Movement as a reflex of higher-order structure building*

Patrick D. Elliott
Leibniz-Zentrum Allgemeine Sprachwissenschaft

April 26, 2019

Merge-based theories of movement prevalent in minimalist syntax (both copy-theoretic and multidominance implementations) raise major unresolved issues for a theory of the syntax-semantics interface. Unless a moved expression is of type e, it cannot receive the same interpretation at both its base position and its landing site – rather, movement should give rise to an *operator-variable* dependency. The standard move in the literature is to posit a rule of Trace Conversion (see, e.g., Sauerland 1998, 2004, Elbourne 2001, Fox 2002 and others). There are different ways to cash this out – either as a syntactic operation that manipulates lower copies, undercutting the conceptual appeal of the Merge-based theories of movement, or as a non-compositional interpretation rule. In this paper, we aim to preserve the conceptual appeal of Merge-based theories of movement, while providing a syntax-semantics interface for movement dependencies that doesn't necessitate an operation such as Trace Conversion. We take it that a desideratum for theories of the syntax-semantics interface is that each step in the syntactic derivation should correspond directly to a step in the semantic computation. We develop a theory with this character by first developing a new perspective on movement – we treat movement as a reflex of *higher-order merge*, i.e., we allow in syntactic operations which introduce higher-order functions over structures. Such machinery has been a foundation of formal semantics since at least Montague's work on quantification, but is rarely applied to the realm of syntax. We then provide a compositional way of interpreting Merge-based derivations using machinery developed by Charlow (2014), Barker & Shan (2014), and others.

*I'm grateful to Simon Charlow, Julian Grove, and Deborah Wong for feedback. I'm also particularly indebted to Rob Pasternak, conversations with whom lead to the ideas underlying this paper.

1 Movement in Minimalist Syntax

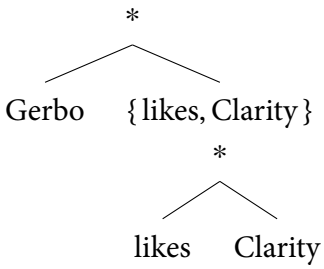
1.1 Merge-based theories of movement

We take as our starting point Chomsky's (1995) hypothesis that the basic structure building operation in natural language is Merge ($*$). We define Merge in the standard way – it is a function that takes two Syntactic Objects (sos) $\mathbb{X}$ and $\mathbb{Y}$ and returns a new (unlabelled) so: $\{\mathbb{X}, \mathbb{Y}\}$. We take the *type* of a so to be $\Bbbt$.[1]

(2) Merge (def.)
$\mathbb{X} * \mathbb{Y} := \{\mathbb{X}, \mathbb{Y}\}$ $::= \Bbbt \to \Bbbt \to \Bbbt$[2]

Merge successively applies to sos, constructing a structured representation, as in (3). Note that the tree is a graph of the *derivation*, rather than a representation in its own right.

(3) $\{\text{Gerbo}, \{\text{likes}, \text{Clarity}\}\}$



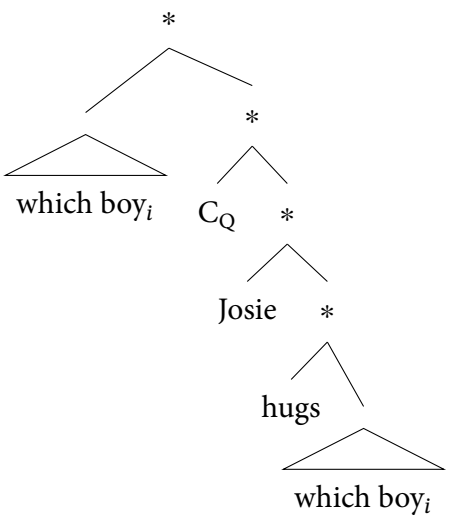
There are certain expressions in natural language which are pronounced in positions other than where they are interpreted, such as *wh*-expressions. The standard approach to this phenomenon in minimalist syntax is so-called Internal Merge. This can be cashed out in two different ways. According to the copy-theory of movement, schematized in (4), movement involves re-merging a *copy* of a so contained within some derived syntactic structure. An alternative implementation – the multi-dominance theory of movement – treats syntactic structures as multi-dominance graphs, as schematized in (5); movement simply amounts to re-merging some so contained within the derived structure, resulting in a so with two distinct parents. For concreteness, I'll adopt copy-theoretic representations in what follows (although nothing hinges on this).

