.

We say that ξ is a *possible context for* t if there is a grammatical term s which has a subterm occurrence o of t such that $ctx(o, s) = \xi$. Thus, $CXT_{\mathbf{E}}$ is the set of possible contexts of grammatical terms in $\mathbf{E}$. If ξ is a possible context for some term $\overline{\alpha}(u_1, \dots, u_n)$, then $\xi^\frown\langle(\alpha, i)\rangle$ is a possible context for u_i. In this case, we say that ξ *allows extension by* α.

$CXT_{\mathbf{E}}$ may be large — usually infinite — but when studying the effect of linguistic context on meaning we may be interested only in a small number of *types* of contexts. For example, Frege in the quote earlier in effect isolates three types of context: an 'ordinary' (null) context type (when talking about the references of words), a quotation context type (talking about the words themselves), and an 'intensional' context type (talking about their senses). This motivates the following definition:

(L_{ctype}) A *context typing* is a partition C of $CXT_{\mathbf{E}}$ satisfying the following requirement, where $[\xi]$ is the equivalence class of ξ:

(i) If $[\xi] = [\xi']$, and ξ and ξ' both allow extension by an n-ary grammar rule α, then, $[\xi^\frown\langle(\alpha, i)\rangle] = [\xi'^\frown\langle(\alpha, i)\rangle]$ for $1 \leq i \leq n$.

The elements of C are called *context types*.

Requirement (i) says that if two contexts are of the same type, and can be extended in the same way, then the corresponding extensions also have the same type. This means that the context type of a subterm occurrence $\overline{\alpha}(o_1, \dots, o_k)$ in s determines the context types of $o_1, \dots, o_k$ in a way which is *independent of* s, according to the partial function Φ_C, which takes a n-ary grammar rule α, a natural number i (an argument place of α), and a context type c to a context type $\Phi_C(\alpha, i, c)$, and is defined as follows:

(14) $$\Phi_C(\alpha, i, [\xi]) = \begin{cases} [\xi^\frown\langle(\alpha, i)\rangle] & 1 \leq i \leq n, \text{ if } \xi \text{ allows extension by } \alpha \\ \text{undefined} & \text{otherwise} \end{cases}$$

Thus, $\Phi_C(\alpha, i, c)$ is defined if and only if some $\xi \in c$ is a possible context for some term of the form $\overline{\alpha}(u_1, \dots, u_n)$.

Finally, it is straightforward to see that using Φ_C, one can define by induction a function Θ_C taking a context type c and a context ξ as arguments, such that:

(15) If s occurs in context type c, and ξ is the context of an occurrence of t in s, then the context type of that occurrence is $\Theta_C(c, \xi)$.

5 General Compositionality

The first main idea for an extension of a semantic framework to handle linguistic context dependence is that when a semantic function μ is applied to a term s, the value depends on the relevant context type. The second main idea is that the context type of an immediate subterm of a term s may be different from the context type of s itself. As we shall see, there are two equivalent styles of implementing this: either with a semantic function that takes a context type as a second argument, or instead with a set of ordinary semantic functions, one for each context type.

5.1 A formulation in terms of formal contexts

The first style builds on the formal notion of context just introduced. Assume a context typing C of $CXT_{\mathbf{E}}$ according to ($\mathrm{L}_{\mathrm{ctype}}$) is given. Just as for extra-linguistic context (Section 3), a semantics μ now takes an additional argument: it is a partial function from $GT_{\mathbf{E}} \times C$.

For simplicity, we shall also require that a Generalized Domain Principle, holds for μ:

(GDP_{μ}) If $\mu(\overline{\alpha}(u_1, \ldots, u_n), c)$ is defined, then $\mu(u_i, \Phi_C(\alpha, i, c))$ is defined for $1 \leq i \leq n$.

In Section 2.2 we noted that DP is sometimes too demanding. When meaning is allowed to depend on linguistic context, however, the case for a (generalized) domain principle is stronger. Rather than not assigning any meaning at all to some subterms of a complex term, we can assign them a *different* meaning when they are in that context (type). In Section 7 we will see an example how this works out, in a way consistent with GDP_{μ}.

Now, the compositionality requirement is similar to C-Funct$(\mu)_w$, the difference being that the context types of the arguments of complex terms may change; indeed the function Φ_C assigns those types. The linguistic contexts of the arguments are different from the context of the term itself, and that difference may be semantically relevant. For example, the context type of the complex term may be a quotation type.

LC-Funct(μ, C) For every $\alpha \in \Sigma$ there is an operation r_α such that for every $c \in C$, if $\mu(\overline{\alpha}(u_1, \ldots, u_n), c)$ is defined, then

(i) $\mu(\overline{\alpha}(u_1, \ldots, u_n), c) = r_\alpha(\mu(u_1, c_1), \ldots, \mu(u_n, c_n), c)$,
where $c_i = \Phi_C(\alpha, i, c)$, $1 \leq i \leq n$.

5.2 A formulation with sets of semantic functions

In the second style, we keep the standard notion of a semantics as an assignment of meanings to grammatical terms, but we use several such semantic functions. Intuitively, there is one function for each context type, but this formulation doesn't require a formal notion of context. We simply suppose that a *set* S of (ordinary) semantic functions is given, together with a *selection function* Ψ; a partial function from triples of a grammar rule, a natural number, and an element of S, to S. The Generalized Domain Principle now takes the form:

(GDP_S) If $\mu \in S$ and $\mu(\overline{\alpha}(u_1, \ldots, u_n))$ is defined, then $\Psi(\alpha, i, \mu)(u_i)$ is defined, $1 \leq i \leq n$.

We also assume that there is a designated (null) function $\mu_d \in S$, the default semantics corresponding to the null context type. μ_d, however, plays no explicit role in the compositionality condition, which, for S satisfying GDP_S, becomes:

LC-Funct(S, Ψ) For every $\alpha \in \Sigma$ and every $\mu \in S$ there is an operation $r_{\alpha,\mu}$ such that if $\mu(\overline{\alpha}(u_1, \ldots, u_n))$ is defined, then

(ii) $\mu(\overline{\alpha}(u_1, \ldots, u_n)) = r_{\alpha,\mu}(\mu_1(u_1), \ldots, \mu_n(u_n))$,
where $\mu_i = \Psi(\alpha, i, \mu)$, $1 \leq i \leq n$.

5.3 Some facts

The two generalized versions of contextual compositionality have standard compositionality as a special case, in the sense of the next, easily verified, fact.

