Operators for Definition by Paraphrase*

Mats Rooth

1 Introduction

Define a semantic lexical entry for *want* as used in (1b) whose first argument is a property contributed by a VP complement, and which makes (1b) equivalent to (1a), assuming our established semantics for the words and constructions in (1a). Next define a semantic lexical entry for *self* as used in (2b) whose argument is the verb *admiring* and which makes (2b) equivalent to (2a). Then give a semantics for DP-embedding *want* as used in (3b) that makes (3b) equivalent to (3a), assuming our semantics for the words and constructions in that structure.

(1) a. Justin$_1$ wants himself$_1$ to be asleep.
b. Justin [$_{VP}$ wants [$_{VP}$ to be asleep]].

(2) a. yeti who$_1$ admires himself$_1$.
b. [[self- admiring] yeti].

(3) a. Justin$_1$ wants to have a duck.
b. Justin [$_{VP}$ wants [$_{DP}$ a duck]].

Finally, define a type-raising operator RAI which as used in (4b) combines with the quantified object *everybody* and the transitive verb *admires* to produce a semantics equivalent to the semantics of (4a), where the object is quantified in.

(4) a. [[everybody] 1 [Justin admires e$_1$]]
b. Justin [admires [RAI everybody]].

These are problems of *definition by paraphrase*, where one is to find a semantics for a given word that creates an equivalence with a sentence or phrase whose lexical and compositional semantics is already established. Such problems

* Thanks especially to students in my Semantics II classes at Cornell for reactions to versions of the material presented here. Thanks also to Dorit Abusch, Marcus Kracht, and Paul Dekker for comments on the manuscript.

 | DOI: 10.1163/9789004279377_005

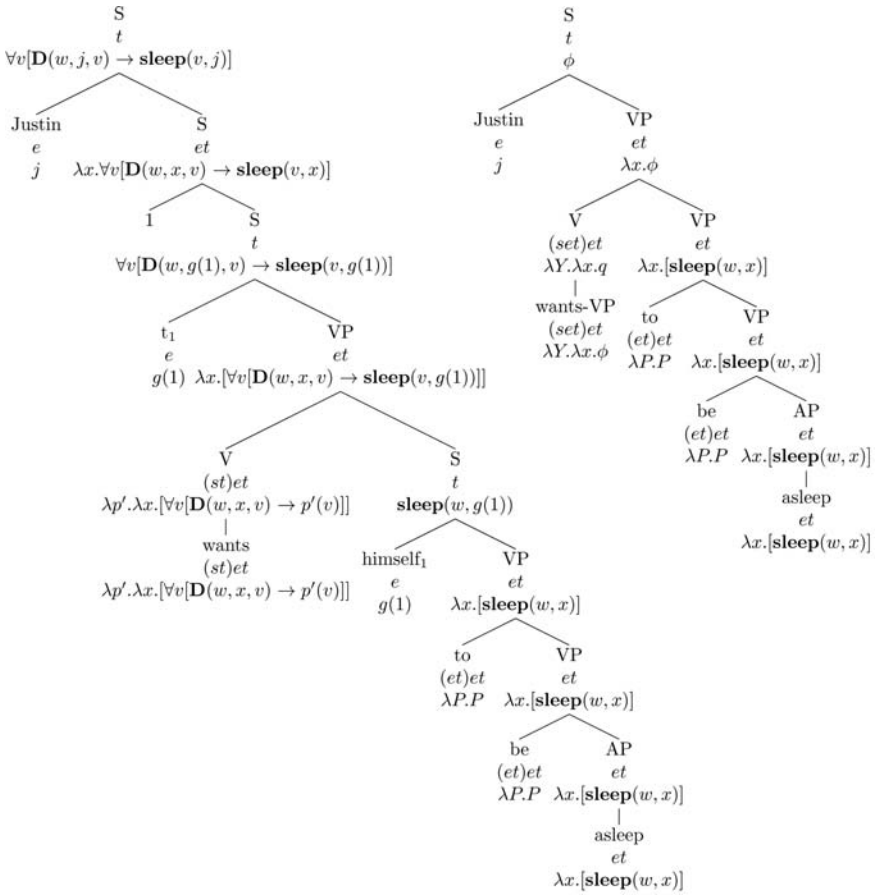

FIGURE 1 *On the left, a semantic derivation for* (1a)*. On the right, a skeletal derivation for* (1b)*.*

come up in semantics problem sets. In research, definitions by paraphrase are used as a clue to lexical-semantic primitives and compositional patterns (e.g. Zimmermann 2006), even when the paraphrases are felt to approximate. Or as in (4), a technical problem can be posed as a problem of definition by paraphrase.

Here is a marginally useful recipe for solution, illustrated for (1). On the left in Figure 1 is a textbook semantic derivation for (1a). Each node is labeled with a denotation expressed as a lambda term, and a semantic type given using Link type labels (Link 1979). The denotations are relative to a world of evaluation w and assignment function g that gives values for free variables. The expressions are built up from lambda, variables, logical vocabulary, and function primitives such as **sleep**, which is a two-place function that maps a world and an

individual to a truth value. **D** is a primitive attitude modality, with $\mathbf{D}(w, x, v)$ understood as "x's desires in world w are satisfied in world v". The semantic rules that are used are function application, variable binding, and function application after binding of the world variable for the argument—so-called intensional function application, which applies at the higher VP node (Heim & Kratzer 1998). The subject *Justin* is quantified in, in order to obtain a closed interpretation. In the formula at the top, w and v are world variables, and the semantics of (1a) with respect to a world of evaluation w is being rendered as "in any world v that satisfies Justin's desires in w, Justin is asleep." On the right in Figure 1 is a skeletal derivation for the second sentence, which shows the meaning of $want_{VP}$ as a lambda term with a dummy body ϕ. The lambda variables correspond to the complement VP (the variable Y) and the upstairs subject (the variable x). To solve the problem, we need to substitute the right thing for ϕ. It is clear that some combination of the lambda variables Y and x, the **D** relation, and logical spices are needed. This is loosely represented in (5a). Trying things out, we may arrive at (5b) as a denotation for *want-VP*, and verify as in Figure 2 that it produces the required equivalence.

