Paracompositionality, MWEs and Argument Substitution

Cem Bozşahin and Arzu Burcu Güven

Cognitive Science Department, Informatics Institute
Middle East Technical University (ODTÜ), Ankara Turkey
bozsahin@metu.edu.tr arzuburcuguven@gmail.com

Abstract. Multi-word expressions, verb-particle constructions, idiomatically combining phrases, and phrasal idioms have something in common: not all of their elements contribute to the argument structure of the predicate implicated by the expression.

Radically lexicalized theories of grammar that avoid string-, term-, logical form-, and tree-writing, and categorial grammars that avoid wrap operation, make predictions about the categories involved in verb-particles and phrasal idioms. They may require singleton types, which can only substitute for one value, not just for one kind of value. These types are asymmetric: they can be arguments only. They also narrowly constrain the kind of semantic value that can correspond to such syntactic categories. Idiomatically combining phrases do not subcategorize for singleton types, and they exploit another locally computable and compositional property of a correspondence, that every syntactic expression can project its head word. Such MWEs can be seen as empirically realized categorial possibilities rather than lacuna in a theory of lexicalizable syntactic categories.

1 Introduction

A type is a set of values. When we write a syntactic type, say NP, we mean a set of expressions (values) which can substitute for that type. This type serves to distinguish some expressions from for example the set of expressions that can substitute for a VP type.

The distinction is crucial for solving the correspondence problem in syntax-semantics. For this purpose we talk about semantic types, for example e for things and t for propositions. The concepts that can substitute for semantic types are not expressions in the sense that syntactic expressions are, because they are not observable, but they leverage a theory to hypothesize about the kind of semantics that these types stand for.

These two species of types are then put in a correspondence in a theory of syntax-semantics connection. The understanding is that if one substitutes a certain expression for a syntactic type, then its corresponding semantic type substitutes for a certain kind of semantic value. We know less about the semantic values; but, at the level of the

correspondence problem, this is not very critical. It is however crucial to make the distinctions and propagate them in a parsing mechanism rather than solving all type-interpretation problems in one go.

We need a theory which provides explicit vocabulary and mechanism for the correspondence, to be more specific about the equal relevance of substitution for subexpressions which purportedly do not contribute to the meaning of the expression.

In the categorial grammar parlance, for which we will use Combinatory Categorial Grammar [30, 31], hereafter CCG, we can exemplify the correspondence as follows, where we use the "result-first argument-next" notation:

(1) a. hits := $(S \backslash NP_{3s})/NP$: $\lambda x \lambda y.hit'xy$
     b. hit := VP_{inf}/NP: $\lambda x \lambda y.hit'xy$

Some syntactic types are further narrowed down by features, such as NP_{3s} above for third person singular NP, which are, in CCG, not re-entrant.

We argue in the paper that in a radically lexicalized theory which adheres to transparency of derivations by type substitution (rather than lexical insertion), such as CCG, there are built-in degrees of freedom to support Multi-word Expressions (MWEs) and idioms without complicating the mechanism.

Paracompositionality is key to projection of their properties in a derivation. It is the idea that, in addition to the compositionality of the lexical correspondence, which is compositional partly because it relies on non-vacuous abstractions, type substitution by (i) what we call singleton types and (ii) what is called head-dependencies in the NLP literature is also compositional because it spells non-vacuous abstraction as part of the correspondence, but as something related to the contingency of the predicate, rather than the argument structure of the predicate. In a radically lexicalized grammar both sources are available in a lexical item. These types are paracompositional also in the sense that whether we have an idiom reading or compositional one is already decided by the category of the head in the derivational process.

The term *contingency* is used here in the sense of Moens and Steedman [24] where it relates to extension of happenings. In the case of events (culminations, points, processes and culminating processes), which have definite extension, it is an event modality of space, time and manner; and, in the case of states where extension is indefinite (e.g. *understand*) it is some property of the state. From now on when we use the term 'contingency' we mean something related to extension of the predicate, rather than who does what to whom in the predicate.

MWEs are expressions involving more than one word in which the properties of the expression are not determined by the composition of the properties of the constituent words, which would be the case for phrases. There is a tendency to treat them as single lexical units [10, 33]; but, as we shall see, CCG does not require the single unit to be the phonological representation to the left of ':=' in the format of (1). This property of CCG naturally extends to coverage of verb-particle constructions e.g. *look the word up* as discontiguous MWEs headed by a lexical item.

Phrasal idioms and idiomatically combining phrases are classes identified by Nunberg, Sag and Wasow [25] to account for systematic variation in syntactic productivity of idioms. Typewise they will relate to singleton types (phrasal idioms) and head-word subcategorization (idiomatically combining phrases) in our formulation.

As a preview of the article, we can think of the meaning distinctions as ranging from "beans" i.e. the nounphrase *beans* itself as a category (this is what we call the singleton type); to NP_{beans} as the category of an NP headed by the word *beans*, which has wider range of substitution; and, to the polyvalent NP with the widest substitution for that type. This much is categorial grammar with type substitution. CCG as an empirical theory adds to this the claim that there is an asymmetry in the range of substitutions: the singleton types can be arguments only, and arguments of arguments and results, but never the result. We shall see that this has implications for the linguist's choice of handling syntactic productivity in a grammar.

Some implications follow: Because of paracompositionality, all expressions requiring a singleton type would involve the semantic type of a predicate, and all idiomatically combining phrases requiring a different interpretation than the compositional one would have the same consequence independent of their syntactic productivity. In short, every idiom must contain a predicate (but not necessarily a verb). We cover these implications in the article.

2 Substitution in a Derivation

In (1a), the '$/NP$' can be substituted for by certain kinds of expressions, for example *John, me, the ball, a stone in the corner*, etc. Its corresponding semantic counterpart in the logical form (LF), written after the colon, has the placeholder x which can be typed as e, to be suitably substituted for by a semantic value described above. The '$\backslash NP_{3s}$' can be substituted by narrower expressions, for example eliminating *I, you*. Because this is an indirect correspondence, its semantic counterpart y can have the same type e.

The tacit assumption of indirectness is sometimes made explicit, for example in Bach's [2] rule-by-rule hypothesis: The derivational process operates with syntactic types only, and when it applies the semantics of the rule, its semantics works only with LF objects. Quoting from Bach: "Neither type of rule has access to the representations of the other type except at the point where a translation rule corresponding to a given syntactic rule is applied." The "syntactic rule" in a lexicalized grammar such as CCG is the combinatory syntactic type of a lexical correspondence. The" translation rule" is the lexically-specified logical form, LF, as in (1).
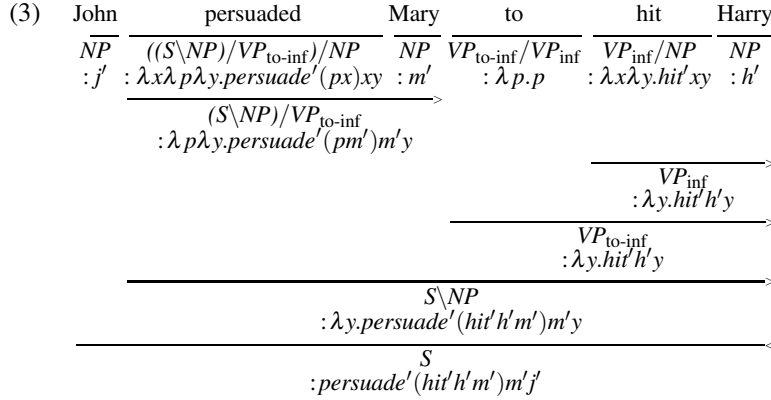
The derivational process reveals partially derived types, for example $S\backslash NP_{3s}$: $\lambda y.hit's'y$ for (1a), if function application substitutes say *a stone* for '$/NP$', with some semantic value s'. The semantic type of such derived categories is concomitantly functional, e.g. $e \mapsto t$ for this syntactic type. *John hits* is $e \mapsto t$ too, with category S/NP: $\lambda x.hit'xjohn'$.