Fact 3
If μ is a partial function from $GT_{\mathbf{E}}$ such that DP holds, then Funct(μ) is equivalent to each the following:

(a) *LC-Funct$(\{\mu\}, \Psi_0)$, where $\Psi_0(\alpha, i, \mu) = \mu$.*

(b) *LC-Funct$(\mu^*, \{CXT_{\mathbf{E}}\})$, relative to the trivial context typing of $CXT_{\mathbf{E}}$ that has $CXT_{\mathbf{E}}$ itself as the only context type, and the trivial context-sensitive version μ^* of μ given by $\mu^*(t, CXT_{\mathbf{E}}) = \mu(t)$ (if $\mu(t)$ is defined).*

Next, we can show that LC-Funct(μ, C) and LC-Funct(S, Ψ) are equivalent in the following sense:

Proposition 4

(a) *If LC-Funct(μ, C) holds, define, for $c \in C$, $\mu^c(t) = \mu(t, c)$ (if defined), and let $\Psi(\alpha, i, \mu^c) = \mu^{\Phi_C(\alpha,i,c)}$ (if $\Phi_C(\alpha, i, c)$ is defined), $1 \leq i \leq n$. Then GDP_S and LC-Funct(S, Ψ) hold for $S = \{\mu^c : c \in C\}$.*

(b) *Conversely, suppose LC-Funct(S, Ψ) holds. Define a partial mapping F from $CXT_{\mathbf{E}}$ to S inductively by:*

$$\begin{cases} F(\langle\rangle) & = & \mu_d \text{ (the null function in } S\text{)} \\ F(\xi^\frown\langle(\alpha, i)\rangle) & = & \Psi(\alpha, i, F(\xi)), \text{ if for some term } \overline{\alpha}(u_1, \ldots, u_n), \\ & & F(\xi)(\overline{\alpha}(u_1, \ldots, u_n)) \text{ is defined (otherwise undefined)} \end{cases}$$

The functions in S themselves are taken to be the context types, or more exactly, their inverse images under F: let

$$[\xi] = \{\xi' : F(\xi') \text{ is defined iff } F(\xi) \text{ is defined, and then } F(\xi') = F(\xi)\}$$

and let $C = \{[\xi] : \xi \in CXT_{\mathbf{E}}\}$. Define a semantics ν, taking a term t and a context type $[\xi]$ as arguments, by

$$\nu(t, [\xi]) = F(\xi)(t), \text{ if both } F(\xi) \text{ and } F(\xi)(t) \text{ are defined}$$

(otherwise undefined). Then GDP_ν is satisfied, C is a context typing of $CXT_{\mathbf{E}}$, and LC-Funct(ν, C) holds.

The proof of this result is given in Appendix 2. In view of Proposition 4, we shall refer to LC-Funct(μ, C) and LC-Funct(S, Ψ) indiscriminately as *general compositionality*, or *g-compositionality*.

5.4 Synonymies

With the general versions of functional compositionality, we obtain, for each choice of (S, Ψ), a variety of synonymy relations. For every $\mu \in S$ there is a corresponding synonymy relation $\equiv_\mu$. This gives synonymy *relative* to members of S. But we can also distinguish three absolute notions. First, define

$$u_i \equiv_{S_{\text{des}}} u_j \quad \text{iff} \quad u_i \equiv_{\mu_d} u_j \quad (\mu_d \text{ is the designated function in } S)$$

Call this *designated synonymy*. Another natural candidate is *total synonymy*:

$$u_i \equiv_{S_{\text{tot}}} u_j \quad \text{iff for every } \mu \in S, \quad u_i \equiv_\mu u_j$$

As is intuitively clear, and as will follow from the semantics in Section 7, if there is a quotation context type in $\mathbf{E}$, then there will be no non-trivial *total* synonymy pairs: a term s is totally synonymous only with itself. Because of this, we may be interested in a concept of *total synonymy except for quotation*, i.e. synonymy in all pure *use-contexts* (as opposed to quotation contexts, where expressions are both used and mentioned). Call this strong notion *use synonymy* (which for languages without quotation coincides with total synonymy); like the others, it is a partial equivalence relation on the set of grammatical terms.

Finally, let us consider the substitution version of general compositionality. We use the version LC-Funct(μ, C). First, modify the notation $s[u_1, \ldots, u_n]$ as follows: let

$$s = s[(u_1, \xi_1), \ldots, (u_n, \xi_n)]$$

indicate that, for $1 \leq i \leq n$, at context ξ_i in s there is an occurrence of u_i, where ξ_i and ξ_j are pairwise *disjoint* for $i \neq j$, i.e. neither is an extension of the other. Then

$$t = s[(t_1, \xi_1), \ldots, (t_n, \xi_n)]$$

is the unique term obtained from s by replacing the occurrence of u_i at ξ_i by t_i. The disjointness assumption entails that the result is independent of whether the replacements are done simultaneously or sequentially.

We also need recourse to the function Θ_C from (15) in Section 4.2. Then we can state:

LC-Subst(μ, C) If $s = s[(u_1, \xi_1), \ldots, (u_n, \xi_n)]$ and $t = s[(t_1, \xi_1), \ldots, (t_n, \xi_n)]$ are both μ-meaningful in the context type $c \in C$, and if, for

$1 \leq i \leq n$, $\mu(u_i, c_i) = \mu(t_i, c_i)$, where $c_i = \Theta_C(c, \xi_i)$, then $\mu(s, c) = \mu(t, c)$.

This says that, for any c, the semantic values of two terms are the same in context type c if one results from the other by means of substitution of subterms that are pairwise μ-equivalent in the type of context where they occur.

Fact 5
If (GDP$_\mu$) holds, LC-Funct(μ, C) is equivalent to LC-Subst(μ, C).

The left-to-right direction of Fact 5 is proved by means of induction over term complexity. The right-to-left direction follows immediately from the special case when the substitution terms are the immediate subterms.

6 Compositional Generalization

LC-Funct(μ, C) holds for a pair (μ, C) of a binary function μ and a context typing C. Similarly, LC-Funct(S, Ψ) holds for a pair (S, Ψ) of a set S of unary functions and a selection function Ψ (for LC-Funct(μ, C) the function Φ_C is determined by the typing itself according to (14)).

Typically, however, we are initially given a grammar **E** and an initial single unary semantic function μ for **E**. In some cases, for instance with a straightforward account of quotation, μ is non-compositional, as we have seen. We are then interested in whether there is a proper corresponding μ-related pair (μ', C) or (S, Ψ) that satisfies LC-Funct(μ, C) or LC-Funct(S, Ψ), respectively. We use the following terminology:

(16) (μ', C) *generalizes* μ iff for all t, $\mu(t) = \mu'(t, [\langle\rangle])$. Similarly, (S, Ψ) *generalizes* μ iff μ is the designated function in S.

Now we are asking when μ is in a relevant sense *compositionally generalizable*. At first blush, the relevant sense seems easy to spell out. In the LC-Funct(S, Ψ) format, the first suggestion would be:

(17) μ is compositionally generalizable iff LC-Funct(S, Ψ) holds for some pair (S, Ψ) generalizing μ.

But this requirement is *empty*: in the sense of (17), *any* semantic function μ is generalizable. To see this, define the function ν by $\nu(t) = \langle \mu(t), t\rangle$. Let $S = \{\mu, \nu\}$, and $\Psi(\alpha, i, \mu) = \Psi(\alpha, i, \nu) = \nu$, for any α and i. Then LC-Funct(S, Ψ) holds.[21]

[21] *Proof.* We must find, for each α, meaning operations $r_{\alpha,\mu}$ and $r_{\alpha,\nu}$ that satisfy the compositional recursion equations. Let

$$r_{\alpha,\mu}(m_1, \ldots, m_n) = \mu(\overline{\alpha}(\pi_2(m_1), \ldots, \pi_2(m_n)))$$

$$r_{\alpha,\nu}(m_1, \ldots, m_n) = \langle \mu(\overline{\alpha}(\pi_2(m_1), \ldots, \pi_2(m_n))), \overline{\alpha}(\pi_2(m_1), \ldots, \pi_2(m_n))\rangle$$

where π_2 is a right projection function: $\pi_2(\langle x, y\rangle) = y$. Then, with ν, S, and Ψ as above:

In the above definition of ν, general compositionality is achieved by way of *hyperdistinctness*: if $\nu(t) = \nu(u)$, then $t = u$. This also ensures ordinary compositionality, in a degenerate way (cf. (5) and (6) in Section 2.2). It is reasonable to block this move by requiring that a semantic function for a particular context type *respect* the semantic equivalences in that context for the original function μ. Staying in the LC-Funct(S, Ψ) format, recall from Proposition 4 that each function in S has the form μ^c, for some context type c in the corresponding typing. The requirement can then be stated as follows (with the notation from Section 5.4):

(MAX) If ξ is a context such that for all terms s containing ξ, $s[(u, \xi)] \equiv_\mu s[(t, \xi)]$, then $u \equiv_{\mu^{[\xi]}} t$.