[1] If we take the type of the atomic unit of syntactic computation to be L (i.e., a lexical item, or a *root*, etc.), we can define the the type of a so recursively as an atomic unit, or a set of sos.

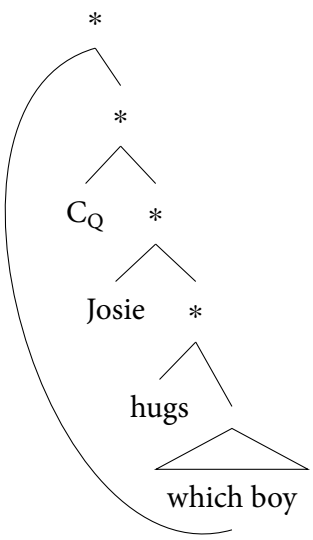
(1) $\Bbbt := \mathsf{L} \mid \{\Bbbt\}$

[2] We use $(\to)$ as the type-constructor for *function* types (cf. Heim & Kratzer's 1998 angled bracket notation). $(\to)$ is *right-associative*, so $\mathsf{a} \to \mathsf{a} \to \mathsf{a}$ is equivalent to $\mathsf{a} \to (\mathsf{a} \to \mathsf{a})$.

(4)



(5)

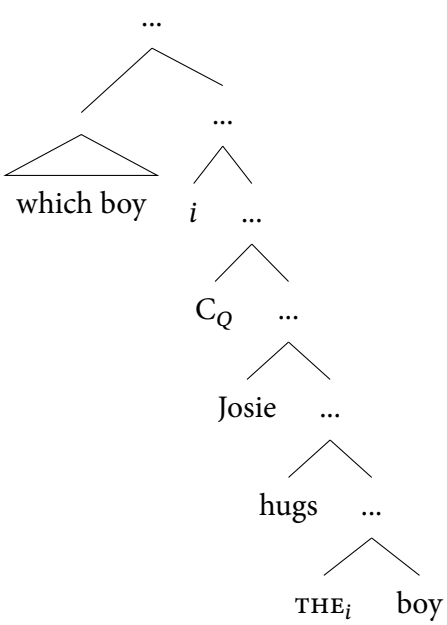


1.2 Trace Conversion and the syntax-semantics interface

Regardless of how Internal Merge is implemented, the representation fed to the semantics must be something like that in (6). If we take *wh-expressions* to have a scopal semantics, then it they cannot be interpreted in both their surface *and* base position, without giving rise to a type-mismatch. This gives rise to a tension with the syntactic representation delivered by internal merge. The standard solution, as illustrated, is to replace the determiner in the lower copy with a bound definite determiner, with the semantics in (7) (Fox 2002; see Sauerland 1998, 2004 for an alternative implementation based on choice functions). There are number of different ways of thinking about how this replacement comes about,[3] but the most straightforward implementation is to treat *trace conversion* as a syntactic operation that applies to the lower copy as a reflex of movement. One could also think of *trace conversion* as a post-syntactic transformation rule that applies to the syntactic representation after it has been fed to the interface.

[3] See, e.g., Fox & Johnson (2016) for an implementation that takes advantage of a *late-insertion* architecture.

(6)



(7) $[\![\text{THE}_i]\!]^g = \lambda P \,.\, \iota x[P\ x \wedge x = g_i]$

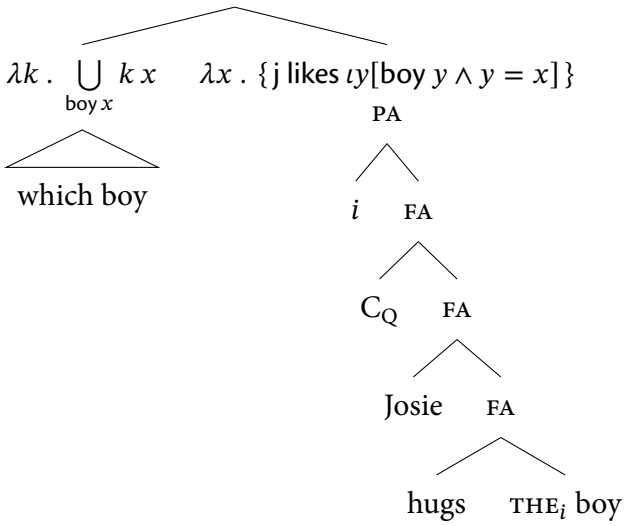
(8) Trace Conversion (def.)
TC $\{\mathbb{D}_i, \mathbb{N}\} \coloneqq \{\text{THE}_i, \mathbb{N}\}$