We can see the relevance of derived types to substitutability in a closer look at (1b). If function application substitutes for the '$/NP$' in the example, the derived category would be VP_{inf}: $\lambda y.hit's'y$ in this case. This is also an $e \mapsto t$ type semantically. However, its syntax is narrower so that we can account for the expressions in (2).[1]

---

[1] This is equivalent to saying that in CCG the type *VP* is not always an abbrevation for $S\backslash NP$, which might be the case in other brands of categorial grammars. The English facts above could be taken care of by featural distinctions such as S_{inf}, $S_{\text{to-inf}}$, S_{fin} in $S\backslash NP$, rather than also positing a *VP*. But in ergative languages the '$\backslash NP$' does not always coincide with the same LF role

(2) John persuaded Mary to/* hit/*hits the target.

The derivational process works as below, with $VP_{\text{to-inf}}$ distinct from VP_{inf}.

(3)

John	persuaded	Mary	to	hit	Harry
NP	$((S\backslash NP)/VP_{\text{to-inf}})/NP$	NP	$VP_{\text{to-inf}}/VP_{\text{inf}}$	VP_{inf}/NP	NP
$:j'$	$:\lambda x\lambda p\lambda y.persuade'(px)xy$	$:m'$	$:\lambda p.p$	$:\lambda x\lambda y.hit'xy$	$:h'$

$$\frac{}{\begin{array}{c}(S\backslash NP)/VP_{\text{to-inf}}\\:\lambda p\lambda y.persuade'(pm')m'y\end{array}}{}^>$$

$$\frac{}{\begin{array}{c}VP_{\text{inf}}\\:\lambda y.hit'h'y\end{array}}{}^>$$

$$\frac{}{\begin{array}{c}VP_{\text{to-inf}}\\:\lambda y.hit'h'y\end{array}}{}^>$$

$$\frac{}{\begin{array}{c}S\backslash NP\\:\lambda y.persuade'(hit'h'm')m'y\end{array}}{}^>$$

$$\frac{}{\begin{array}{c}S\\:persuade'(hit'h'm')m'j'\end{array}}{}^<$$

Here function application is shown in forward form ($>$) and backward form ($<$). Derivation proceeds from top to bottom in display, as standard in CCG; i.e., bottom-up as far as parsing is concerned, and one at a time. For brevity alternative derivations using function composition are not shown; their implications for constituency are discussed in Steedman references. We also eschew the slash modalities of Baldridge and Kruijff [3] to avoid digression, which can further restrict the combination possibilities of syntactic types. They are mentioned later when they are relevant to discussion. The LF contains a structured form, viz. the predicate-argument structure, which is written in linear notation for simplicity; for example $hit'xy$ is same as $((hit'x)y)$; i.e., it is left-associative.

In preparation for final discussion of substitution (§6) in relation to the wrapping operation, we can redraw this derivation by showing the substituting expressions as we proceed, which we do in Figure 1.

MWEs present a challenge for substitution in such correspondences. In Schuler and Joshi's [29]:25 words: "In the *pick .. up* example, there is no coherent meaning for *Up* such that $[\![pick\ X\ Up]\!] = Pick([\![X]\!], Up)$." They go on to show how tree-write in the form of TAG transformations, rather than string-rewrite of CFG transformations such as [28], can deliver different meanings of such expressions *after* a fully compositional tree is established for 'pick', '..' and 'up'.

In such systems, post-processing and reanalysis of a categorial surface derivation are possible, both for TAG and HPSG,[2] therefore these transformations are possible, indeed useful, to simplify large-scale grammar development.

as it does in English, such as in Dyirbal's control construction, where the controlled absolutive argument can be the patient NP of the transitive clause or syntactic subject of an intransitive clause, but not the ergative NP of the transitive clause. It seems to require $VP:\lambda x.pred'x$ where x's role in the controlled clause $pred'$ is determined by verbal morphology of the controlled clause; see [23] for the phenomenon. Assuming a VP cross-linguistically makes narrower predictions about control. We handle this problem elsewhere.

[2] TAG transformations take a phrase structure tree and decompose it to elementary structures to deliver an LF. [21] is a different TAG way to incorporate meaning postulates of [26]. HPSG uses phrasal post-classification to the same effect. For example [4, 28] perform it at the final stage of parsing as a semantic check on bags of predicates for idiom entries, and [17] use semantic frame identification, viz. compositional vs idiomatic, which are built in to theory.
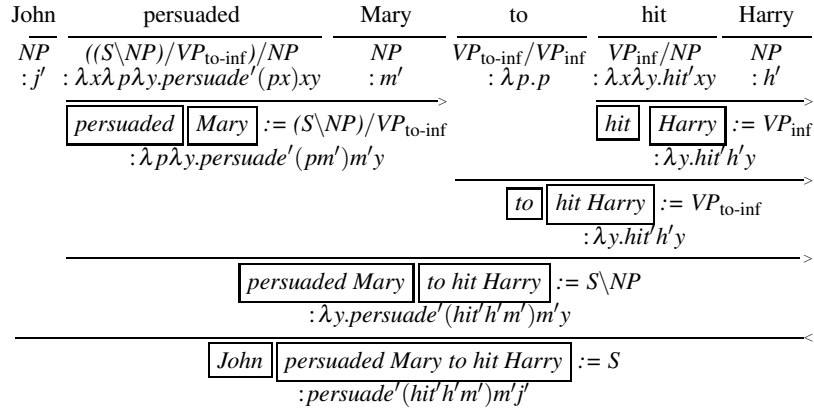
John	persuaded	Mary	to	hit	Harry
NP	$((S\backslash NP)/VP_{\text{to-inf}})/NP$	NP	$VP_{\text{to-inf}}/VP_{\text{inf}}$	VP_{inf}/NP	NP
$:j'$	$:\lambda x\lambda p\lambda y.persuade'(px)xy$	$:m'$	$:\lambda p.p$	$:\lambda x\lambda y.hit'xy$	$:h'$

$$
\boxed{persuaded}\,\boxed{Mary} := (S\backslash NP)/VP_{\text{to-inf}} \qquad \boxed{hit}\,\boxed{Harry} := VP_{\text{inf}}
$$
$$
:\lambda p\lambda y.persuade'(pm')m'y \qquad\qquad :\lambda y.hit'h'y
$$

$$
\boxed{to}\,\boxed{hit\ Harry} := VP_{\text{to-inf}}
$$
$$
:\lambda y.hit'h'y
$$

$$
\boxed{persuaded\ Mary}\,\boxed{to\ hit\ Harry} := S\backslash NP
$$
$$
:\lambda y.persuade'(hit'h'm')m'y
$$

$$
\boxed{John}\,\boxed{persuaded\ Mary\ to\ hit\ Harry} := S
$$
$$
:persuade'(hit'h'm')m'j'
$$

Fig. 1. Substitution of syntactic expressions for syntactic types. Boxes show segments combined. We display some one-at-a-time derivations on the same line to save space.

For radically lexicalized grammars such as CCG where such options are not available, three paths to maintaining compositionality in the presence of "non-compositional" and/or idiomatic parts seem to be available:

(4) a. letting the logical form change the compositional meaning,
     b. introducing surface wrap,
     c. reassessing the substitutability of *argument* types, to the extent that (i) they can be narrowed by head-dependencies, and (ii) the semantic contribution of some parts of the correspondence to the predicate-argument structure can be ignored in a principled way, and locally.

The problem is exacerbated by phrasal idioms which seem to have partially active syntax in some non-compositional parts, for example *kick the (proverbial/old) bucket*, but note ♯*the bucket that John kicked*, ♯*kick the great bucket in the sky*, and *the breeze was shot*. (♯ is used to indicate unavailability of idiomatic reading. The last two examples and judgments are from [28].) However, there are also phrasal idioms which are syntactically quite active, e.g. *the beans that John spilled*, and *spilling the musical/artistic/juicy beans*.