(5) a. $\lambda Y \lambda x[\ldots x \ldots Y \ldots \mathbf{D} \ldots \forall \ldots v \ldots w \ldots \rightarrow \ldots]$
b. $\lambda Y \lambda x[\forall v[\mathbf{D}(w, x, v) \rightarrow Y(v)(w)]]$

This is a recipe that lists the ingredients, and then concludes abruptly with an instruction to combine them in the necessary way—something that calls for creativity or experimentation. The solution itself has an interesting positive point though. (5b) as used in Figure 2 produces a result that is not just semantically equivalent to the target, but syntactically identical to it as a lambda term. It turns out that the other problems can also be solved to give a result that is syntactically identical to the target, in the sense of being an alphabetic variant of it. This suggests that it might be possible to solve problems of definition by paraphrase syntactically, by some kind of tree-geometric manipulations and/or type-raising operators at the syntax-semantics interface. Figure 3 illustrates syntactic equivalence as lambda-terms for the *self-* problem.

2 The Scoping Method

In Figure 2, $want_{VP}$ combines with arguments *to be asleep* and *Justin* that are also present as constituents in the compositional structure for the target. Suppose that starting from the target derivation, we scope these constituents out in two steps:

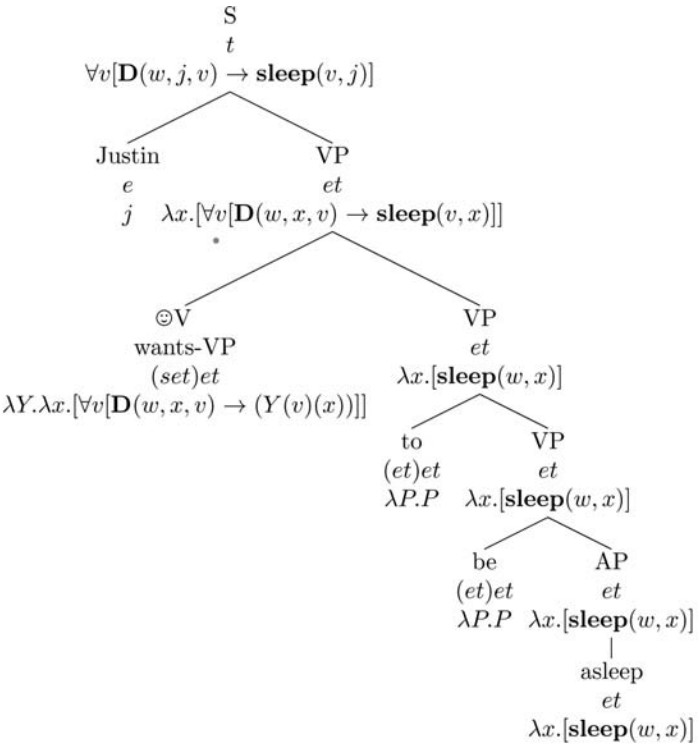

FIGURE 2 *The semantic lexical entry (5b) for* want *produces a lambda term for the semantics of (1b) that is syntactically identical to the one for the target (1a).*

(6) a. [[$_{\text{NP}}$ Justin] [⟨1, *e*⟩ [t⟨1, *e*⟩ wants [$_{\text{S}}$ himself$_1$ [$_{\text{VP}}$ to be asleep]]]]]
b. [[$_{\text{NP}}$ Justin] [[$_{\text{VP}}$ to be asleep] [⟨2, *set*⟩ [⟨1, *e*⟩ [t⟨1, *e*⟩ wants [$_{\text{S}}$ himself$_1$ t⟨2, *set*⟩]]]]

Using the intensional type *set* for the trace of the VP in the second movement will ensure that scoping out VP does not change interpretation, because the moved constituent is modally closed. Another point about the VP movement is that it is a "zippering" QR, where the landing site for movement is between the moved subject and its binding index. In this process, the derivation at the left in Figure 1 has been unzipped into a right-branching structure, putting two arguments at the top, and the corresponding binders in a second block below them. Zippering LF movement (QR) is nothing new as a tree-geometric operation—it just involves a particular choice for the target of movement. Sauerland (1998) disscussed an application of zippering QR to the semantics of cumulative predication, and Barker (2007) used it in an analysis of comparatives, in both cases

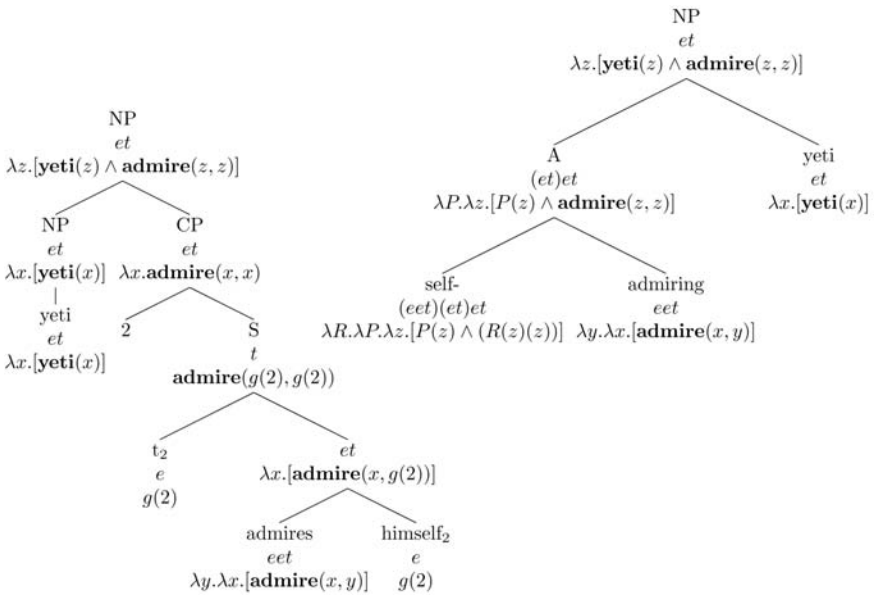

FIGURE 3 *Syntactically equivalent results in derivations for* yeti who admires himself *and* self-admiring yeti

in connection with non-standard semantic interpretations for scope. Here the semantic interpretation is the standard one.