Additionally, a binding index must be inserted immediately below the higher copy, in order to trigger abstraction over the lower copy (Heim & Kratzer's abstraction rule is given in (9)). Again, this can be thought of as a syntactic reflex of the movement operation itself, or as a post-syntactic transformation rule. The resulting semantic composition is schematized in (10).[4]

(9) Predicate Abstraction (def.)
$[\![\{i, \mathbb{X}\}]\!]^g = \lambda x \,.\, [\![\text{X}]\!]^{g[i \to x]}$

[4] I've made a couple of non-trivial assumptions concerning the semantic contribution of *wh*-expressions and the interrogative complementizer here, none of which are crucial. I assume that C_Q is responsible for lifting propositions into Hamblin denotations; *wh*-expressions scope alternatives over questions – see, e.g., Heim 1994, Cresti 1995, Charlow 2017.

(10) $\{\text{j hugs } \iota y[\text{boy } y \wedge y = x] \mid \text{boy } x\}$



The need for an operation of Trace Conversion, and insertion of a *binding index*, means that much of the elegant simplicity underlying the conceptual appeal of Internal Merge is lost – arguably, *trace conversion* is the name for a problem rather than a solution (compare other magical syntactic operations, such as Fiengo & May's 1994 *vehicle change*).

In the following section, I'll sketch an approach to syntactic displacement which makes use of *higher order sos*, whilst retaining the conceptual appeal of Internal Merge. This new formulation will mean that interpretation of movement operations can proceed in tandem without the need for syntactic magic, such as binding indices and trace conversion. The basic tenet of minimalist syntax – namely, that structure building proceeds via successive applications of Merge – will be preserved.

2 A new approach to movement

2.1 Higher-order structure building

In orthodox syntactic derivations, we typically only avail ourselves of functions ranging over sos, which return sos, e.g., Merge. The shift in perspective I'd like to adopt is that, over the course of the derivation, certain functions *themselves* take functions as their input. The use of such *higher-order functions* is totally standard in the literature on formal semantics (see, e.g., Heim & Kratzer 1998). For example, quantificational DPs are typically treated as being of $(\mathsf{e} \to \mathsf{t}) \to \mathsf{t}$, i.e., as functions from *functions* to a truth value. In a certain sense then, the theory that I outline below can be thought of as a *scopal* theory of syntactic structure building.

As a starting point, we'll retain the standard formulation of Merge (repeated below) as our primitive structure-building operation:

(11) Merge (def.)

$\mathbb{X} * \mathbb{Y} \coloneqq \{\mathbb{X}, \mathbb{Y}\}$ $\quad :: = \Bbbt \to \Bbbt \to \Bbbt$

We'd like to suggest a new formulation of internal merge, which we call s-Merge. The formulation of s-Merge is really where the novelty of the proposal lies, so we'll dwell first here. s-Merge is a function from a k – itself a function from an so to an so – that returns an so. Here is the informal intuition that this definition captures – $\star\,\mathbb{X}$ contributes an $\mathbb{X}$ *locally*, and additionally contributes a Merge operation which *takes scope* over an so.

(12) s-Merge (def.)
$\star\,\mathbb{X} \coloneqq \lambda k\,.\,\mathbb{X} * (k\,\mathbb{X})$ $\qquad ::= (\Bbbt \to \Bbbt) \to \Bbbt$

Barker & Shan (2014) introduced a convenient abbreviation for scopal values and types: the so-called *tower* notation.

(13) tower types (def.):
$$\frac{\mathsf{b}}{\mathsf{a}} \coloneqq (\mathsf{a} \to \mathsf{b}) \to \mathsf{b}$$

(14) tower values (def.):
$$\frac{f\,[\,]}{x} \coloneqq \lambda k\,.\,f\,(k\,x)$$

We can now rewrite our definition of s-Merge using tower notation. In the following, we'll flip-flop between lambda notation and towers, depending on what is convenient for exposition.