Informally, if u and t make the same contribution to the meaning of complex expressions in which they occur, *in a certain linguistic context*, then their meaning *in* the type of that context is the same. This is similar to Hodges' notion of a *Fregean extension* (or full abstraction; see Hodges [15]), which can be seen as expressing a version of Frege's Context Principle, but this time for meanings that are sensitive to linguistic context.

Suppose the trivial generalization of μ into $S = \{\mu, \nu\}$ above satisfies MAX. Since for all distinct terms t, u, $\nu(t) \neq \nu(u)$, this means that in the ν context type, to which every embedded (non-null) context belongs according to the associated Ψ, distinct terms are semantically non-equivalent. By MAX, it follows that for any distinct terms t, u, and for any embedded context ξ where t and u can occur, there is a term s such that

$$s[(u, \xi)] \not\equiv_\mu s[(t, \xi)]$$

As a consequence, if for any two terms t and u, $\mu(t) = \mu(u)$, then there is substitution failure *in every embedded linguistic context* (although not for every term that contains it). μ is then non-compositional, not just with respect to *some* operator α, but with respect to *every* syntactic operator. Although such semantic functions are possible, we seem not to find them in natural language. Generating a language L' by adding quotation to a language L does make L' non-compositional if there are non-trivial synonymies in L, but only because of the quotation contexts. It does not make all the operators in the L fragment non-compositional as well. So, for an ordinary semantics μ, generalizing it into $S = \{\mu, \nu\}$ as above does not satisfy MAX. MAX seems to be a reasonable requirement on generalization.

However, even under MAX, any unary semantic function μ turns out to be generalizable to *some* (S, Ψ) satisfying LC-Funct, provided there is no limitation on the number of context types:

(i) For any unary function μ, LC-Funct(S, Ψ) holds.

□

Fact 6
For any unary semantics μ, there is (S, Ψ) generalizing μ such both MAX and LC-Funct(S, Ψ) hold.

The key to the proof, which is given in Appendix 2, is to use many context types; in fact, we let each context be its own type. Even though that is hardly a situation one encounters in practice, there is thus reason to introduce a second requirement on compositional generalizability.

The type of a context ξ may in principle depend on all the operators in ξ (recall that the formal notion of a context is a sequence starting with the null context, followed by operator-index pairs), in the order in which they occur, or on any particular subset of the set of operators in ξ, again in their order in ξ. In the simplest case, however, only one operator matters. We can spell this out as follows.

Call an operator-index pair (α, i) is a *switcher* (relative to a context typing C) iff there are at least two context types, and for any two contexts ξ, ξ' in which a term $\overline{\alpha}(t_1, \ldots, t_n)$ can occur,

$$[\xi^\frown\langle(\alpha, i)\rangle] = [\xi'^\frown\langle(\alpha, i)\rangle]$$

This means that there is a context type c such that whatever the context type of (a grammatical occurrence of) $\overline{\alpha}(t_1, \ldots, t_n)$, the context type of the corresponding occurrence of t_i is c. Let us also say that (α, i) is a *keeper* iff there are at least two context types, and for any admissible context ξ,

$$[\xi^\frown\langle(\alpha, i)\rangle] = [\xi]$$

This means that (α, i) never changes context type. We can now define a *first-grade* g-compositional semantics (S, Ψ) to be a g-compositional semantics where every operator-index pair is either a switcher or a keeper. In such a semantics, the context type of a context ξ is determined by the *last* switcher pair in the sequence that constitutes ξ, or else is the default type if there is no switcher in the sequence. If all pairs are switcher pairs, then the type of a context $\xi'^\frown\langle(\alpha, i)\rangle$ is always determined by (α, i), irrespective of ξ'.

We can analogously define a *second-grade* g-compositional semantics by defining *switcher pairs* of operator-index pairs. In this case, the type of a context ξ is determined by the last switcher *pair* of operator-index pairs in the sequence that constitutes ξ. Similarly, switcher triples determine context type in a *third-grade* semantics. And so on. A Fregean semantics for belief contexts, where each iteration of the belief operator switches to a new type in an infinite hierarchy of indirect context types, is an ω-grade semantics.

Clearly, along this parameter, the simplest g-compositional semantics are the first-grade ones. The number of context types for such semantics is at most equal to the number of operator-index pairs. A particular operator-index pair is responsible for the context type. The number of rules can be kept low, and no more than one operator-index pair needs to be kept in memory at any single recursion step of processing the semantics. This indicates that the following terminology is appropriate:[22]

(GEN) μ is *compositionally generalizable* iff there is a first-grade pair (S, Ψ) generalizing μ such that LC-Funct(S, Ψ) and MAX hold.

GEN is *not* an empty requirement. A proof of the following is outlined in Appendix 2:

Fact 7
For each n there is a grammar $\mathbf{E}$ and a semantics μ for $\mathbf{E}$ which is not generalizable by any nth-grade pair (S, Ψ) satisfying LC-Funct and MAX.

The g-compositional semantics for pure quotation given in the next section satisfies GEN. The one-place quotation operator always triggers a quotation context type. All other operator-index pairs are keepers, preserving either the quotation context type or the default type. In particular, there is no switcher that takes the semantics *out* of a pure quotation context, which seems in accordance with ordinary intuitions (but see note 27).[23]

7 A G-compositional and Straightforward Account of Pure Quotation

Our final task is to show that a straightforward semantics for pure quotation (Section 1) is compositionally generalizable in the sense of GEN. We formulate this as a *compositional extension* result: *given* any semantics μ which is compositional in the ordinary sense, there is a natural way to extend it to a semantics

[22] In Pagin and Westerståhl [23], we distinguish *1st level compositionality*, where the meaning of a complex term depends on the meanings of its immediate subterms, from *2nd level compositionality*, where also the meanings of the immediate subterms of the immediate subterms may be required, and so on. This is somewhat analogous to the grade distinction here, and standard compositionality is 1st level, which is yet another reason for choosing the formulation in GEN. Also, many semantics where linguistic context is allowed to play a role in fact satisfy GEN; see the next section, and note 23. The ω-grade Fregean semantics, on the other hand, requires infinitely many context types, which threatens to trivialize the compositionality requirement.

[23] General compositional semantics appear in earlier work by Kathrin Glüer and Peter Pagin. [11] presents a semantics accounting for the behavior of proper names in modal contexts. The idea is to keep proper names as non-rigid designators, with non-constant intensions, and achieve the rigidity-like effect by means of the interaction of the modal operator with the term. The modal operator then acts as a switcher, creating an *actualist* context where the name reference is the value of its intension in the actual world.

The resulting semantics is equivalent to a standard semantics with rigid designators w.r.t. truth, and almost equivalent w.r.t. logical consequence, as shown in [12]. In [11], an appendix considers the addition of a naive belief operator that switches from an actualist context type back to the default *possibilist* context type.

The idea has also been extended to handle *general terms* in modal contexts in [13]. In unpublished work ([21]), Pagin develops a more adequate g-compositional account of belief sentences. In all these cases, GEN is in fact satisfied, although this condition is not mentioned.

which handles pure quotation straightforwardly and is compositionally generalizable.[24] We use the formulation LC-Funct(S, Ψ), but Proposition 4 provides an immediate translation to the version with explicit context types.