Figure 4 is a semantic derivation for the unzipped paraphrase. As expected, we see the same result as in the the original derivation for the paraphrase. The new part is that the higher levels of the tree are isomorphic to the semantic derivation in Figure 2 for VP-embedding *want*. The two-place abstract [1 [2 [...]]] that is marked with smiley in Figure 4 is annotated with the same lambda term as the verb *wants-VP* that is marked with smiley in Figure 2. And they combine with the same arguments in the same order. In consequence the interpretations at the top in the two derivations are syntactically equivalent as lambda terms, and the interpretation of the abstract [1 [2 [...]]] can be used as a systematically derived interpretation for the verb *want-VP*, as shown in (7). One way of thinking about this is that if *want-VP* were syntactically complex and had exactly the syntax of the abstract [1 [2 [...]]], then the derivation for the VP-complement example would be isomorphic all the way down to the derivation for the unzipped paraphrase. See Figure 5.

(7) $[\![\text{want}_{\text{VP}}]\!] =_{\text{def.}} [\![[1[2[\ldots]]]]\!]$

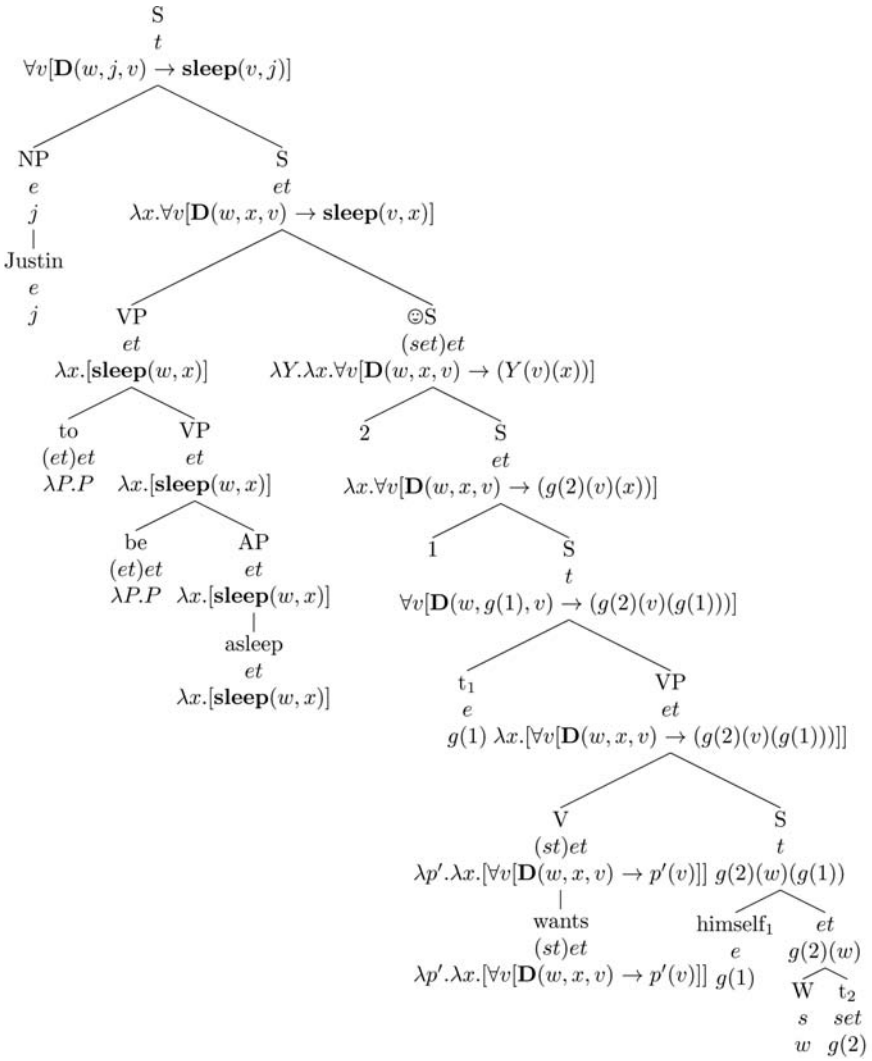

FIGURE 4 *The derivation on the left in Figure 1 unzipped. Working from the position between VP and 2 in the tree, the VP complement and matrix NP subject are above (in that order), and the corresponding binders 2 and 1 below (in that order).*

To obtain the equivalence, it is essential that unzipping the derivation for the paraphrase does not affect the semantics. In this case, this is so because of the intensional variable used for the trace of VP_2. A related technical point is that at the position of the trace, the trace semantics has to be applied to the local world of evaluation in order to fit it into the local compositional context. This can be achieved with a special "intensional traces and pronouns rule", or