Option (4a) does not always necessitate post-processing of MWEs in CCG, but, as we shall see later in (23), it does not guarantee locality of derivations either. One way to realize it is the following:

(5) $kicked := (S\backslash NP)/NP : \lambda x\lambda y.\text{if } head(x) = bucket' \text{ then } die'y \text{ else } kick'xy$

This approach to phrasal idioms which is similar to meaning postulates for the same task such as [26] would then have to make sure that the head meaning *bucket'* has some predefined cluster of modifiers such as *proverbial* or *old*, but not much else, for example ♯*kick the bucket that overflowed*. It would also have to overextend itself to avoid the idiomatic reading in ♯*the bucket that you kicked*.

The diversity of approaches in the volume for idioms [14] is testimony to the practice that the idioms are decisive factors in polishing our theories linguistically, psychologically and computationally.

As an alternative, the type NP_{bucket} below is inspired by trainable stochastic CFGs which can distinguish argument PPs from adjunct PPs by encoding head dependencies for CFG rules, for example $VP_{\text{put}} \rightarrow V_{\text{put}}$ NP PP_{on}: (We shall fix the unaccounted vacuous abstraction in it later in the paper.)

(6)   kicked := $(S\backslash NP)/NP_{\text{bucket}}: \lambda x \lambda y.die'y$

It might appear to be LF-motivated just like (5) above; but, it is actually a case of (4c/i). NP_{bucket}, meaning NP headed by *bucket*, can be made distinct from NP_{buckets} because different surface expressions can be substituted for them. (6) overgenerates for the examples given above, but it might be the right degree of freedom to exploit in the syntax-semantics correspondence of idiomatically combining MWEs such as NP_{beans} for *spill the beans*.

In the remainder of the paper, we show that option (4c/ii) has been implicit in CCG theory all along but never used, in the form of syntactic types for which only one value can substitute (§3). We call them singleton types. This way of lexical categorization and subcategorization predicts very limited syntax, but not as metalinguistic marking that [28] proposed for *kick the proverbial/old bucket*. It is due to having to enumerate different senses and contingencies of phrasal idioms (e.g. proverb bucket for senses above, also covering e.g. *when I face the proverbial bucket*), and *pick up* for MWEs. In §4 we show that idiomatically combining phrases have principled distinctions from singleton types. Head-word subcategorization such as (6) is the more promising option for them, which radically lexicalized grammars can handle without extension. There are also idioms which require analysis combining both options such as those with semantic reflexives where the referent is not part of the idiom, e.g. *I twiddle my/*his thumbs*. §5 covers these cases.

These findings reveal some aspects of type substitution and its projection when the expressions are not fully compositional at the level of the predicate-argument structure. As such they may have implications beyond CCG.

Finally we show that adopting option (4b) to analyze for example *pick $\cdots$ up* as *pick up* $(\cdots)_{\text{wrap}}$ overgenerates in the combinatory version of wrap (§6), and complicates the grammar with a domino-effect in the surface version of wrap; therefore, it would do more damage than good if adopted for (discontiguous) MWEs and phrasal idioms. CCG can continue to avoid all forms of wrap in the presence of all kinds of MWEs and phrasal idioms.

3   Singleton Types

A brief preview of the proposal for (4c/ii) is as follows. A singleton syntactic type self-represents because it can substitute for one value only. We designate such types with strings, such as *"up"* or *"the bucket"*; for example:

(7)   a.   picked := $(S\backslash NP)/"up"/NP: \lambda y \lambda x \lambda z.cause'(init'(hold_x'yz))z$
        b.   kicked := $(S\backslash NP)/"the\ bucket": \lambda x \lambda y.die_x'y$

(*Init'* is a function that yields a culminating state in the sense of [24].)

We call categories in (7) 'paracompositional' to highlight the fact that, although their LF correspondence is intact so that the derivational process is transparent, they

might have seemingly vacuous abstraction from the perspective of the predicate-argument structure, symbolized by the placeholders x above.[3]

However, one can make a case that this abstraction, corresponding respectively to singleton categories *"up"* and *"the bucket"*, might have a role inside the LF constants shown in primes, as contingencies. We write them for example as $die'_x y$ (as ceremonial death, reported death, etc.), rather than $die' y$. These LF 'constants' are convenient generalizations in CCG standing in for a plethora of features anyway, so it seems natural to think of them as having their own abstraction. (The semantic types corresponding to these contingencies are then $\alpha \mapsto t$ for some α.)

It will be seen in §3.2 that the examples in (7) differ in their sense from *picked up the book* and *kicked the blue bucket*, therefore a separate grammar entry is empirically justified. The sense distinction is reflected explicitly in the LF, as we shall see later. Both possibilities for substitution, for the syntactic type and for its placeholder in the LF, are principally restricted by CCG.

Singletons also engender a way for such entries to be morphologically more transparent, for example by being susceptible to inflection, e.g. *picking*, by providing a segmental alternative to contiguous but MWE *pick up* $\cdots$, which would need a morphological pointer for inflection, as noted by [28, 33] for their analyses. Nunberg, Sag and Wasow's [25] dichotomy between phrasal idioms and idiomatically combining items also vanishes, because of the singleton types and head-word subcategorized argument types. The distinction between syntactically pseudo-active *kick the bucket* and more active *spill the beans* naturally follows from whether the idiomatic part has a role in the predicate-argument structure, which we capture by systematically choosing between option (4c/i) and (4c/ii) per lexical correspondence.

3.1   Parsing and Correspondence with Singleton Types

The crucial property of a category in a lexical correspondence such as $\alpha := A/\text{"}s\text{"}$ with singleton s, is that the string "s" *as a category* does have its own correspondence. This cannot be a literal match without categorial processing of the surface string, with s to the right of α. It is a compositional derivational process arising from (a) below, to lead to (b). The lexically specifiable difference from a polyvalent category such as NP, VP is that the item α subcategorizes for the string s, hence treat it as a category, rather than subcategorize for the category of s, viz. B in the example. To obtain B, the derivational process works as usual for s, independent of the item α. We shall see in (9) that rules of function application need no amendment for this interpretation. (8b) is lexically determined by α.

---

[3] van der Linden [22], which is another categorial approach to idioms, allows vacuous abstractions, i.e. define semantics without mention of x in the LF of (7b). Apart from our empirical claim that they have a place in LF because they relate to contingency, vacuous abstractions seem to open ways to resource insensitivity which is unheard of in natural language; for example, the **K** combinator with its vacuous abstraction $\lambda x \lambda y.x$ can delete things from LF. We have yet to find a word or morpheme that does this; see [5]:81 for some speculation.

[22]'s treatment of phrasal idioms such as *kick the bucket* assumes partial involvement of the head verb *kick* for the semantics of the idiom, whereas in our conception it is fully responsible for the idiom with the aid of singleton types.

(8) a. $s := B{:}s'$

  b.
$$\frac{A/\text{``}s\text{''}{:}\lambda x.p_x \quad B{:}s'}{\alpha s := A{:}p_{s'}}{>}$$

Same idea applies to backward application, for $\alpha := A\backslash\text{``}s\text{''}$ and the sequence $s\alpha$.

In other words, the surface string s is derived by the derivational process as well. It is just that the item α carrying the singleton type as an argument decides what to do with its semantics, which we indicated schematically above as modal contribution to contingency of p, as p_x of α. This is not post-processing of a category in a radically lexicalized grammar, in which all and only head functors decide what to do with the semantics of their arguments.