Let, then, $\mathbf{E} = (E, A, \Sigma)$ be a given grammar and μ a semantics for $\mathbf{E}$ such that Funct(μ) holds. There is a string value function V from $GT_{\mathbf{E}}$ to E (Section 2.1). Now add a new syntactic operation κ that puts quote marks around any expression e, and interprets the result as referring to e:

(18) a. $\kappa(e) =$ 'e' (i.e. the string leftquote$^\frown e^\frown$rightquote)

b. $\mu'(\overline{\kappa}(t)) = V'(t)$

where μ' is μ extended to the new terms, and V' the corresponding string value function. Roughly, the claim is that the set $S = \{\mu', V'\}$ (with a suitable selection function) is g-compositional and satisfies GEN with respect to μ'.

However, to deal with sentences like those in (1), Section 1, we need to quote not only well-formed expressions but also, say, single letters of some alphabet, expressions in other languages or dialects, misspellings, even arbitrary concatenations of letters. A general theory of quotation must take care to handle such cases correctly. Since we are restricting attention here to pure quotation, we shall simply add a mechanism for quoting arbitrary finite strings of letters from some suitable alphabet L, which is assumed to contain the letters occurring in atoms of $\mathbf{E}$ as well as a space symbol and the quote marks. We then have $A \subseteq E \subseteq L^*$, where L^* is the set of all finite strings of letters. Some strings in L^* are well-formed, most strings aren't, but in principle we should be able to quote all of them.

One way to achieve this is to add all strings of the form 'u', when $u \in L^*$, as new atoms. This is the *proper name theory* of quotation, and we indicated in Section 1 why it is unsatisfactory: (a) we get infinitely many primitive expressions, and (b) quoting expressions have no structure. To remove (a), we generate all strings from letters and the concatenation operation. (b) is removed by using κ whenever we want to quote something (grammatical or not), thus keeping the account straightforward.[25]

In more detail, we extend the grammar $\mathbf{E}$ as follows. Let the new set of atoms be

(19) $A' = A \cup L$

(assuming $A \cap L = \emptyset$). The new rules are

(20) $\Sigma' = \{\alpha' : \alpha \in \Sigma\} \cup \{cc, \kappa\}$

---

[24] A compositional extension result has the form: if μ is a compositional semantics for $\mathbf{E}$, there is a (possibly unique) extension μ' of μ satisfying a certain property. Hodges [15] and Westerståhl [43] prove theorems concerning the extension from a partial to a total semantics. Westerståhl [42] gives extension results for adding idiomatic expressions to a language.

[25] Thus, the task of quoting arbitrary strings leads us to something like the *description theory* of quotation, and in this respect our account is similar to the one in Werning [40]. But the crucial difference is that we combine this with a standard syntactic quotation operator.

where cc is a binary concatenation operation generating, we assume, L^* from L, and κ is the function in (18a), taken to be *total*. Each α' extends α in some way that naturally incorporates the new expressions into the grammar. Lacking a detailed description of $\mathbf{E}$, we cannot specify exactly how this is done, but the idea should be clear. Using κ, we want to generate strings like *John likes 'Mary'* (meaning that he likes the name), or with iterated quotation as in

(21) 'John likes 'Mary'' is a sentence

but not, for example, *John 'likes' Mary* (we deal only with pure quotation), or *'John likes' Mary*. This could be done by treating all quoting expressions (expressions of the form $\kappa(e)$) as *noun phrases*, and adapting the old rules accordingly.

Finally E' is the closure of A' under the functions in Σ', and

$$\mathbf{E}' = (E', A', \Sigma')$$

is the extended grammar. As to the term algebra, we now have a primitive grammatical term

$$\bar{l}$$

for each l in L. Let the set of *string terms*, ST, be defined inductively by

(22)
$$\bar{l} \in ST \text{ for } l \in L$$
$$\overline{cc}(t, u) \in ST \text{ when } t, u \in ST$$

The new set of grammatical terms, $GT_{\mathbf{E}'}$, is the closure of $\{\overline{a} \colon a \in A'\}$ under the functions $\overline{cc}$, $\overline{\kappa}$, and $\overline{\alpha'}$ for $\alpha \in \Sigma$, where $\overline{cc}$ is only defined on ST, whereas $\overline{\kappa}$ is total (every term can be quoted). The new string value function V' is as usual:[26]

(23)
$$\begin{cases} V'(\overline{a}) & = & a, \text{ for } a \in A' \\ V'(\overline{\delta}(t_1, \ldots, t_n)) & = & \delta(V'(t_1), \ldots, V'(t_n)), \text{ for } \delta \in \Sigma' \end{cases}$$

It is now fairly clear how the given semantics μ can be extended to a semantics μ' suitable for $\mathbf{E}'$. We adjoin L^* to the given domain of interpretation M (assuming $M \cap L^* = \emptyset$), and adapt the given composition functions r_α for $\alpha \in \Sigma$ to functions $r_{\alpha'}$ that work for the new terms. The meaning of terms of the form $\overline{\kappa}(t)$ was given in (18b), but μ' is *undefined* for string terms.

As we have emphasized, the extended semantics μ' is *not compositional*: one cannot substitute synonymous terms in the scope of $\overline{\kappa}$ and expect meaning to be preserved. However, (S, Ψ) is g-compositional, where $S = \{\mu', V'\}$, and the selection function Ψ is defined, for $\delta \in \Sigma'$and $\nu \in S$, by

(24)
$$\Psi(\delta, i, \nu) = \begin{cases} V' & \text{if } \delta = \kappa \\ \nu & \text{otherwise} \end{cases}$$

[26] Strictly speaking, $\mathbf{E}'$, $GT_{\mathbf{E}'}$, and V' should be defined by simultaneous induction.

That is, μ' only switches to V' when something is quoted, and then every more deeply embedded subterm is also evaluated with V'.[27]

Next, LC-Funct(S, Ψ) requires us to define operations $r_{\delta,\nu}$ for each $\delta \in \Sigma'$ and $\nu \in S$. Specifically, we must satisfy

(25) $$V'(\overline{\delta}(t_1, \ldots, t_n)) = r_{\delta,V'}(V'(t_1), \ldots, V'(t_n))$$

so it follows from (23) that we should set

(26) $$r_{\delta,V'} = \delta$$

for all $\delta \in \Sigma'$. In particular, when $\delta = \kappa$ we then have

(27) $$r_{\kappa,V'}(V'(t)) = \kappa(V'(t)) = \text{`}V'(t)\text{'}$$

This will apply only in iterated quotation contexts (see the example below). As to $r_{\delta,\mu'}$, we set

(28) $$\begin{cases} r_{\alpha',\mu'}(m_1, \ldots, m_n) & = & r_{\alpha'}(m_1, \ldots, m_n), \text{ for } \alpha \in \Sigma \\ r_{\kappa,\mu'}(m) & = & m \end{cases}$$

(We don't need to define $r_{cc,\mu'}$, since $\mu'(\overline{cc}(t, u))$ is always undefined.)

In particular, we get

(29) $$\mu'(\overline{\kappa}(t)) = r_{\kappa,\mu'}(V'(t)) = V'(t)$$

in accordance with (18).