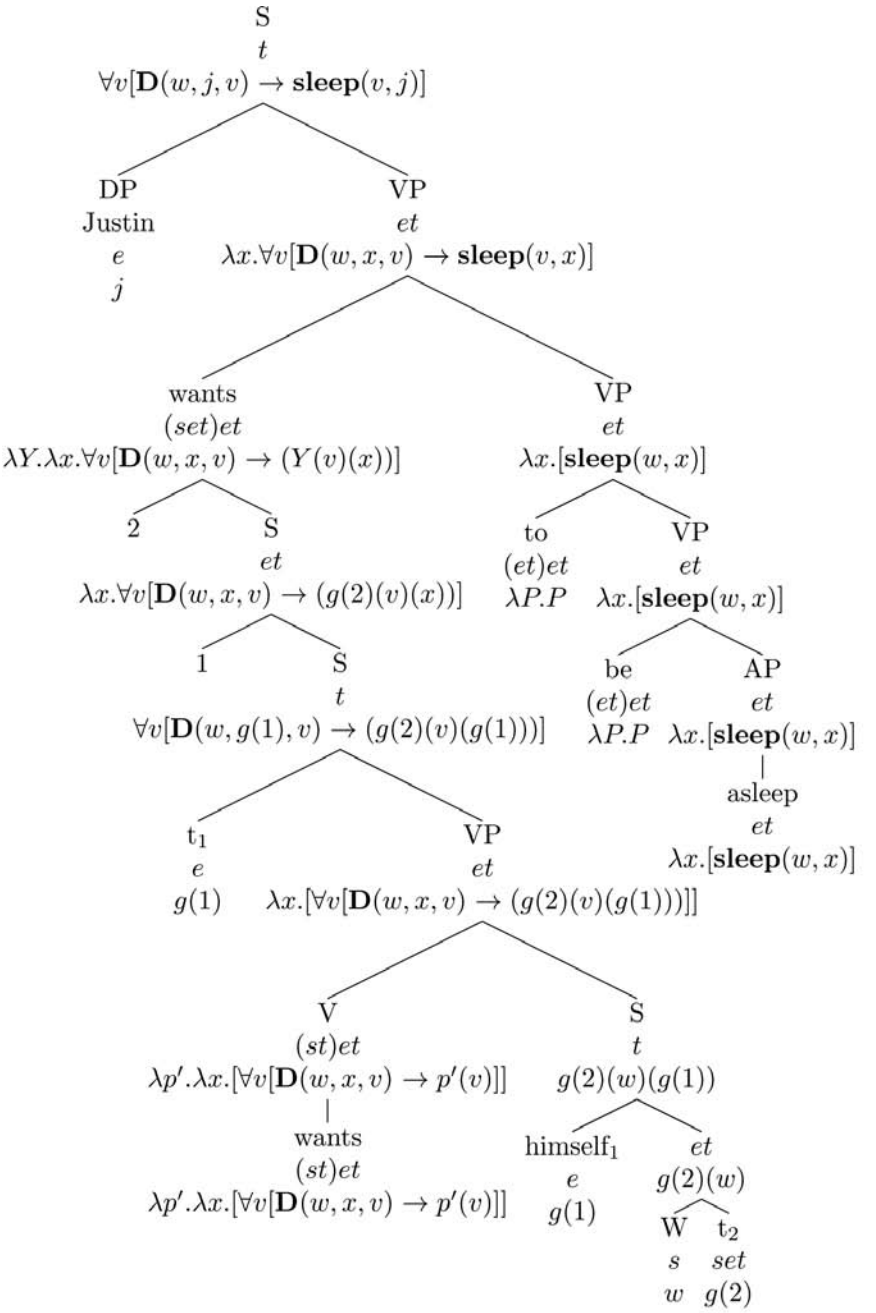

FIGURE 5 *The lambda abstract* [2 [1 ...]] *from the unzipped paraphrase used as a syntactic definition of VP-embedding* want

as in Figure 4 by putting the world variable explicitly into the tree; see Abusch (2012) for a statement of the intensional trace rule and an application of it. In generalizing the procedure, intensional variables should be used systematically for higher-type arguments.

Note that we obtain an equivalence between (8a) and (8b) for any NP_1 and VP_2, not just the particular ones seen in (1).

(8) a. NP_1 wants him/himself VP_2.
b. NP_1 wants VP_2.

Turning to the self-example, since in (2b) *admiring* and *yeti* are to be the arguments of the morpheme *self*-, the pattern of definition is the same, because there are two arguments. Figure 6 shows the result of definition by paraphrase, with the syntactic abstract coming from an unzipped paraphrase used as a syntactically structured definition for the target word. Vertiginously, the syntax tree below *self*- consists entirely of variables and binding indices. The derivation in Figure 6 uses an extensional system of interpretation.

The derivations in this section were computed using the derivation calculator designed and implemented by Lucas Champollion and his colleagues (Champollion & Tauberer & Romero 2007, Champollion & Tauberer & Romero & Bumford 2013).

3 Operators for Definition by Paraphrase

This section reworks the procedure from Section 2, using type-raising operators in place of syntactic scoping. The goal is to tie the combinatory syntax of the defined word with the derived semantics, and to derive that semantics completely mechanically. This will be done with some cross-categorial operators that are employed uniformly. The procedure is illustrated in (9) for the problem of defining the operator RAI that combines with a quantified nominal, a transitive verb, and its subject to quantify in the nominal with minimal scope. As shown in (9a), the first argument in the source derivation is modified with an operator A3 (for the argument that is third from the last). The second argument (the transitive verb) is modified with A2 (for argument that is second from the last), and the last argument in the source derivation (the subject) is modified with A1 (for last argument). The operators A1, A2, and A3 are designed so that both the target meaning for RAI, which is (9b), and its target syntactic category, which is (9c), can be extracted trivially from the semantics and categorial syntactic category for (9a).