(15) $\star\,\mathbb{X} = \dfrac{\mathbb{X} * [\,]}{\mathbb{X}}$ $\qquad ::= \Bbbt \to \dfrac{\Bbbt}{\Bbbt}$

How do higher-order sos merge with existing syntactic structure? First, we need a function Lift ($\uparrow$) that takes a so, and returns a *trivially* higher-order so. Next, we define Higher Order Merge ($\circledast$), which is itself just defined in terms of Merge ($*$). To get us used to the parallel between lambda-notation and towers, we've also written Lift and Higher Order Merge using tower notation in (21).[5]

[5] The continuation applicative is a tuple consisting of a type-constructor $\mathsf{K_b}$, and two functions π and S, s.t.:

(16) $\mathsf{K_b} \coloneqq \dfrac{\mathsf{b}}{\mathsf{a}}$ $\qquad$ (17) $x^{\pi} \coloneqq \dfrac{[\,]}{x}$ $\qquad$ (18) $\dfrac{f\,[\,]}{x}\,\mathsf{S}\,\dfrac{g\,[\,]}{y} \coloneqq \dfrac{f\,[\,g\,[\,]\,]}{\mathsf{A}\,x\,y}$

Lift is just a typed instantiation of (π). Higher Order Merge can be derived by lifting Merge into the continuation applicative. In, e.g. Haskell, lifting a binary function into an applicative in this fashion is captured via `LiftA2`.

(19) $(\circledast) \coloneqq \lambda m\,.\,(\mathsf{S})\,((*)^{\pi}\,\mathsf{S}\,m)$

(20) a. LIFT (def.)

$\mathbb{X}^{\uparrow} \coloneqq \lambda k \,.\, k\, \mathbb{X}$ $\quad ::= \Bbbt \to \frac{\Bbbt}{\Bbbt}$

b. HIGHER-ORDER MERGE (def.)

$m \circledast n \coloneqq \lambda k \,.\, m\, (\lambda \mathbb{X} \,.\, n\, (\lambda \mathbb{Y} \,.\, k\, (\mathbb{X} * \mathbb{Y})))$ $\quad ::= \frac{\Bbbt}{\Bbbt} \to \frac{\Bbbt}{\Bbbt} \to \frac{\Bbbt}{\Bbbt}$

(21) a. $\mathbb{X}^{\uparrow} \coloneqq \frac{[\,]}{\mathbb{X}}$

b. $\frac{f\,[\,]}{\mathbb{X}} \circledast \frac{g\,[\,]}{\mathbb{Y}} \coloneqq \frac{f\,[\,g\,[\,]\,]}{\mathbb{X} \circledast \mathbb{Y}}$

Finally, we need a way of lowering a higher-order SO to a normal SO. We accomplish this by simply saturating the k argument of the higher-order SO (i.e., the *continuation* argument) with the identity function.

(22) LOWER (def.)
$m^{\downarrow} \coloneqq m\ \mathsf{id}$

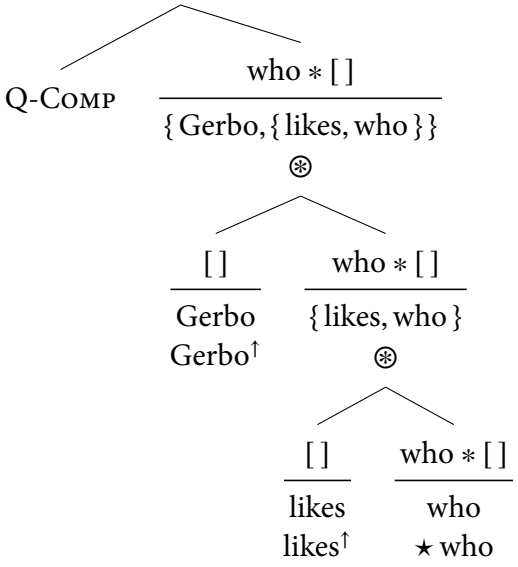
We can think of the syntactic contribution of the interrogative complementizer as an *instruction* to merge a syntactic head and LOWER.

(23) Q-COMP $m \coloneqq (\mathrm{C}_{\mathrm{Q}} \circledast m)^{\downarrow}$ $\quad \frac{\Bbbt}{\Bbbt} \to \Bbbt$

Higher-order structure building may now proceed as in (44). Note that the resulting syntactic representation is the same as that produced by the copy theory of movement.

(24) Who does Gerbo like?

{who, {C_Q, {Gerbo, {likes, who}}}}



2.2 Successive cyclic movement

There is overwhelming evidence that movement is successive-cyclic (see, e.g., Mccloskey 2002) – namely, that it leaves behind copies in intermediate positions. The machinery we need for successive-cyclic movement is already implicit in Lower. For convenience, we define an Intermediate Merge function in (25). Intermediate Merge lowers its argument, thereby evaluating the movement side-effect, while also *retaining* the movement side-effect.