To see how S and Ψ work, consider sentence (21), and suppose, for simplicity of illustration, that this string is the value of the term

$$t = \overline{isa'}(\overline{\kappa}(\overline{\alpha'}(\overline{John}, \overline{\beta'}(\overline{like}, \overline{\kappa}(\overline{Mary})))), \overline{sentence})$$

(so sentences of the form 'NP is a N' are analyzed as isa(NP,N)). Its designated meaning is calculated, using (23) – (29), as shown in Figure 2. Here we have assumed that in the given grammar, a string like *John likes Mary* is the value of the term $\overline{\alpha}(\overline{John}, \overline{\beta}(\overline{like}, \overline{Mary}))$, so *John likes 'Mary'* has the same form in the extended grammar, except that the quote marks are introduced by κ, and α and β are extended to α' and β' which apply to quoted strings as well.

We can summarize our extension result for quotation as follows:

Fact 8
Suppose Funct(μ) holds, and μ is extended to a semantics μ' for $\mathbf{E}'$, which is generalized to (S, Ψ) as above.

(a) *μ' handles pure quotation correctly, i.e. it gives the intended meanings to sentences containing such quotation.*

[27] So the semantics is first-grade in the sense of Section 6. In a slightly different, but still first-grade semantics we can add an operator that switches back from V' to μ', e.g. if one wants a mechanism for quantifying into quotation contexts.

$$
\begin{aligned}
\mu'(t) &= r_{isa',\mu'}(\mu'(\overline{\kappa}(\overline{\alpha'}(\overline{John},\overline{\beta'}(\overline{like},\overline{\kappa}(\overline{Mary}))))),\mu'(\overline{sentence}))\\
&= r_{isa'}(\mu'(\overline{\kappa}(\overline{\alpha'}(\overline{John},\overline{\beta'}(\overline{like},\overline{\kappa}(\overline{Mary}))))),\text{SENT})\\
&= r_{isa'}(V'(\overline{\alpha'}(\overline{John},\overline{\beta'}(\overline{like},\overline{\kappa}(\overline{Mary})))),\text{SENT})\\
&= r_{isa'}(\alpha'(V'(\overline{John}),V'(\overline{\beta'}(\overline{like},\overline{\kappa}(\overline{Mary})))),\text{SENT})\\
&= r_{isa'}(\alpha'(John,\beta'(V'(\overline{like}),V'(\overline{\kappa}(\overline{Mary})))),\text{SENT})\\
&= r_{isa'}(\alpha'(John,\beta'(like,\kappa(V'(\overline{Mary})))),\text{SENT})\\
&= r_{isa'}(\alpha'(John,\beta'(like,\text{‘}Mary\text{’})),\text{SENT})\\
&= r_{isa'}(\alpha'(John,like\ \text{‘}Mary\text{’}),\text{SENT})\\
&= r_{isa'}(John\ likes\ \text{‘}Mary\text{’},\text{SENT})\\
&= \text{T iff } John\ likes\ \text{‘}Mary\text{’} \in \text{SENT}
\end{aligned}
$$

Figure 2: Evaluation of (21)

(b) *μ' extends μ in the sense that when $t \in GT_{\mathbf{E}}$, $\mu'(t) = \mu(t)$.*

(c) *If μ satisfies DP, then GDP_S and LC-Funct(S, Ψ) hold, even though Funct(μ') in general fails.*

(d) *(S, Ψ) satisfies GEN: it is first-grade and MAX holds (the semantics is hyperdistinct in quotation contexts while μ' itself applies in the null context type.)*

In the absence of a detailed specification of the given grammar **E**, we have only been able to indicate certain parts of the extension, but it should be fairly clear how to apply these indications to specific grammars.

The extended semantics allows quotation of an arbitrary string z in L^*, using $\overline{\kappa}(s)$, where s is a string term such that $V'(s) = z$. So it can handle all the examples (1). If z is also the string value of a μ'-meaningful grammatical term t, we can quote t instead, with the same result: $\mu'(\overline{\kappa}(t)) = \mu'(\overline{\kappa}(s))$. Using t is then more natural, and will be important, for example, when extending the account to deal with mixed quotation.[28]

[28]What about quoting ill-formed expressions, such as ‘eckullectic’ uttered by Bush, as discussed in Shan [36]? There seem to be three alternatives: (a) the string is in fact part of Bush’s vocabulary, in which case it is well-formed, but part of an unusual idiolect; (b) the utterance was a temporary performance slip, and the reporter uses the string literally to represent Bush’s speech performance, but then it is not a case of quotation proper, rather a case of *depicting* speech by other means (see note 5); (c) the report about Bush really means that he made an utterance that could be reproduced by a standard-rule pronunciation of the string ‘eckullectic’, this is the case we handled in this section. Shan deals with case (a) by adding a parallel language (Bush English) to the fragment, specifying formally how it can be accessed from the main language. We have not attempted anything similar here.

8 Further directions

The work begun here could be extended in at least two directions. First, one should try to cover other uses of quotation, such as the interplay between direct and indirect discourse, or mixed quotation (as in *Quine says that quotation 'has the overwhelming practical convenience of visible reference'*). Indeed, no account of quotation would be complete without this. Our aim in this paper has not been to give a complete account of quotation, however, but to present the notion of general compositionality, and to show that even for the simple case of pure quotation, this kind of compositionality is all one can hope for. Second, as we have indicated, linguistic context dependence as outlined here can be exploited for other constructions, notably various kinds of intensional contexts.

Appendix 1: Another approach?

Do we really need the apparatus of Sections 4 – 7 to take care of pure quotation compositionally? One referee suggested that we could avoid it by treating words as *ambiguous* between a default and a quote reading, so that the lexicon provides, for example, instead of one atomic term $\overline{Cicero}$, two terms, $\overline{Cicero}_d$ and $\overline{Cicero}_q$ with the same surface form: $V(\overline{Cicero}_d) = V(\overline{Cicero}_q) = Cicero$. We have three problems with this idea.

First, it is not a straightforward account (since $\overline{Cicero}_d$ is not a constituent of $\overline{Cicero}_q$ or of any term referring to *Cicero*).

Second, it is counterintuitive. While there are good reasons to let the lexicon distinguish two or more meanings of *bank*, for example, there are no good reasons why the lexicon should care about quotation. Although the same word can sometimes be used to refer to a man and sometimes to itself, this has nothing to do with its lexical meaning but is a systematic feature of the quotation context.

Third, it is not enough to introduce such massive ambiguity for lexical items: the same must be done for *all* complex terms generated by the grammar, since every expression can be quoted. One can think of this carried out in such a way that each term t of the old grammar now comes in two versions, t_d and t_q. However, the distribution of these subscripts is not arbitrary. For example, one should not allow a d(efault) subscript inside a complex term with a q(uote) subscript. To be sure, it is easy to state the rule governing the correct distribution; the rule would exactly mirror our selection function Ψ in Section 7. In other words, the result would essentially be a *notational variant* of our proposal. Where we let μ evaluate t in context type c (or use more than one meaning assignment), the alternative account, using a meaning assignment ν, say, would evaluate t_c: $\nu(t_c) = \mu(t, c)$.