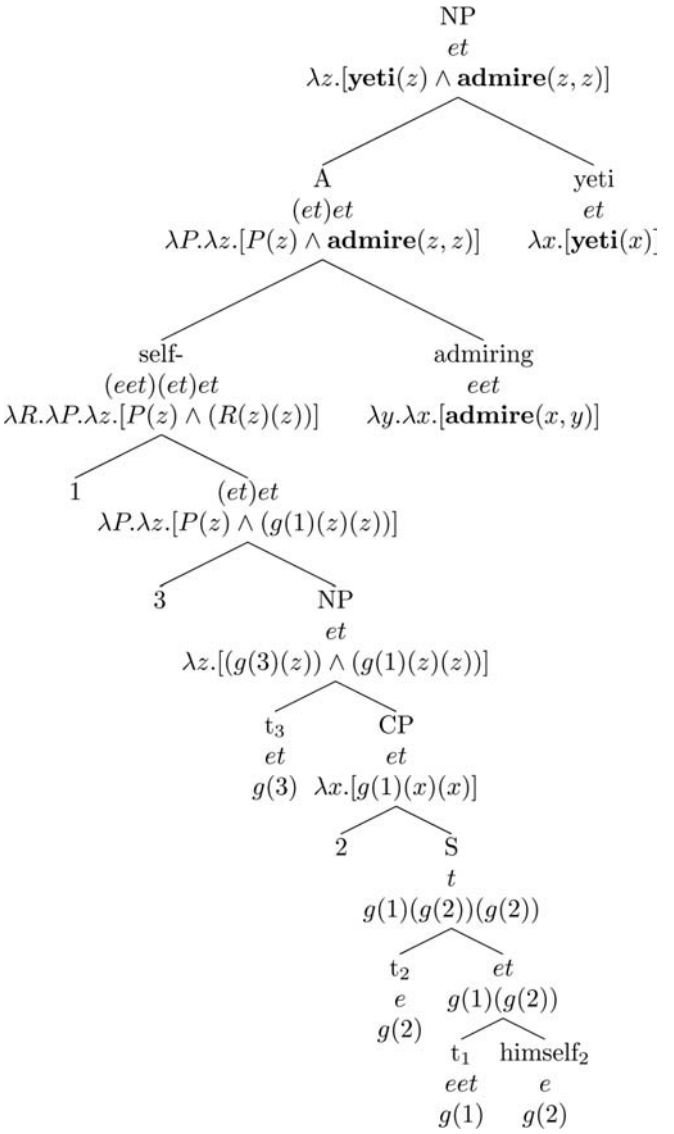

FIGURE 6 Self- *in* self-admiring yeti *defined by paraphrase from* yeti who admires himself

(9) a. (A1 Justin) (A2 admired) (A3 somebody)
b. $\lambda r \lambda r_1 \lambda r_2 .\, r(\lambda r_3 .\, r_1(r_3)(r_2))$
c. $((((e\backslash t)/e)\backslash(e\backslash t))/(t/(e\backslash t)))$
d. [Justin [admired [RAI somebody]]]

We have switched to a categorial syntax, where a transitive verb has category $((e\backslash t)/e)$, a subject generalized quantifier has category $(t/(e\backslash t))$, and a verb phrase has category $(e\backslash t)$. This already determines the category (9c) of RAI: something with this syntactic type combines with a generalized quantifier on the right (the object), then with a transitive verb on the left, then with the individual type e on the left, to produce the sentence type t.

My definitions for the A-operators use the continuation scope calculus of Shan & Barker (2006). For our purposes this calculus can be characterized as using special categorial symbols of the form $c/\!\!/(a\backslash\!\!\backslash b)$. Something of category $c/\!\!/(a\backslash\!\!\backslash b)$ has the local combinatory properties of category a, and takes scope at category b to produce something of category c. To illustrate, (10a) characterizes the category symbol for a scope-taking DP *everyone*. It has the local category e, because it has the local combinatory properties of (for instance) a proper name. And it takes scope at the sentence category t, resulting in something that again has type t. (10b) shows the category for the quantificational determiner *every*. It first takes an NP argument on the right, resulting in a scope-taking quantifier. NP is represented categorially as $e\backslash_N t$. The idea is that NP is similar to the VP type $e\backslash t$, but can not directly act as a functor in the syntax. The indexed slash category $e\backslash_N t$ can only be used syntactically as an argument of something else, such as *every*.

(10)

	category	local	scope	result	argument
everyone	$t/\!\!/(e\backslash\!\!\backslash t)$	e	t	t	
every	$(t/\!\!/(e\backslash\!\!\backslash t))/(e\backslash_N t)$	e	t	t	$e\backslash_N t$

(11) gives information parallel to (9) for the derivation of *self-*. In structure (11a), the noun *yeti* is modified with A1, because the noun is the last argument of the defined operator. The verb *admired* is modified with A2, because it is the penultimate argument of the defined operator *self-*. The symbol t is the trace of *who*, and B is a binding operator that binds the pronoun *himself* to its antecedent, the trace. In Barker and Shan's system, such binding operators are used in place of indexing. The items *B*, *t*, and *who* have complex syntactic and semantic definitions in the categorial system, which are not directly relevant here—see Barker and Shan's paper.

(11) a. (A1 yeti) who B t (A2 admired) himself
b. $\lambda r_1 \lambda r_2 \lambda a.\, r_2(a) \wedge r_1(a)(a)$
c. $((e\backslash_N t)/(e\backslash_N t))/((e\backslash t)/e)$
d. [[self- admiring] yeti]

The project now is to define the A-operators in a way that mimics the scope relations in the unzipped derivations from Section 2. Look at Figure 6, where the surface position of *yeti* in the zippered derivation corresponds to the trace t_3. This trace locally has category NP (categorial type $e\backslash_N t$). The moved phrase takes scope at NP. So the type for [A1 yeti] is of the form $c/\!\!/((e\backslash_N t)\backslash\!\!\backslash(e\backslash_N t))$, and the type for this use of A1 is of the form $(c/\!\!/((e\backslash_N t)\backslash\!\!\backslash(e\backslash_N t)))/(e\backslash_N t)$, with the argument of A1 added to the right. What is the result type c? Earlier I said that the A-operators would derive also the syntactic category of the defined word. To achieve this, the right choice for c is $(e\backslash_N t)/(e\backslash_N t)$, reflecting the fact that [self-admiring] takes an NP argument on the right and also creates an NP. To control the derivation, I will instead use $(e\backslash_N t)/_1(e\backslash_N t)$, with the subscript marking the last argument of the defined word. The category for this use of A1 is recorded in the line *self-* in (12). A1 as used in (9a) has simpler types. The argument category is the subject, and has type e rather than $e\backslash_N t$. The scope is type t rather than $e\backslash_N t$. And since RAI takes its last argument on the left, the result type is $e\backslash_1 t$. See the line RAI in (12).