It means that, whether an argument type is polyvalent or singleton, there has to be an LF placeholder for it, otherwise the derivational process, which is completely driven by syntactic types in CCG, cannot proceed. It can be seen in the basic primitive of CCG, viz. function application:

(9) $X/Y{:}f \; Y{:}a \quad \to X{:}fa$ $(>)$
     $Y{:}a \quad X\backslash Y{:}f \to X{:}fa$ $(<)$

The LF of the functor, f, has to be a lambda abstraction, to be able to take any Y and yield fa. This is true of singleton '$/Y$' and '$\backslash Y$' too.

We can clearly see the role of substitution rather than insertion in projection of types. The rule $(>)$ above is in fact realized as below (similarly for others):

(10) $\alpha := X/Y{:}f \;\; \beta := Y{:}a \;\; \to \;\; \alpha\beta := X{:}fa$ $(>)$

There is no sense in which we can insert something into α and β as they form $\alpha\beta$ because these are surface expressions.

The singleton types present an asymmetry in argument-result (or domain-range) specification. Functors such as A/B and $A\backslash B$ have domain B and range A, and, apart from trivial identities where A and B are the same singleton, the interpretation where the range itself (A) is a singleton is problematic. Since $A|B$ is a function *into A* for some slash '$|$', if it is not a trivial case of singleton identity, say *"up"/"up"*, it is difficult to see how A can be singleton. Although there are no formal reasons to avoid singleton results, and results of results, we conjecture that singletons are arguments, and arguments of results and arguments, because there seems to be no nontrivial function of a singleton-result with grammatical significance.

A related argument can be made about a singleton's potential to be the overall syntactic category of a lexical item. The notion of extending the phonological range of an item such as (a) below coincides naturally with "words with spaces" idea (e.g. *ad hoc, by and large, every which way*), which is common in NLP of MWEs, but (b) is also an option.

(11) a.   every which way $:= (S\backslash NP)\backslash(S\backslash NP){:}\lambda p \lambda x.omni'px$
     b.   every which way $:= \text{``}every\;which\;way\text{''}{:}omniway'$

Notice that (b) is different than having *scored* $:= (S\backslash NP)/\text{``}every\,which\,way\text{''}$ for lexically specified verbal adjunction in the manner of [13], which, given (8), must either use entries similar to (11), or derive *every which way* syntactically, and choose to trump its category because it wants a narrower LF due to singleton subcategorization. However we think that both options may be redundant, because of the following.

In CCG the head functor decides the semantics of its entry even if it subcategorizes for a singleton category. Therefore the entries in (a–b) above which we use in (a-b) below *may* be redundant if the words in "words with spaces" are part of the grammar, and if they can combine in any way, say as in (c) below for some *A, B, C*:

(12) a.

My team	scored	every which way
NP	$(S\backslash NP)/$"*every which way*"	$(S\backslash NP)\backslash(S\backslash NP)$

$$S\backslash NP \quad {}_{>}$$

b.

scored	every which way
$(S\backslash NP)/$"*every which way*"	"*every which way*"

$$S\backslash NP \quad {}_{>}$$

c.

scored	every which way
$(S\backslash NP)/$"*every which way*"	A/B B/C C

$$B \quad {}_{>}$$
$$A \quad {}_{>}$$
$$S\backslash NP \quad {}_{>}$$

There would be no post-processing or reanalysis in these cases; they would be multiple analyses because of redundancy. The transparency of derivation requires that in configurations like (8b) the constituents of the rule applying can themselves be derived.

The rules that allow CCG to rise above function application in projection, composition and substitution also maintain the transparency of the syntactic process, by being oblivious to the nature of argument types in these rules:[4]

(13) $\quad X/Y\!:\!f \quad Y/Z\!:\!g \rightarrow X/Z\!:\!\lambda x.f(gx)$ $\hspace{4cm}$ $(>\mathbf{B})$
$\quad X/Y/Z\!:\!f \ Y/Z\!:\!g \rightarrow X/Z\!:\!\lambda x.fx(gx)$ $\hspace{3.6cm}$ $(>\mathbf{S})$

If the result categories are not singletons, as we argued, then the rules above never face a case where Y is a singleton. This means that, since singletons are arguments, meaning they bear a slash, say '$|A$' for some slash '$|$' in $\{\backslash, /\}$, the slash is inherently application-only, equivalently '$|_\star A$' in [3] terminology.[5]

This is corroborated by examples like below where there is no idiomatic reading: (We show the derivation for the hypothetical case where singletons would be allowed

---

[4] We show only one directional variant of each rule for brevity. The same idea applies to all variants; see Steedman references for a standard set of rules, and [5] for review of proposals for combinatory extensions.

Bozşahin [5]:§10 shows that all projection rules of CCG can be packed into one monad to enable monadic computation with just one rule of projection. This is possible because CCG is radically lexicalized in the sense that combinatory rules cannot project anything which is not in the lexicon. What appears to be rule choice when presented as (9/13) becomes dependency passing within monad with one rule of combination.

[5] The way this is implemented in many CCG systems including ours is for example to constrain the slashes as follows:

$\quad X/_\star Y\!:\!f \ Y\!:\!a \quad \rightarrow X\!:\!fa$ $\hspace{5cm}$ $(>)$
$\quad X/_\diamond Y\!:\!f \ Y/_\diamond Z\!:\!g \rightarrow X/_\diamond Z\!:\!\lambda x.f(gx)$ $\hspace{3.5cm}$ $(>\mathbf{B})$

It is easier to describe slash-modal control from the perspective of syntactic types of expressions accessing these rules. '$\star$-rules' are accessible by all categories, '$\diamond$' and '$\times$' are compatible only with themselves, and with the most permissive slash.

to compose. Typing the singleton as '$/_\star$"*the bucket*"' eliminates the derivation. The slashes in the paper are harmonic '$\backslash_\diamond$' or '$/_\diamond$' unless stated otherwise.)

(14)

♯John kicked	and	Mary	did	not	kick
$S/$"*the bucket*"	$(X\backslash_\star X)/_\star X$	$S/(S\backslash NP)$	$(S\backslash NP)/VP_{\text{inf}}$	$VP_{\text{inf}}/VP_{\text{inf}}$	$VP_{\text{inf}}/$"*the bucket*"

$$\underline{\quad\quad\quad\quad\quad}_{>\mathbf{B}}$$
$$S/VP_{\text{inf}}$$
$$\underline{\quad\quad\quad\quad\quad\quad\quad\quad\quad}_{>\mathbf{B}}$$
$$S/VP_{\text{inf}}$$
$$\underline{\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad}_{>\mathbf{B}}$$
$$S/\text{"the bucket"}$$
$$\underline{\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad}_{\&}$$
$$S/\text{"the bucket"}$$

the bucket
NP

For polyvalent types, one-to-one correspondence of syntactic types and placeholder types is meant to capture the thematic structure in CCG, for example for *the door opened* versus *someone opened the door*, by having two different (albeit related) correspondences for *open*.

For a singleton, its functor (and there must be one, since they can only be arguments) decides lexically whether there is a predicate-argument structural role for the placeholder in the LF, as we see in the distinction of *spill the beans*, where *secret'* is an argument of *divulge'*, versus *kick the bucket*, where *bucket'* or anything related to it is not an argument of *die'*.

Therefore, for CCG, MWEs and phrasal idioms are not exceptions that need non-transparent derivation, apart from lexical specification as something special. They are consequences of the nature of categories and radical lexicalization.

Also because of the properties described in this section, a string as a category cannot be empty, which would violate CCG's principle of adjacency and principle of transparency (see Steedman references). No rule in (9) or (13) can apply if one of the categories is empty. Therefore the surface string itself for the singleton (*s* in example (8)) cannot be empty either.

Having explored the possibilities for the singleton types in combinatory categories, we look at their use.

3.2 Verb-particles and Phrasal Idioms with Singleton Types

In verb-particle constructions, the differences in the syntax-semantics correspondence force the following lexical distinctions. We now write the categories in more detail than in the preview.