But here is the crucial point. It might *seem* that, in spite of the problems already mentioned, the ambiguity account has the advantage of being able to rely on the standard notion of compositionality, since there is just one meaning assignment, and it applies just to (indexed) terms. If that were true, it would indeed be an advantage. But it isn't: the alternative account would still only be *general* compositional. To see this, recall that LC-Funct(μ, C) requires that to α corresponds an operation r_α such that for every context type c,

$$\mu(\overline{\alpha}(u_1, \ldots, u_n), c) = r_\alpha(\mu(u_1, c_1), \ldots, \mu(u_n, c_n), c),$$

where $c_i = \Phi_C(\alpha, i, c)$. The alternative account would need a similar requirement for the meaning of $\overline{\alpha}(u_1, \ldots, u_n)_c$. This has to take the form

$$\nu(\overline{\alpha}(u_1, \ldots, u_n)_c) = r_\alpha(\nu(u_{1_{c_1}}), \ldots, \nu(u_{n_{c_n}}), c) \tag{30}$$

Thus, we still need a rule to give us the indices $c_1, \ldots, c_n$ and, moreover, we still need the semantic operation to take c as an argument. There is no operation s_α such that (30) can be written

$$\nu(\overline{\alpha}(u_1, \ldots, u_n)_c) = s_\alpha(\nu(u_{1_{c_1}}), \ldots, \nu(u_{n_{c_n}}))$$

In other words, the meaning of a complex expression is *not* determined by the meaning of its immediate parts and the mode of composition (α), you also need the context type (index). This is not compositionality, but general compositionality.

Summing up, we can see no way in which the alternative would be superior, but at least two ways in which it would be inferior. But the main point is that even if we choose the alternative, we still need the notion of general compositionality, and thus the machinery developed in Sections 4 – 7.

Appendix 2: Some proofs

Proof of Proposition 4: (a): Assume that μ is a partial function from $GT_{\mathbf{E}} \times C$ satisfying GDP$_\mu$, where C is a context typing of $CXT_{\mathbf{E}}$ such that LC-Funct(μ, C) holds, and that μ^c for $c \in C$ and Ψ are as described in (a), with $S = \{\mu^c : c \in C\}$. Take $\alpha \in \Sigma$ and $\mu^c \in S$ such that $\mu^c(\overline{\alpha}(u_1, \ldots, u_n))$ is defined. Using GDP$_\mu$, we see that $\Psi(\alpha, i, \mu^c)(u_i)$ is defined, $1 \leq i \leq n$. This shows that GDP$_S$ holds. Also, by LC-Funct(μ, C) we have

$$\begin{aligned}
\mu^c(\overline{\alpha}(u_1, \ldots, u_n)) &= \mu(\overline{\alpha}(u_1, \ldots, u_n), c) \\
&= r_\alpha(\mu(u_1, \Phi_C(\alpha, 1, c)), \ldots, \mu(u_n, \Phi_C(\alpha, n, c)), c) \\
&= r_\alpha(\mu^{\Phi_C(\alpha,1,c)}(u_1), \ldots, \mu^{\Phi_C(\alpha,n,c)}(u_n), c) \\
&= r_\alpha(\Psi(\alpha, 1, \mu^c)(u_1), \ldots, \Psi(\alpha, n, \mu^c)(u_n), c) \\
&= s_{\alpha,\mu^c}(\Psi(\alpha, 1, \mu^c)(u_1), \ldots, \Psi(\alpha, n, \mu^c)(u_n)),
\end{aligned}$$

where s_{α,μ^c} is defined by

$$s_{\alpha,\mu^c}(m_1, \ldots, m_n) = r_\alpha(m_1, \ldots, m_n, c)$$

if there are $t_1, \ldots, t_n$ such that $\mu^c(\overline{\alpha}(t_1, \ldots, t_n))$ is defined and $m_i = \Psi(\alpha, i, \mu^c)(t_i)$, $1 \leq i \leq n$ (undefined otherwise).

Note that s_{α,μ^c} is well-defined, since it follows from the above that if $\mu^c = \mu^{c'}$, then $r_\alpha(m_1, \ldots, m_n, c) = r_\alpha(m_1, \ldots, m_n, c')$. This proves LC-Funct$(S, \Psi)$.

(b): Now suppose S is a set of functions and Ψ a selection function satisfying GDP$_S$ such that LC-Funct(S, Ψ) holds. First, note that F as defined in (b) is a well-defined partial function, by induction over the length of ξ: if it is clear for which contexts of length at most k F is defined and what its values are, and η is a context of length $k+1$, then η has the form $\xi^\frown\langle(\alpha, i)\rangle$. If $F(\xi)$ is undefined, or if $F(\xi)$ is an element of S which is not defined for any term of the form $\overline{\alpha}(u_1, \ldots, u_n)$, then $F(\eta)$ is undefined. Otherwise, $F(\eta) = \Psi(\alpha, i, F(\xi))$; this is defined by GDP$_S$.

Next, note that the definition of the sets $[\xi]$ is formulated in such a way that if $F(\xi)$ is defined, $[\xi] = \{\xi' \colon F(\xi') = F(\xi)\}$, whereas if $F(\xi)$ is undefined, $[\xi] = \{\xi' \colon F(\xi') \text{ is undefined}\}$. Thus, C is a partition of $CXT_{\mathbf{E}}$. We must check that C is a context typing, i.e. that condition (i) in (L$_{\text{ctype}}$) holds.

So suppose $[\xi] = [\xi']$. If $F(\xi)$ is defined, then so is $F(\xi')$ and $F(\xi') = F(\xi)$. If there is a term $\overline{\alpha}(u_1, \ldots, u_n)$ such that $F(\xi)(\overline{\alpha}(u_1, \ldots, u_n))$ is defined, then $F(\xi^\frown\langle(\alpha, i)\rangle)$ is defined, and $\Psi(\alpha, i, F(\xi)) = \Psi(\alpha, i, F(\xi'))$, so $F(\xi^\frown\langle(\alpha, i)\rangle) = F(\xi'^\frown\langle(\alpha, i)\rangle)$, i.e. $[\xi^\frown\langle(\alpha, i)\rangle] = [\xi'^\frown\langle(\alpha, i)\rangle]$, as desired. If there is no such term, then $F(\xi^\frown\langle(\alpha, i)\rangle)$ and $F(\xi'^\frown\langle(\alpha, i)\rangle)$ are both undefined, so again $[\xi^\frown\langle(\alpha, i)\rangle] = [\xi'^\frown\langle(\alpha, i)\rangle]$. Finally, if $F(\xi)$ is undefined, so is $F(\xi')$, and again $F(\xi^\frown\langle(\alpha, i)\rangle)$ and $F(\xi'^\frown\langle(\alpha, i)\rangle)$ are both undefined. This proves (i).

Next, it clearly follows from our definitions that the function ν is well-defined, i.e. that if $[\xi] = [\xi']$, then, for any term t, $\nu(t, \xi)$ is defined iff $\nu(t, \xi')$ is defined, and when both are defined they have the same value.

We must also verify that GDP$_\nu$ holds. Suppose $\nu(\overline{\alpha}(u_1, \ldots, u_n), [\xi])$ is defined. Then $F(\xi)$ is defined and $\nu(\overline{\alpha}(u_1, \ldots, u_n), [\xi]) = F(\xi)(\overline{\alpha}(u_1, \ldots, u_n))$. By GDP$_S$ and the definition of F, $\Psi(\alpha, i, F(\xi))(u_i) = F(\xi^\frown\langle(\alpha, i)\rangle)(u_i)$ is defined. Thus, each $\nu(u_i, [\xi^\frown\langle(\alpha, i)\rangle])$ is defined. Again, this reasoning is independent of the choice of context in $[\xi]$. Thus, GDP$_\nu$ holds.