(12)

	A1
self-	$(((e\backslash_N t)/_1(e\backslash_N t))/\!\!/((e\backslash_N t)\backslash\!\!\backslash(e\backslash_N t)))/(e\backslash_N t)$
RAI	$((e\backslash_1 t)/\!\!/(e\backslash\!\!\backslash t))/e$

In schematizing the category of A1, there are three dimensions. There is an argument type α, which for *self* is $e\backslash_N t$ and for RAI is e. There is a scope type β, which for *self* is $e\backslash_N t$ and for RAI is t. And finally, we allow the defined word to take its last argument on the right (like *self-*) or on the left like RAI. This results in the schematized categories $A1_L$ and $A1_R$, as defined in (13). α and β are type variables which can be instantiated with any categorial syntactic type.

(13)

$A1_R$	$((\beta/_1\alpha)/\!\!/(\alpha\backslash\!\!\backslash\beta))/\alpha$
$A1_L$	$((\alpha\backslash_1\beta)/\!\!/(\alpha\backslash\!\!\backslash\beta))/\alpha$

Following the plan of the derivations in Section 2, A2 should always scope directly above A1—or technically, it should scope above $\beta/_1\alpha$, corresponding to a defined word that takes its final argument on the right. Or it should scope above $\alpha\backslash_1\beta$, corresponding to a final argument on the left. Independently, the defined word could take its penultimate argument either on the left or on the right. This results in four schematized definitions for A2, as given in (14). To explain the notation, a word defined using $A2_{LR}$ takes its penultimate argument on the right (indicated with R), and takes its final argument on the

left (indicated with L). α_1 is a type variable corresponding to the final argument of the defined word, and α_2 is a type variable corresponding to the penultimate argument. As before β is the result type.

(14)
$$
\begin{array}{ll}
\mathrm{A2_{RR}} & (((\beta/_1\alpha_1)/_2\alpha_2)/\!\!/(\alpha_2\backslash\!\!\backslash(\beta/_1\alpha_1)))/\alpha_2 \\
\mathrm{A2_{LR}} & (((\alpha_1\backslash_1\beta)/_2\alpha_2)/\!\!/(\alpha_2\backslash\!\!\backslash(\alpha_1\backslash_1\beta)))/\alpha_2 \\
\mathrm{A2_{RL}} & ((\alpha_2\backslash_2(\beta/_1\alpha_1))/\!\!/(\alpha_2\backslash\!\!\backslash(\beta/_1\alpha_1)))/\alpha_2 \\
\mathrm{A2_{LL}} & ((\alpha_2\backslash_2(\alpha_1\backslash_1\beta))/\!\!/(\alpha_2\backslash\!\!\backslash(\alpha_1\backslash_1\beta)))/\alpha_2
\end{array}
$$

The semantics of the A-operators is simple. Let σ be the function that maps syntactic to semantic types. For instance, for the transitive verb type. $\sigma(((e\backslash t)/e)) = eet$ and for an ordinary subject quantifier, $\sigma((t/(e\backslash t))) = (et)t$. In general, a continuation scope type $c/\!\!/(a\backslash\!\!\backslash b)$ has a semantic type $(\sigma(a)\sigma(b))\sigma(c)$. For instance, the lexical continuation quantifier *someone* has syntactic type $t/\!\!/(e\backslash\!\!\backslash t)$, and the denotation $\lambda P \exists x.\, \mathbf{person}(x) \wedge P(x)$ of semantic type $(et)t$. The continuation scope calculus is set up so that the denotation of a phrase of type $c/\!\!/(a\backslash\!\!\backslash b)$ gets applied to the abstract of type $\sigma(a)\sigma(b)$ that is obtained by binding with lambda a variable of type $\sigma(a)$ in the local position of the phrase. The binding takes place at a scope level of type b.

Since $\mathrm{A1_R}$ has syntactic type $((\beta/_1\alpha)/\!\!/(\alpha\backslash\!\!\backslash\beta))\ /\alpha$, its denotation should have type $(\sigma(\alpha)((\sigma(\alpha)\sigma(\beta))\sigma(\beta)))$. The initial argument of type $\sigma(\alpha)$ corresponds to the phrase of type α that is modified by $\mathrm{A1_R}$, such as *yeti* in the case of the definition of *self-*. We are not really interested in the semantics of *yeti*—we just want the derivation to pick up its syntactic and semantic type. Looking back to Section 2, a variable of the right type in the local position of *yeti* should be bound with lambda. Since the work of binding is already done by the continuation scope calculus, we obtain the simple denotations for the A-operators as projection functions in (15). The other varieties of A1 and A2 are identical in the semantics.

(15)

	syntax
$\mathrm{A1_R}$	$((\beta/_1\alpha)/\!\!/(\alpha\backslash\!\!\backslash\beta))\ /\alpha$
$\mathrm{A2_{RR}}$	$(((\beta/_1\alpha_1)/_2\alpha_2)/\!\!/(\alpha_2\backslash\!\!\backslash(\beta/_1\alpha_1)))/\alpha_2$

	semantics
$\mathrm{A1_R}$	$\lambda x_{\sigma(\alpha_1)}\lambda k_{\sigma(\alpha_1)\sigma(\beta)}.\, k_{\sigma(\alpha_1)\sigma(\beta)}$
$\mathrm{A2_{RR}}$	$\lambda x_{\sigma(\alpha_2)}\lambda k_{\sigma(\alpha_2)\sigma(\alpha_1)\sigma(\beta)}.\, k_{\sigma(\alpha_1)\sigma(\alpha_1)\sigma(\beta)}$

With these definitions, definitions by paraphrase can be obtained systematically. (16a) is the source for the definition by paraphrase of *self-*. $\mathrm{A1_R}$ and $\mathrm{A2_{RR}}$

are used because *self-* takes both of its argments on the right. Running the derivation results in the syntactic type (16b), or (16d) after erasure of slash indices, and the semantics (16c). The derived lexical entry for *self-* is the syntax (16d) paired with the semantics (16c). With *self-* defined in this way, (17a) has the semantics (17b). As required in the problem statement from Section 1, this is also the semantics of (17c). Naturally, we also get equivalence when the alternative phrasings are embedded, as in (18).