(15) a. picked := $(S\backslash NP)/$"*up*"$/NP_{\text{-heavy}} : \lambda y \lambda x \lambda z.cause'(init'(hold_x'yz))z$
      b. picked := $(S\backslash NP)/NP_{\text{+lexc}}/$"*up*"$: \lambda x \lambda y \lambda z.pick_x'yz$
      c. picked := $(S\backslash NP)/NP : \lambda x \lambda y.pick'xy \wedge choose'xy$

The features above are all finite-state computable, just like morphological ones, as phonological weight ($\mp$heavy) and lexical content ($\mp$lexc) in an expression substituting for a category. All CCG category features can be interpreted this way, because combinators do all the syntactic work.

The reason for having two different grammar entries (a–b) for *pick up* follows from the fact that they are not equally substitutable, for example as an answer to *What did you do?*

(15b) leads to achievement, and (15a) to culmination. Both cases also differ from (c), which provides wider substitution for *NP*, and with a different meaning. We treat (a–c) distinctions surface-compositionally, which are transparently projected without wrap:

(16)

I	picked	the book	up
NP_{1s}	$(S\backslash NP)/\text{``up''}/NP_{\text{-heavy}}$	NP	$((S\backslash NP)\backslash(S\backslash NP))/NP$
$:i'$	$:\lambda y\lambda x\lambda z.cause'(init'(hold'_{xyz}))z$	$:def'book'$	$:\lambda x\lambda p\lambda y.up'(py)x$

$$\frac{(S\backslash NP)/\text{``up''}}{:\lambda x\lambda z.cause'(init'(hold'_x(def'book')z))z}\,{}^{>}$$

$$\frac{S\backslash NP}{:\lambda z.cause'(init'(hold'_{\lambda x\lambda p\lambda y.up'(py)x}(def'book')z))z}\,{}^{>}$$

$$\frac{S}{:cause'(init'(hold'_{\lambda x\lambda p\lambda y.up'(py)x}(def'book')i'))i'}\,{}^{<}$$

where *hold'* at the end of the derivation can interpret its event modality (contingency) compositionally, since it is a closed lambda term.

Notice that the word *up* knows nothing about the verb-particle construction. Its category is for a PP head, say PP_{up}, as a predicate modifier. It is the verb that delivers the distinct meaning. Its subcategorization is for a singleton, which eschews the syntactic category of the word *up* but not its phonology and semantics, as described in (8b).

(15b) can be assumed to arise from the syntactic category $VP/NP_{\text{+lexc}}/\text{``up''}$ by finite inflection. CCG has options here, to accommodate morphology without having to have a "morphological insertion point" in a contiguous but MWE entry *pick up* := $VP/NP_{\text{+lexc}}$, to avoid ?*pick upped*.[6] This is made possible by singleton types.

Examples (15a–b) use a degree of freedom which is relevant to phrasal idioms. The singleton syntactic type "up" corresponding to the LF placeholder *x* maintains the compositionality of the correspondence; but, it may have no contribution to the predicate-argument structure at all in some cases, which would make it paracompositional, because its semantic type is a closed lambda term as far as predicate-argument structure is concerned. Notice that in (8b), s' is not in the predicate-argument structure of *p*; it is a contingency of *p*.

Consider the following examples in this regard, where *x* for *bucket'* as an event modality might mean 'ceremonial death', 'reported death', etc.:
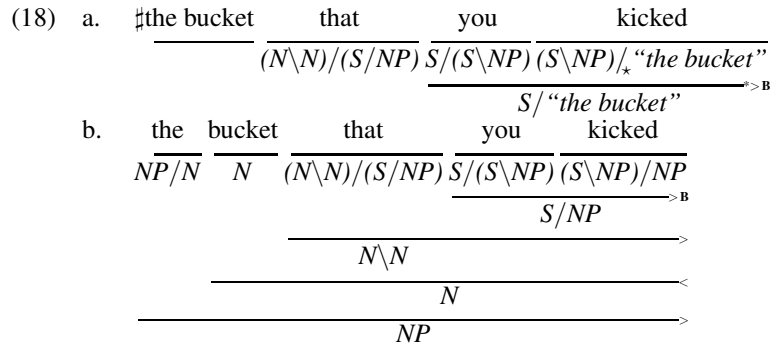
(17) a. kicked := $(S\backslash NP)/_{\star}\text{``the bucket''}$: $\lambda x\lambda y.die'_x y$
      b. kicked := $(S\backslash NP)/NP$: $\lambda x\lambda y.kick'xy$

They anticipate very limited syntax in the semantically paracompositional part in the idiom reading (a) because of having to enumerate them (*kick the old/proverbial bucket* vs *kick the bucket that John thought overflowed*).[7] These assumptions cannot give rise to the idiom reading in *the bucket that you kicked*, with no further stipulation

---

[6] The fact that this form is also attested in child and adult language suggests that these entries may be bonafide lexical options.

[7] It is tempting to try $NP_{\text{proverbial bucket}}:proverb'death'$ for *kick the proverbial bucket* which is a head-subcategorizing category; but, we would have to overextend ourselves to eliminate the idiom reading in *kick the proverbial bucket that overflowed* if we have to. In this sense we suggest that phrasal idioms are best treated with singleton types.
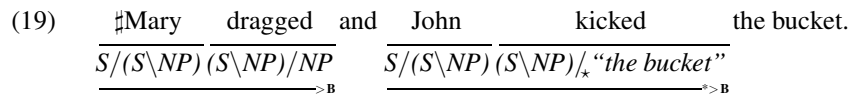
than singleton categories in a lexical entry (cf. a–b; '*' on the right of a derivation means it is not possible):

(18) a. ♯the bucket   that   you   kicked
$$\frac{\qquad\qquad\quad \overline{(N\backslash N)/(S/NP)}\ \ \overline{S/(S\backslash NP)}\ \ \overline{(S\backslash NP)/_\star\text{``the bucket''}}}{\qquad\qquad\qquad\qquad\quad S/\text{``the bucket''}}{}_{*>\mathbf{B}}$$

  b. the bucket   that   you   kicked
$$\frac{\overline{NP/N}\ \ \overline{N}\ \ \overline{(N\backslash N)/(S/NP)}\ \dfrac{\overline{S/(S\backslash NP)}\ \ \overline{(S\backslash NP)/NP}}{S/NP}{}_{>\mathbf{B}}}{}$$
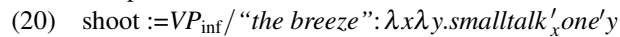$$\frac{N\backslash N}{\dfrac{N}{NP}{}_{>}}{}_{<}$$

Given the polyvalent argument category of the relative pronoun, we can see that relativization out of phrasal idioms would not be possible even if we allowed composition of singleton types, therefore the syntactic productivity of idiomatically combining phrases arises from their use of head-dependencies rather than singletons, as we shall soon see in derivations similar to (b), in (26).