Finally, take a rule α and a context type $c \in C$ such that $\nu(\overline{\alpha}(u_1, \ldots, u_n), c)$ is defined. Let $c = [\xi]$. Using LC-Funct(S, Ψ) and the definitions of F, ν, and Φ_C, we calculate:

$$\begin{aligned}
\nu(\overline{\alpha}(u_1, \ldots, u_n), [\xi]) &= F(\xi)(\overline{\alpha}(u_1, \ldots, u_n)) \\
&= r_{\alpha,F(\xi)}(\Psi(\alpha, 1, F(\xi))(u_1), \ldots, \Psi(\alpha, n, F(\xi))(u_n)) \\
&= r_{\alpha,F(\xi)}(F(\xi^\frown\langle(\alpha, 1)\rangle)(u_1), \ldots, F(\xi^\frown\langle(\alpha, n)\rangle)(u_n)) \\
&= r_{\alpha,F(\xi)}(\nu(u_1, [\xi^\frown\langle(\alpha, 1)\rangle]), \ldots, \nu(u_n, [\xi^\frown(\alpha, n)])) \\
&= s_\alpha(\nu(u_1, [\xi^\frown\langle(\alpha, 1)\rangle]), \ldots, \nu(u_n, [\xi^\frown(\alpha, n)]), [\xi]) \\
&= s_\alpha(\nu(u_1, \Phi_C(\alpha, 1, [\xi])), \ldots, \nu(u_n, \Phi_C(\alpha, n, [\xi])), [\xi])
\end{aligned}$$

where the operation s_α is defined by

$$s_\alpha(m_1, \ldots, m_n, [\xi]) = r_{\alpha,F(\xi)}(m_1, \ldots, m_n)$$

These calculations are independent of the choice of ξ, as long as $F(\xi)$ is defined, which is an assumption. Therefore, s_α is well-defined. This shows that LC-Funct(ν, C) holds. $\square$

Proof of Fact 6: Let μ be given, and let the context typing C be such that each context is its own type, i.e. $[\xi] = \{\xi\}$. Define, for each ξ, the relation $\equiv^\xi$ by

(31) $\quad u \equiv^\xi t$ iff for all terms s containing ξ, $s[(u,\xi)] \equiv_\mu s[(t,\xi)]$

$\equiv^\xi$ is a partial equivalence relation on $GT_{\mathbf{E}}$, with $dom(\equiv^\xi) = \{u : u \equiv^\xi u\}$. Let $[[t]]^\xi = \{t' : t' \equiv^\xi t\}$ be the equivalence class of t under $\equiv^\xi$. Note that, since for all terms s, t, $s[(t, \langle\rangle)] = t$, we have

(32) $\quad \equiv^{\langle\rangle} \; = \; \equiv_\mu$

Then let, for each ξ, $\mu^{[\xi]}$ be a semantics whose corresponding synonymy relation is $\equiv^\xi$, i.e. $\equiv_{\mu^{[\xi]}} = \equiv^\xi$. For definiteness, put

(33) $$\mu^{[\xi]}(t) = \begin{cases} \mu(t) & \text{if } \xi = \langle\rangle \text{ and } \mu(t) \text{ is defined} \\ [[t]]^\xi & \text{if } \xi \neq \langle\rangle \text{ and } t \in dom(\equiv^\xi) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then (cf. Hodges [15], Lemma 1), we have

$$\equiv_{\mu^{[\xi]}} \; = \; \equiv^\xi$$

Also,

$$\mu^{[\langle\rangle]} \; = \; \mu$$

so (S, Ψ) generalizes μ, with $S = \{\mu^{[\xi]} : \xi \in CXT_{\mathbf{E}}\}$, and

$$\Psi(\alpha, i, \mu^{[\xi]}) = \mu^{[\xi^\frown\langle(\alpha,i)\rangle]}$$

Now it is immediate from (31) that MAX holds. To show compositionality, we prove:

(34) $\quad$ If s contains ξ, and if $t \equiv^{\xi'^\frown\xi} u$, then $s[(t,\xi)] \equiv^{\xi'} s[(u,\xi)]$.

For let s' be any term containing the context ξ'. Then, for all terms v, $s'' = s'[(s[(v,\xi)], \xi')]$ contains the context $\xi'^\frown\xi$, and

$$s'[(s[(v,\xi)], \xi')] = s''[(v, \xi'^\frown\xi)]$$

By assumption and (31), $s''[(t, \xi'^\frown\xi)] \equiv_\mu s''[(u, \xi'^\frown\xi)]$, and so $s'[(s[(t,\xi)], \xi')] \equiv_\mu s'[(s[(u,\xi)], \xi')]$. Since s' was arbitrary, it follows by (31) that $s[(t,\xi)] \equiv^{\xi'} s[(u,\xi)]$. This proves (34).

Now, in the chosen context typing C, the function Θ_C from (15) in Section 4.2 is simply given by

$$\Theta_C([\xi'], \xi) = [\xi'^\frown\xi]$$

But this means that (34) entails LC-Subst(μ, C), or if you wish the corresponding principle for (S, Ψ). Note here that LC-Subst(μ, C) is about simultaneous substitution of terms, but in view of (31), if $t \equiv^{\xi'^\frown\xi} u$, substitution of u for t can never lead from a meaningful to a meaningless term. Therefore, the substitutions in LC-Subst(μ, C) can be performed one by one,[29] and (34) says that meaning is preserved at each step. Thus, LC-Funct(S, Ψ) holds as well (cf. Proposition 4 and Fact 5.) $\square$

Proof of Fact 7: We prove the case $n = 1$ (i.e. GEN) and indicate how the idea extends to other n. Consider a grammar with three atomic terms a, b, c and one one-place operator α, and let μ be a total semantics such that

$$a \equiv_\mu b \equiv_\mu c, \quad \overline{\alpha}(a) \equiv_\mu \overline{\alpha}(b) \not\equiv_\mu \overline{\alpha}(c), \text{ and } \overline{\alpha}(\overline{\alpha}(a)) \not\equiv_\mu \overline{\alpha}(\overline{\alpha}(b)).$$

We claim that μ is not compositionally generalizable. Assume, for *reductio*, that there is a pair (S, Ψ) generalizing μ that satisfies GEN. Observe first that since in the context $\xi = \langle(\alpha, 1)\rangle$ we have

$$\overline{\alpha}(b) = \overline{\alpha}(b)[(b, \xi)] \not\equiv_\mu \overline{\alpha}(b)[(c, \xi)] = \overline{\alpha}(c),$$

it follows by LC-Funct(S, Ψ) (or the corresponding substitution condition) that $b \not\equiv_{\mu^{[\xi]}} c$, where $\mu^{[\xi]} = \Psi(\alpha, 1, \mu)$. Since $b \equiv_\mu c$, the pair $(\alpha, 1)$ is not a keeper, whence it must be a switcher, which means that:

(35) For any function ν in S, $\Psi(\alpha, 1, \nu) = \mu^\xi$.

Next, by assumption, for any term s containing ξ we have

$$s[(a, \xi)] \equiv_\mu s[(b, \xi)]$$

It then follows by MAX that

(36) $a \equiv_{\mu^{[\xi]}} b$

Let $\xi' = \xi^\frown\langle(\alpha, 1)\rangle$. Now, since

$$\overline{\alpha}(\overline{\alpha}(a)) = \overline{\alpha}(\overline{\alpha}(a))[(a, \xi')] \not\equiv_\mu \overline{\alpha}(\overline{\alpha}(a))[(b, \xi')] = \overline{\alpha}(\overline{\alpha}(b))$$

we must, as above, have $a \not\equiv_{\mu^{[\xi']}} b$, where $\mu^{[\xi']} = \Psi(\alpha, 1, \mu^{[\xi]})$. But from (35) it follows that $\mu^{[\xi']} = \mu^{[\xi]}$, which contradicts (36). Thus, μ is not first-grade compositionally generalizable.