(16) a. ($A1_{RR}$ yeti) who B t ($A2_{RR}$ admired) himself
b. $(((e\backslash_N t)/_1(e\backslash_N t))/_2((e\backslash t)/e))$
c. $\lambda r_1 \lambda r_2 \lambda a.\, r_2(a) \wedge r_1(a)(a)$
d. $((e\backslash_N t)/(e\backslash_N t))/((e\backslash t)/e)$

(17) a. self-admiring yeti
b. $\lambda a_e.\, \mathbf{yeti}(a_e) \wedge \mathbf{admire}(a_e, a_e)$
c. yeti who B t admires himself

(18) a. Every self- admiring yeti admires Justin.
b. $\forall x_e.\, \mathbf{yeti}(x_e) \wedge \mathbf{admire}(x_e, x_e) \rightarrow \mathbf{admire}(x_e, j)$
c. Every yeti who B t admires himself admires Justin.

(19)–(20) run through the phrase orders that are derived using the different operators. Four categories are derived in (19), and these result in the four surface orders for *self-admiring yeti* that are seen in (20). The semantics in each case is as in (17). These four lexical entries illustrate that the A-operators derive both a semantics and a parallel categorial syntax for the defined word.

(19) self_{RR} $A1_R$ yeti who B t $A2_{RR}$ admired himself $((e\backslash_N t)/(e\backslash_N t))/((e\backslash t)/e)$
self_{LR} $A1_L$ yeti who B t $A2_{LR}$ admired himself $((e\backslash t)/e)\backslash((e\backslash_N t)\backslash(e\backslash_N t))$
self_{RL} $A1_R$ yeti who B t $A2_{RL}$ admired himself $((e\backslash_N t)/(e\backslash_N t))/((e\backslash t)/e)$
self_{LL} $A1_L$ yeti who B t $A2_{LL}$ admired himself $((e\backslash t)/e)\backslash((e\backslash_N t)\backslash(e\backslash_N t))$

(20) a. self_{RR} admiring yeti
b. yeti self_{LR} admiring
c. admiring self_{RL} yeti
d. yeti admiring self_{LL}

Additional arguments work the same way. (21) defines the four versions of A3, and (22a) is the source structure for defining RAI. Running a syntactic and semantic derivation results in the syntactic type (22b), and the semantics (22c).

Erasing indices in (22b) results in (22d), which paired with (22c) is the syntactic and semantic lexical entry for RAI.

(21) $A3_{RR}$ $(((\beta/_2\alpha_2)/_3\alpha_3)/\!\!/(\alpha_3\backslash\!\!\backslash(\beta/_2\alpha_2)))/\alpha_3$
$A3_{LR}$ $(((\alpha_2\backslash_2\beta)/_3\alpha_3)/\!\!/(\alpha_3\backslash\!\!\backslash(\alpha_2\backslash_2\beta)))/\alpha_3$
$A3_{RL}$ $((\alpha_3\backslash_3(\beta/_2\alpha_2))/\!\!/(\alpha_3\backslash\!\!\backslash(\beta/_2\alpha_2)))/\alpha_3$
$A3_{LL}$ $((\alpha_3\backslash_3(\alpha_2\backslash_2\beta))/\!\!/(\alpha_3\backslash\!\!\backslash(\alpha_2\backslash_2\beta)))/\alpha_3$

(22) a. ($A1_L$ Justin) ($A2_{LL}$ admired) ($A3_{LR}$ somebody)
b. $((((e\backslash t)/e)\backslash_2(e\backslash_1 t))/_3(t/(e\backslash t)))$
c. $\lambda r\lambda r_1\lambda r_2.\, r(\lambda r_3.\, r_1(r_3)(r_2))$
d. $((((e\backslash t)/e)\backslash(e\backslash t))/(t/(e\backslash t)))$

4 *Want-VP* and *Want-DP*

The problems involving *want* require an intensional semantics, since the basic and derived versions of *want* are intensional. (23) illustrates an intensional version of the categorial system, with world variables added to basic relations. As in Section 1, $\mathbf{sleep}(\nu, j)$ is understood as "j sleeps in world ν", and $D(w, j, \nu)$ is understood as "world ν is a buletic alternative for individual j in world w". As shown in the lexical entries of (23c), the last argument of any lexical entry is a world. To match worlds, for any functor category, this world is supplied as an argument of each argument.

(23) a. B Justin wanted himself to be asleep
b. $\lambda w\forall\nu.\, D(w, j, \nu) \rightarrow \mathbf{sleep}(\nu, j)$
c.

sleep	$(e\backslash t)$	$\lambda x\lambda w.\, \mathbf{sleep}(w, x(w))$
want	$(e\backslash t)/t$	$\lambda p\lambda x\lambda w.\, \forall\nu D(w, a(w), \nu) \rightarrow p(\nu)$
Justin	e	$\lambda w.\, j$

Adding intensionality does not necessitate any adjustment in the A-operators. (24a) is the source structure for defining the VP-embedding *want*. A derivation for this structure produces the syntactic category (24a), or (24d) after erasure of indexes, and the semantics (24c). The new lexical entry for *want* is (24d) paired with (24c). Using this in (24e) results in the same semantics as in (23).

(24) a. (A1 Justin) wanted himself (A2 to be asleep)
b. $((e\backslash_1 t)/_2(e\backslash t))$
c. $\lambda r\lambda x\lambda w\forall\nu.\, D(w, x(w), \nu) \rightarrow r(x)(\nu)$

d. $((e\backslash t)/(e\backslash t))$
e. Justin wanted to be asleep.