We note that carrying the head-word in a polyvalent category to have the same effect, for example $kick := (S\backslash NP)/_\star NP_\text{bucket}$, overgenerates the idiom reading, because *the bucket that John thought overflowed* can substitute for NP_bucket.
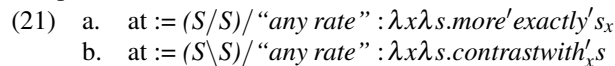
The direct approach to categories that we see in radically lexicalized grammars, whether they are polyvalently substitutable or not, contrasts with systems of rewrite and/or record keeping in which post-processing is possible. For example there is no reanalysis or post-processing mechanism needed to eliminate the idiomatic reading below:

(19)   ♯Mary   dragged   and   John   kicked   the bucket.
$$\frac{\overline{S/(S\backslash NP)}\ \ \overline{(S\backslash NP)/NP}}{}{}_{>\mathbf{B}}\qquad \frac{\overline{S/(S\backslash NP)}\ \ \overline{(S\backslash NP)/_\star\text{``the bucket''}}}{}{}_{*>\mathbf{B}}$$

We can then follow [32] in assuming that passive is a polyvalent lexical process headed by the passive morpheme, mapping for example VP_inf/NP to VP_pass, which eliminates passivization **the breeze was shot* from the entry:

(20) shoot $:=VP_\text{inf}/\text{``the breeze''}:\lambda x \lambda y.smalltalk'_x one' y$

Idioms such as *at any rate, beside the point* further demonstrate that all idioms needing restricted types must contain a predicative element in the domain of locality of their head because we are required by paracompositionality to record the special reading and contingency, for example as extension of discursive clarification (a) and comparison (b):

(21) a. at $:= (S/S)/\text{``any rate''}:\lambda x \lambda s.more' exactly' s_x$
     b. at $:= (S\backslash S)/\text{``any rate''}:\lambda x \lambda s.contrastwith'_x s$

4 Head-word Subcategorization and Idioms

The difference between idiomatically combining phrases and phrasal idioms such as kicking the bucket is clear: The syntactically active ones are active because the idiomatic part has a role in the predicate-argument structure. 'Secret' is an argument of

'divulge', whereas 'bucket' is not an argument of 'die'. For example, *spill the beans* seems to require categorization such as (a) below in the manner of (6), rather than (b) fashioned from (5) or singleton-subcategorizing (c). Cf. also the non-idiomatic *spill* in (d). Tense morphology renders finite versions of VP_{inf} below as $S\backslash NP$, eg. *spilled*:=$(S\backslash NP)/NP_{beans}$ for (a).

(22)　a.　spill := VP_{inf}/NP_{beans}: $\lambda x\lambda y.divulge'_x secret' y$

　　　b.　spill := VP_{inf}/NP : $\lambda x\lambda y.$if $head(x) = beans'$then $divulge'_x secret' y$
　　　　　　　　　　　　　　　else $spill' xy$

　　　c.　spill := $(VP_{inf}/$"beans"$)/PredP$: $\lambda p\lambda x\lambda y.divulge'_{px} secret' y$

　　　d.　spill := VP_{inf}/NP: $\lambda x\lambda y.spill' xy$

PredP is a predicative phrase type, which includes the quantifier phrase. The syntactic type of the idiomatic argument in (a) encodes the head-dependency from surface structure. It avoids the idiomatic reading in *to spill the bean*, which (b) may not. (b)-style solutions would depend on LF objects, which may not always reflect surface forms in full. In fact (b) requires post-processing to eliminate the idiom reading in the following example:

(23)♯You　　　spilled　　　　　and　　　　Mary　　cooked　　the beans

$\dfrac{\overline{S/(S\backslash NP)}\ \overline{(S\backslash NP)/NP}\ \overline{(X\backslash_\star X)/_\star X}\ \overline{S/(S\backslash NP)}\ \overline{(S\backslash NP)/NP}\ \overline{NP_{beans}}}{}$

$:\lambda p.p you'\ :\lambda x\lambda y.if\cdots$ $:\lambda p\lambda q\lambda z.and'(pz)(qz)$ $:\lambda p.p m'$ $:\lambda x\lambda y.cook' xy$ $:def' beans'$

$\dfrac{S/NP}{:\lambda x.\text{if } head(x)=\cdots}{}_{>B}$　　　　　$\dfrac{S/NP}{:\lambda x.cook' xm'}{}_{>B}$

$\dfrac{S/NP}{:\lambda z.and'(\text{if } head(z)=\cdots)(cook' zm')}{}_\&$

$\dfrac{S/NP:and'(\text{if } head(def' beans')=}{beans'\text{ then } divulge' secret' you'\cdots)(cook'(def' beans') m')}{}_>$

This is still the case if we treat the construction as multi-headed, as [15]:238 do, by also assuming *the beans* := NP_{beans}:$secret'$, and changing the LF choice condition of *spill* to 'if $head(x)=secret'$ then $divulge' xy$ else $spill' xy$'. *Cook'* does not refer to this entry.

　　　The process of marking head-word dependencies requires statistical learning, as the category such as NP_{beans} in (22a) implies. It has been known in TAG systems with supertags since [6] that disambiguating such categories is feasible with training. The earliest approach to such marking in CCG is [8, 9] as far as we know, where probabilistic CCGs are similarly trained. Later work such as [1] shows further progress in disambiguation of head-dependencies.

　　　NP_{beans} is a polyvalent type, not a singleton. Therefore we get the following accounted for by (22a) (some of the examples are from [33]):
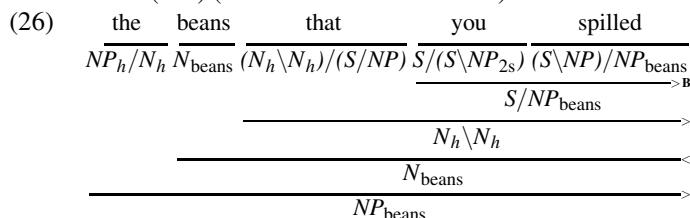
(24)　a.　spill /several/the musical/the artistic/mountains of/loads of/ beans

　　　b.　spill the beans no one cares about

Head-marking of an argument category by the idiom's head is required because of examples such as below, where an idiomatic reading is eliminated despite relatively free syntax because the coordinands would not be like-typed:

(25)　　♯You　　　　spilled　　　　and　　　Mary　　cooked　　the beans

$\dfrac{\overline{S/(S\backslash NP)}\ \overline{(S\backslash NP)/NP_{beans}}\ \overline{(X\backslash_\star X)/_\star X}\ \overline{S/(S\backslash NP)}\ \overline{(S\backslash NP)/NP}\ \overline{NP_{beans}}}{}$

$\dfrac{S/NP_{beans}}{}{}_{>B}$　　　　　　　$\dfrac{S/NP}{}{}_{>B}$

$\dfrac{}{}{}_{*\&}$

Right-node raising succeeds when non-idiomatic entries such as (22d) do not subcate-gorize for head-word marked arguments. (25) is unproblematic with it.

When the head of the construction does not require identical types as does the con-junction above, head-projection works with simple term match; cf. the one for kicking the bucket in (18a) (h is for head-word feature):

(26)

the	beans	that	you	spilled
NP_h/N_h	N_{beans}	$(N_h\backslash N_h)/(S/NP)$	$S/(S\backslash NP_{2s})$	$(S\backslash NP)/NP_{\text{beans}}$

$$\frac{\qquad S/(S\backslash NP_{2s}) \quad (S\backslash NP)/NP_{\text{beans}}\qquad}{S/NP_{\text{beans}}}{}^{>\mathbf{B}}$$

$$\frac{(N_h\backslash N_h)/(S/NP) \qquad S/NP_{\text{beans}}}{N_h\backslash N_h}{}^{>}$$

$$\frac{N_{\text{beans}} \qquad N_h\backslash N_h}{N_{\text{beans}}}{}^{<}$$

$$\frac{NP_h/N_h \qquad N_{\text{beans}}}{NP_{\text{beans}}}{}^{>}$$

The example also shows that argument types of idiomatically combining phrases must be composable; therefore; (22c) is inadequate.[8]

5 Idioms Requiring a Combined Approach

There seems to be cases where a combination of singletons and head-marked poly-valent subcategorization is needed. The *give creeps* construction, which is sometimes considered not an idiom because of its compositionality [19], is paracompositional in our sense, and idiomatically combining in [25] terminology, because although *creeps* seems to be an event modality of *revulse'* rather than its argument, *fear'* is an argument. A simple head-marking approach such as '$/NP_{\text{creeps}}$' would overgenerate in cases such as ♯*give me some creeps*, but we have *give me the absolute/shivering/full-on creeps*. No-tice also that the construction and related items resist dative shift (judgments are from [20]; '*' seems to be equivalent to '♯' in our terms):

(27)   a.   The Count gave me the creeps./ *The Count gave the creeps to me.
       b.   His boss gave Max the boot./ *His boss gave the boot to Max.