We can easily extend this substitution failure format to yield a function μ that is not generalizable into a *second-grade* g-compositional semantics: just start with four (μ-)synonymous atoms, of which three are synonymous under $\overline{\alpha}$, of which two are synonymous under $\overline{\alpha}(\overline{\alpha})$, which in turn are not synonymous under $\overline{\alpha}(\overline{\alpha}(\overline{\alpha}))$. And so on. $\square$

[29]Hodges [15] notes that a sufficient condition for this is the *Husserl property*, i.e. that synonymous terms are meaningful in the same linguistic contexts. The semantics given in (31) clearly has the Husserl property.

The construction in this proof gives a natural generalization of the substitution failure format, where standard substitution failure is the first element of this sequence, with two terms that are atomically synonymous but not under $\overline{\alpha}$. This also speaks in favor of the naturalness of the generalization requirements in GEN. We may note that Frege's infinite hierarchy of indirect reference can provide substitution failure for generalization of any finite grade.

References

[1] Chris Barker and Pauline Jacobson, editors. *Direct Compositionality*. Oxford University Press, Oxford, 2007.

[2] Herman Cappelen and Ernest LePore. *Language Turned on Itself. The Semantics and Pragmatics of Metalinguistic Discourse*. Oxford University Press, Oxford, 2007.

[3] Herman Cappelen and Ernest LePore. Quotation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2008.

[4] Herbert Clark and Richard Gerrig. Quotations as demonstrations. *Language*, 66(4):764–805, 1990.

[5] Donald Davidson. Quotation. *Theory and Decision*, 11:27–40, 1979. Reprinted in [6], 79–92.

[6] Donald Davidson. *Inquiries into Truth and Interpretation*. Clarendon Press, Oxford, 1984.

[7] David Dowty. Compositionality as an empirical problem. In Barker and Jacobson [1], pages 23–101.

[8] Gottlob Frege. Über Sinn und Bedeutung. In Peter Geach and Max Black, editors, *Translations from the Philosophical Writings of Gottlob Frege,* 1952, pages 56–78. Blackwell, Oxford, 1892.

[9] Gottlob Frege. Letter to Russell 28.12 1902. In Gottfried Gabriel, editor, *Wissenschaftlicher Briefwechsel*, pages 234–7. Felix Meiner Verlag, Hamburg, 1976.

[10] P. Geach. *Mental Acts*. Routledge Kegan Paul, London, 1957.

[11] Kathrin Glüer and Peter Pagin. Proper names and relational modality. *Linguistics & Philosophy*, pages 507–35, 2006.

[12] Kathrin Glüer and Peter Pagin. Relational modality. *Journal of Logic, Language and Information*, 17:307–22, 2008.

[13] Kathrin Glüer and Peter Pagin. General terms and relational modality. *Nous*, forthcoming, 2011.

[14] Wolfram Hinzen, Edouard Machery, and Markus Werning, editors. *The Oxford Handbook of Compositionality*. Oxford University Press, Oxford, to appear.

[15] Wilfrid Hodges. Formal features of compositionality. *Journal of Logic, Language and Information*, 10:7–28, 2001.

[16] David Kaplan. Demonstratives: An essay on the semantics, logic, metaphysics, and epistemology of demonstratives and other indexicals. In Joseph Almog, John Perry, and Howard Wettstein, editors, *Themes from Kaplan*, pages 481–566. Oxford University Press, Oxford, 1989.

[17] Christopher Kennedy. Vagueness and grammar: the semantics of relative and absolute gradable adjectives. *Linguistics & Philosophy*, 30:1–45, 2007.

[18] David Lewis. Index, context, and content. In Stig Kanger and Sven Öhman, editors, *Philosophy and Grammar*. D. Reidel, Dordrecht, 1980.

[19] Peter Pagin. Compositionality and context. In Gerhard Preyer and Georg Peter, editors, *Contextualism in Philosophy*, pages 303–48. Oxford University Press, Oxford, 2005.

[20] Peter Pagin. Communication and the complexity of semantics. In Hinzen et al. [14].

[21] Peter Pagin. General compositionality and belief sentences. Talk given at SPE 3, Paris, May 2010.

[22] Peter Pagin. Compositionality, computability, and complexity. Manuscript, unpublished.

[23] Peter Pagin and Dag Westerståhl. Compositionality I: Definitions and variants. *Philosophy Compass*, 5(3):250–264, 2010.

[24] Peter Pagin and Dag Westerståhl. Compositionality II: Arguments and problems. *Philosophy Compass*, 5(3):265–282, 2010.

[25] Terence Parsons. What do quotation marks name? Frege's theory of quotations and that clauses. *Philosophical Studies*, 42:3:315–328, 1982.

[26] Barbara H. Partee. Compositionality. In Fred Landman and Frank Veltman, editors, *Varieties of Formal Semantics*, pages 281–311. Foris, Dordrecht, 1984. Reprinted in Barbara H. Partee (2004), *Compositionality in Formal Semantics*, Blackwell, Oxford, 153–181.

[27] Christopher Potts. The dimensions of quotation. In Barker and Jacobson [1], pages 405–31.

[28] Stefano Predelli. The demonstrative theory of quotation. *Linguistics & Philosophy*, 31:5:555–572, 2008.

[29] W.V.O. Quine. *Mathematical Logic*. Harvard University Press, Boston, Mass., 1940.

[30] W.V.O. Quine. *Word and Object*. MIT Press, Cambridge, Mass., 1960.

[31] François Recanati. Open quotation. *Mind*, 110:637–687, 2001.

[32] François Recanati. Open quotation revisited. *Philosophical Perspectives*, 22:443–471, 2008.

[33] François Recanati. Compositionality, flexibility and context-dependence. In Hinzen et al. [14].

[34] P. Saka. Quotation and the use-mention distinction. *Mind*, 107:113–136, 1998.

[35] John Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, 1969.

[36] Chung-chieh Shan. The character of quotation. *Linguistics and Philosophy*, this volume, 2011.

[37] A. Tarski. The semantic conception of truth. In L. Linsky, editor, *Semantics and the Philosophy of Language*, pages 13–47. University of Illinois Press, Urbana, 1952.

[38] A. Tarski. The concept of truth in formalized languages. In *Logic, Semantics, Metamathematics, 2nd ed.*, pages 152–278. Hackett, 1983, Indianapolis, 1983.

[39] C. Washington. The identity theory of quotation. *The Journal of Philosophy*, 89:582–605, 1992.

[40] Markus Werning. Right and wrong reasons for compositionality. In Edouard Machery, Markus Werning, and Gerhard Schurz, editors, *The Compositionality of Meaning and Content: Foundational Issues*, volume I, pages 285–309. Ontos, Frankfurt/Lancaster, 2005.

[41] Dag Westerståhl. On mathematical proofs of the vacuity of compositionality. *Linguistics & Philosophy*, 21:635–643, 1998.

[42] Dag Westerståhl. On the compositionality of idioms; an abstract approach. In D. Barker-Plummer et al., editor, *Words, Proofs, and Diagrams*, pages 241–271. CSLI Publications, Stanford, 2002.

[43] Dag Westerståhl. On the compositional extension problem. *Journal of Philosophical Logic*, 33:549–582, 2004.

[44] Dag Westerståhl. Compositionality in Kaplan style semantics. In Hinzen et al. [14].