For DP-embedding *want*, there is an additional complexity. Assuming we are looking for a version of *want* that combines with an ordinary quantifier of category $(t/(e\backslash t))$, it is necessary to fit this into the derivation. This can be done with an intensional version of the RAI operator. (25) shows the derivation for defining DP-embedding *want*.

(25) a. (A2 Justin) wanted to have RAI (A1 a duck)
b. $((e\backslash_1 t)/_2(t/(e\backslash t)))$
c. $\lambda r\lambda r_1\lambda w\forall v.\, D(w, r_1(w), v) \rightarrow r(\lambda r_3\lambda w.\, \mathbf{have}(w, r_1(w), r_3(w)))$
d. $((e\backslash t)/(t/(e\backslash t)))$

In the following way, the A-operators encapsulate a process of defininition by paraphrase. Starting from categorial grammar (i.e. a lexicon) that derives the paraphrase, and a target syntax for the word that is to be defined, one or more operators that correspond to the target syntax are inserted in the paraphrase. Solving for a derivation produces a semantics for the defined word in the form of a lambda term, and a syntax in the form of a categorial label. Transforming the latter trivially (by deleting slash indices) produces the syntactic categorial label for the defined word.

This process can actually be executed computationally using Barker and Shan's derivation calculator for continuation categorial grammar (Barker & Shan 2005). Source files for my derivations are included in Rooth (2014). The lexical items, operators, and results differ only notationally from those presented here.

5 Discussion

The operators from Section 3 provide a generic solution to the paraphrase problems posed in Section 1. Starting from a grammar that generates the syntax and semantics of the paraphrase, the syntax and semantics of the target lexical item are derived formally.

The method gives a systematic way of solving a class of problems that are frequently posed in semantics classes—one can hope that their being automatically solvable will not undermine the interest and utility of such problems. Equally, the method can be used to address problems that come up in research, in a way that is guaranteed in virtue of the formal methodology

to get the technical minutae right. Certainly, definitions with the complexity of the definition of DP-embedding *want* in (25c) are frequently erroneous in some minor technical way, even in publications.

It came out in Section 2 that it is possible to think of the lexical entries that are derived by the scoping method syntactically, as providing a syntactically structured lexical entry for the target word that is in a certain way parallel to the syntax and compositional semantics of the paraphrase. In this there are general connections with theoretical models that employ syntactically structured derivations for surface words, such as generative semantics (McCawley 1976) and distributed morphology (Halle & Marantz 1993; Harley 2012). There is a more specific connection with the proposal in Hale and Keyser (1993; 2002) that some lexical items have a clause-like internal syntax that is generated in the lexicon. To explore connections with these theoretical approaches in morphology, one could try to cast the A-operators as null affixes that are involved in morphological and/or syntactic derivations.

References

Abusch, Dorit (2012): "Circumstantial and temporal dependence in counterfactual modals". *Natural Language Semantics* 20.3, 273–297.

Barker, Chris (2007): "Parasitic scope". *Linguistics and Philosophy* 30.4, 407–444.

Barker, Chris & Chung-chieh Shan (2005): *Reference parser for: explaining crossover and superiority as left to right evaluation*. URL: http://semanticsarchive.net/Archive/TI1M2UxN/.

Champollion, Lucas & Josh Tauberer & Maribel Romero (2007): *The Penn Lambda Calculator. Pedagogical Software for Natural Language Semantics*. In: King, Tracy Holloway & Emily M. Bender eds.: Proceedings of the GEAF 2007 Workshop, July 13–15 2007. CSLI On-line Publications. URL: http://ling.upenn.edu/ champoll/lambda-geaf .pdf.

Champollion, Lucas & Josh Tauberer & Maribel Romero & Dylan Bumford (2013): *The lambda calculator for students and teachers of natural language semantics*. URL: http://github.com/dylnb/LambdaCalculator.

Hale, Ken & Samuel J. Keyser (1993): "On argument structure and the lexical expression of syntactic relations". In: Hale, Ken & Samuel J. Keyser, eds.: *The View from Building 20*. Cambridge: MIT Press.

Hale, Ken & Samuel J. Keyser (2002): *Prolegomenon to a Theory of Argument Structure*. Cambridge: MIT Press.

Halle, Morris & Alec Marantz (1993): "Distributed morphology and the pieces of inflection". In: Hale, Ken & Samuel J. Keyser, eds.: *The View from Building 20*. Cambridge, MA: MIT Press, 111–176.

Harley, Heidi (2012): "Semantics in distributed morphology". In: Maienborn, Claudia & Klaus von Heusinger & Paul Portner, eds.: *Semantics. An International Handbook of Natural Language Meaning*. Vol. 3. Berlin/New York: Mouton de Gruyter, 2151–2171.

Heim, Irene & Angelika Kratzer (1998): *Semantics in Generative Grammar*. Oxford: Blackwell.

Link, Godehard (1979): *Montague-Grammatik. Die logischen Grundlagen*. München: Fink.

McCawley, James (1976): *Grammar and Meaning*. New York: Academic Press.

Rooth, Mats (2014): "Replication data for: Operators for definition by paraphrase." *Harvard Dataverse Network*. URL: http://dx.doi.org/10.7910/DVN/25512.

Sauerland, Uli (1998): "Plurals, derived predicates, and reciprocals". In: Sauerland, Uli & Orin Percus, eds.: *The Interpretive Tract*. Vol. 25. MIT Working Papers in Linguistics, 177–204.

Shan, Chung-chieh & Chris Barker (2006): "Explaining crossover and superiority as left-to-right evaluation". *Linguistics and Philosophy* 29.1, 91–134.

Zimmermann, Thomas Ede (2006): "Quaint 'paint'". In: Gärtner, Hans-Martin, ed.: *Between 40 and 60 Puzzles for Krifka*. URL: http://www.zas.gwz-berlin.de/fileadmin/material/40-60-puzzles-for-krifka/pdf/zimmermann.pdf.