Richards [27] observes that (a) below can be the unaccusative of *give*; and, (b) is widely attested in the web (but recall ♯*give me some creeps*).

(28)   a.   Mary got the creeps.
       b.   give some creeps
       c.   give := $VP/N_{\text{creeps}}/\text{``the''}/NP : \lambda x \lambda y \lambda z \lambda w.cause'(init'(revulse'_z fear'_y x))w$

Assuming that dative shift is polyvalent, following [32], in the form of lexical mapping from $VP/NP/NP$ to $VP/PP_{\text{to}}/NP$, we can eliminate it for the type in (c), which we think captures the insight of Richards, and permits adjunction within an N, e.g. *mountains of creeps*.

Another class of idioms forces a combined approach as well. Semantic reflexives in *I twiddled my thumbs/ate my words/racked my brain/lose my mind* are not morphologi-cal reflexives and they are inherently possessive, for example:

---

[8] One way to put it altogether is to use a feature such as ∓special in addition to h, which or-dinary verbs negatively specify, heads of idiomatic combination positively specify, and heads of syntactic constructions eg. coordinators and relative markers (under)specify as they see fit. The value '+special' need not be further broken down for singletons because they are self-representing, and, presumably, featureless. For example phonological weight is intrinsically captured in *"the beans"*; also, lexical content.

(29)   twiddled := $(S \backslash NP_{\mathrm{agr}}) / \text{"thumbs"} / NP_{\text{-lexc,+poss,agr}}$
$$: \lambda x \lambda y \lambda z. pass'_y time'_{(self'_z)} z \wedge inalien'(xyz)$$

The LF captures the properties that the subject idles on his own time, the lexical possessive in the LF of *x* which is presumably lexically *poss'* is inalienable and belongs to the subject. This is a reflexive in the sense that it must be bound in its local domain determined by *pass.'* The referent (*z*) is available in one domain of locality in a radically lexicalized grammar because the head of the idiom does not require a VP in phrase-structure sense but a clause. Agreement is locally available too; by insisting on same agreement features. The head-dependency is that the argument does not contain lexical material, leaving out examples such as *John twiddled John's thumbs* as an idiom.

6   No Wrap

We have seen that options (4c/i) and (4c/ii) are not mutually exclusive. We also suggested that singleton type is a forced move to avoid loss of meaning composition. One consequence of this is the treatment of verb-particles without wrap, which are not related to idioms although they are MWEs. We now consider option (4b) in more detail from this perspective, which at first sight seems to be just as lexical as the two alternatives we have considered so far.

The projection principle of CCG, which says that lexical specification of directionality and order of combination cannot be overridden during derivations, eliminates (30) from projection because it has the second-combining argument (*Y*) of a function applying before its first-combining argument (*Z*), an operation of the general class that has been proposed in other categorial approaches under the name of "wrap."

(30) $(X/Y)/Z{:}f \ Y{:}a \rightarrow X/Z{:}\lambda z.fza$ (*)

Wrap of the kind in (30) has a combinatory equivalent, namely Curry's combinator **C** (see [11]). CCG's adjacency principle eliminates this combinator on empirical grounds, rather than formal, as a freely operating rule. Adding (30) to CCG's projection has the effect of treating VSO and VOS as both grammatical, which is not the case for Welsh, and to carry the same meaning, which is not the case for Tagalog although both VSO and VOS are fine. These properties must be part of a lexicalized grammar rather than syntactic projection.

The version of wrap which [2, 12, 16] employ is different, which was eliminated from consideration so far because it is non-combinatory; and, it violates adjacency of functors and arguments. That wrap is the following:

(31)
$$\frac{\dfrac{s_1}{X/_W Y{:}f} \quad \dfrac{s_2}{Y{:}a}}{first(s_1)\,s_2\,rest(s_1) := X{:}fa}\text{wrap}$$

where *first*() function gives the first element in a list of surface expressions for Bach [2], or first word for Dowty [12]; and, *rest*() returns the rest of the expression. The wrapping slash '$/_W$' of Jacobson [16] does the infixation of s_2.
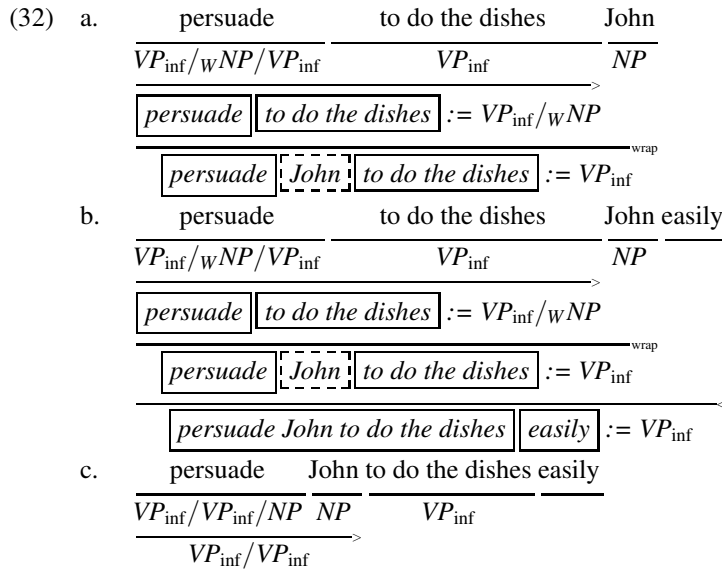
Semantically, it is function application. Syntactically, no combinator can do what this rule does to its input expressions, which is to rip apart one surface expression (s_1)

and insert into it. It differs from **C**, which wraps one independent expression in *two* independent expressions.

The appeal of surface wrap to MWEs was to be able to write a category for *pick* $\cdots$ *up* as for example $pick := (S \backslash NP)/_W NP/P_{up}: \lambda x \lambda y \lambda z.pick_x' yz$; cf. (16).

Syntactic wraps such as above, whether combinatory or non-combinatory, have domino effects on dependency and constituency, unlike 'lexical wrap', where a lexical entry specifies its correspondence; for example, for the strictly VSO Welsh verb $gwelodd := (S/NP)/NP_{3s}: \lambda x \lambda y.saw' yx$; note the LF.

An example of global complications in grammar caused by wrap can be seen below, where dashed boxes denote wrapped-in material; cf. Figure 1.

(32) a.

persuade	to do the dishes	John
$VP_{inf}/_W NP/VP_{inf}$	VP_{inf}	NP

$\dfrac{}{}$ >

$\boxed{persuade} \ \boxed{to\ do\ the\ dishes} := VP_{inf}/_W NP$

$\boxed{persuade} \ \dashbox{John} \ \boxed{to\ do\ the\ dishes} := VP_{inf}$ ^{wrap}

b.

persuade	to do the dishes	John easily
$VP_{inf}/_W NP/VP_{inf}$	VP_{inf}	NP

>

$\boxed{persuade} \ \boxed{to\ do\ the\ dishes} := VP_{inf}/_W NP$

$\boxed{persuade} \ \dashbox{John} \ \boxed{to\ do\ the\ dishes} := VP_{inf}$ ^{wrap}

<

$\boxed{persuade\ John\ to\ do\ the\ dishes} \ \boxed{easily} := VP_{inf}$

c.

persuade	John	to do the dishes easily
$VP_{inf}/VP_{inf}/NP$	NP	VP_{inf}

$\dfrac{}{VP_{inf}/VP_{inf}}$ >

Derivation (a) is Bach's use of non-combinatory wrap rule in (31). Given these categories which involve wrap, there is one interpretation for (b), where the adverb can only modify *persuade*. With the unwrapped version of *persuade* in (c), two interpretations are possible: one modifies the VP complement of *persuade*, and the other, *persuade John*, both of which are required for adequacy.

7 Conclusion

One point of departure of CCG from other categorial grammars and from tree-rewrite systems is that (i) we can complicate the basic vocabulary of the theory, but (ii) not its basic mechanism such as introducing wrap, if a better explanation can be achieved. The first point has been made by Chomsky repeatedly since [7]:68. Singleton types could be viewed as one way of doing that, much like $S \backslash NP$ vs. VP distinction. We have argued that it is actually not a complication at all in CCG's case, because the possibility has been available, in the notion of type as a set of values, which can be a singleton set. CCG differs from Chomskyan notion of category substitution by eliminating *move*, empty categories and lexical insertion altogether, which means that all computation is

local, type-driven, and there is no action-at-a-distance, to address the second point. The expressions substituting for these types are then locally available in the course of a derivation. This seems critical for MWEs.

The possibility of a singleton value is built-in to any type. The asymmetry of CCG's singletons' categorization, that they can be arguments, and arguments of arguments and results, and, their inherent applicative nature, deliver MWEs and phrasal idioms as natural consequences rather than stipulation or a "pain in the neck for NLP." Syntactically active idioms are not singleton-typed because they have relevance to predicate-argument structure; and, their narrower syntax, compared to free syntax, seems to necessitate head-marking of some argument categories, which is known to be probabilistically learnable.

Some implications of our analyses are that all idioms can be made compositional at the level of a lexical correspondence without losing semantic distinctions, and without meaning postulates or reanalysis. Categorial post-processing of MWEs and phrasal idioms, and multi-stage processing of them in the lexicon, as done by [10, 33], may be unnecessary if we assume type substitution to be potentially having one value, and surface head-marking to be an option for polyvalent argument types. One conjecture is that any idiom in any language has to involve a predicate implicated by some predicative element in the expression to keep the meaning assembly paracompositional.

The analyses in the article can be replicated by running the CCG tool at `github.com/bozsahin/ccglab`. The particular fragment in the chapter is at `github.com/bozsahin/ccglab-grammars/cb-ag-fg2018-grammar`.

Bibliography

[1] Artzi, Y., Lee, K., Zettlemoyer, L.: Broad-coverage CCG semantic parsing with AMR. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1699–1710. Lisbon, Portugal (2015)

[2] Bach, E.: An extension of Classical Transformational grammar. In: Problems in Linguistic Metatheory: Proceedings of the 1976 Conference at Michigan State University. pp. 183–224. Michigan State University, Lansing, MI (1976)

[3] Baldridge, J., Kruijff, G.J.: Multi-modal Combinatory Categorial Grammar. In: Proceedings of 11th Annual Meeting of the European Association for Computational Linguistics. pp. 211–218. Budapest (2003)

[4] Bond, F., Ho, J.Q., Flickinger, D.: Feeling our way to an analysis of english possessed idioms. In: Müller, S. (ed.) Proceedings of the 22nd International Conference on Head-Driven Phrase Structure Grammar. pp. 61–74. CSLI Publications, Stanford, CA (2015)

[5] Bozşahin, C.: Combinatory Linguistics. De Gruyter Mouton, Berlin (2012)

[6] Chen, J., Bangalore, S., Vijay-Shanker, K.: New models for improving supertag disambiguation. In: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics. pp. 188–195. Association for Computational Linguistics (1999)

[7] Chomsky, N.: Some empirical issues in the theory of transformational grammar. In: Peters, S. (ed.) Goals of Linguistic Theory. Prentice-Hall, Englewood Cliffs, NJ (1972)

[8] Clark, S., Curran, J.R.: Wide-coverage efficient statistical parsing with CCG and log-linear models. Computational Linguistics 33(4), 493–552 (2007)

[9] Clark, S., Hockenmaier, J., Steedman, M.: Building deep dependency structures with a wide-coverage CCG parser. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. pp. 327–334 (2002)

[10] Copestake, A.: Representing idioms. In: HPSG Conference. Copenhagen (1994)

[11] Curry, H.B., Feys, R.: Combinatory Logic. North-Holland, Amsterdam (1958)

[12] Dowty, D.: Dative movement and Thomason's extensions of Montague Grammar. In: Davis, S., Mithun, M. (eds.) Linguistics, Philosophy, and Montague Grammar, pp. 153–222. University of Texas Press, Austin (1979)

[13] Dowty, D.: The dual analysis of adjuncts/complements in categorial grammar. pp. 33–66 (2003), in [18]

[14] Everaert, M., Van der Linden, E.J., Schreuder, R., Schenk, A. (eds.): Idioms: structural and psychological perspectives. Lawrence Erlbaum, New Jersey (1995)

[15] Gazdar, G., Klein, E., Pullum, G., Sag, I.: Generalized Phrase Structure Grammar. Harvard University Press (1985)

[16] Jacobson, P.: Flexible categorial grammars: Questions and prospects. In: Levine, R. (ed.) Formal Grammar, pp. 129–167. Oxford University Press, Oxford (1992)

[17] Kay, P., Sag, I.A., Flickinger, D.: A lexical theory of phrasal idioms (ms), available at: www1.icsi.berkeley.edu/∼kay/idiom-pdflatex.11-13-15.pdf

[18] Lang, E., Maienborn, C., Fabricius-Hansen, C.: Modifying adjuncts. Walter de Gruyter (2003)

[19] Larson, R.: On the double object construction. Linguistic Inquiry 19, 335–392 (1988)

[20] Larson, R.: On "dative idioms" in English. In: Workshop on Syntax-semantics. Fuji Women's University (2012)

[21] Lichte, T., Kallmeyer, L.: Same syntax, different semantics: A compositional approach to idiomaticity in multi-word expressions. In: Piñón, C. (ed.) Empirical Issues in Syntax and Semantics, vol. 11, pp. 111–140. CSSP, Paris (2016)

[22] van der Linden, E.J.: Incremental processing and the hierarchical lexicon. Computational linguistics 18(2), 219–238 (1992)

[23] Manning, C.D.: Ergativity: Argument Structure and Grammatical Relations. CSLI, Stanford, CA (1996)

[24] Moens, M., Steedman, M.: Temporal ontology and temporal reference. Computational Linguistics 14, 15–28 (1988), reprinted in Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas (eds.) *The Language of Time: A Reader*. Oxford University Press, 93-114.

[25] Nunberg, G., Sag, I.A., Wasow, T.: Idioms. Language 70(3), 491–538 (1994)

[26] Pulman, S.G.: The recognition and interpretation of idioms. In: Cacciari, C., Tabossi, P. (eds.) Idioms: Processing, structure, and interpretation, pp. 249–270. Lawrence Erlbaum, Hillsdale, NJ (1993)

[27] Richards, N.: An idiomatic argument for lexical decomposition. Linguistic inquiry 32(1), 183–192 (2001)

[28] Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword expressions: A pain in the neck for NLP. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 1–15. Mexico City (2002)

[29] Schuler, W., Joshi, A.: Tree-rewriting models of multi-word expressions. In: Proceedings of the ACL Workshop on Multiword Expressions: from Parsing and Generation to the Real World. pp. 25–30. ACL, Portland, OR (2011)

[30] Steedman, M.: The Syntactic Process. MIT Press, Cambridge, MA (2000)

[31] Steedman, M.: Taking Scope: The Natural Semantics of Quantifiers. MIT Press, Cambridge, MA (2012)

[32] Steedman, M., Baldridge, J.: Combinatory Categorial Grammar. In: Borsley, R., Börjars, K. (eds.) Non-transformational syntax, pp. 181–224. Blackwell, Oxford (2011)

[33] Villavicencio, A., Copestake, A., Waldron, B., Lambeau, F.: Lexical encoding of MWEs. In: Proceedings of the Workshop on Multiword Expressions: Integrating Processing. pp. 80–87. Association for Computational Linguistics (2004)