(25) Intermediate Merge (def.)

a. $\Updownarrow m \coloneqq \lambda k \,.\, m\,(\lambda \mathbb{X} \,.\, m^{\downarrow})$ $\quad ::= \frac{\Updownarrow}{\Updownarrow} \rightarrow \frac{\Updownarrow}{\Updownarrow}$

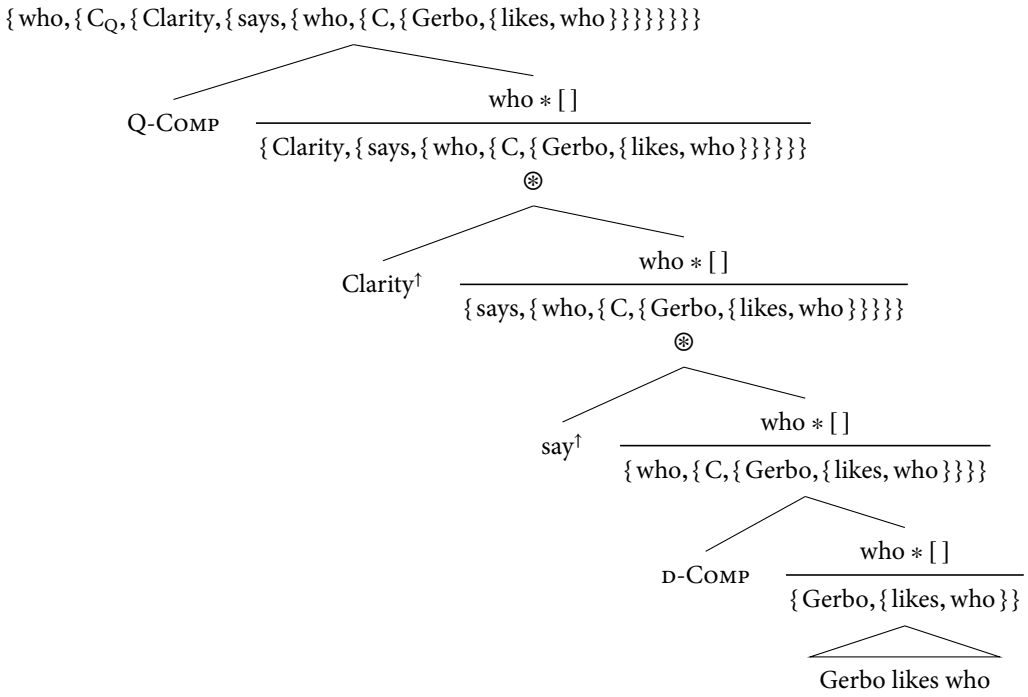
b. $\Updownarrow \frac{f\,[\,]}{\mathbb{X}} \coloneqq \frac{f\,[\,]}{f\,\mathbb{X}}$

We can treat an intermediate (i.e. declarative) complementizer as an *instruction* to insert a syntactic head, and trigger Intermediate Merge. If there are no unevaluated movement side-effects present, this is vacuous.

(26) d-Comp $m \coloneqq \Updownarrow (\mathrm{C} \circledast m)$

Here is an example derivation for successive-cyclic movement. We assume for the time being that the complement of *say* is a position at which Intermediate Merge is triggered.

(27) Who did Clarity say that Gerbo likes?



2.3 Incorporating a basic feature calculus

Let's assume that movement operations, such as *wh-movement*, are driven by the presence of unvalued features on the to-be-moved elements. For example, let's assume that *wh*-expressions carry an unvalued *Q* feature: $\text{who}_{[uQ]}$ (Pesetsky & Torrego 2007). We can redefine s-Merge as an operation that deletes unvalued features on the contained value.

(28) s-Merge (feature-sensitive version)
$\star\mathbb{X}_{[uF]} \coloneqq \lambda k \,.\, \mathbb{X}_{[uF]} * (k\ \mathbb{X})$

We can furthermore redefine Merge as feature sensitive, by building a limited form of specifier-head agreement into the definition. Concretely, if one of the sos to be merged carries as uninterpretable *Q* feature, the result is undefined unless a daughter of the other so to be merged carries an interpretable *Q* feature. In order to account for successive-cyclic movement, we can assume that declarative complementizers carry a feature $[intQ]$ that triggers deletion of $[uQ]$.

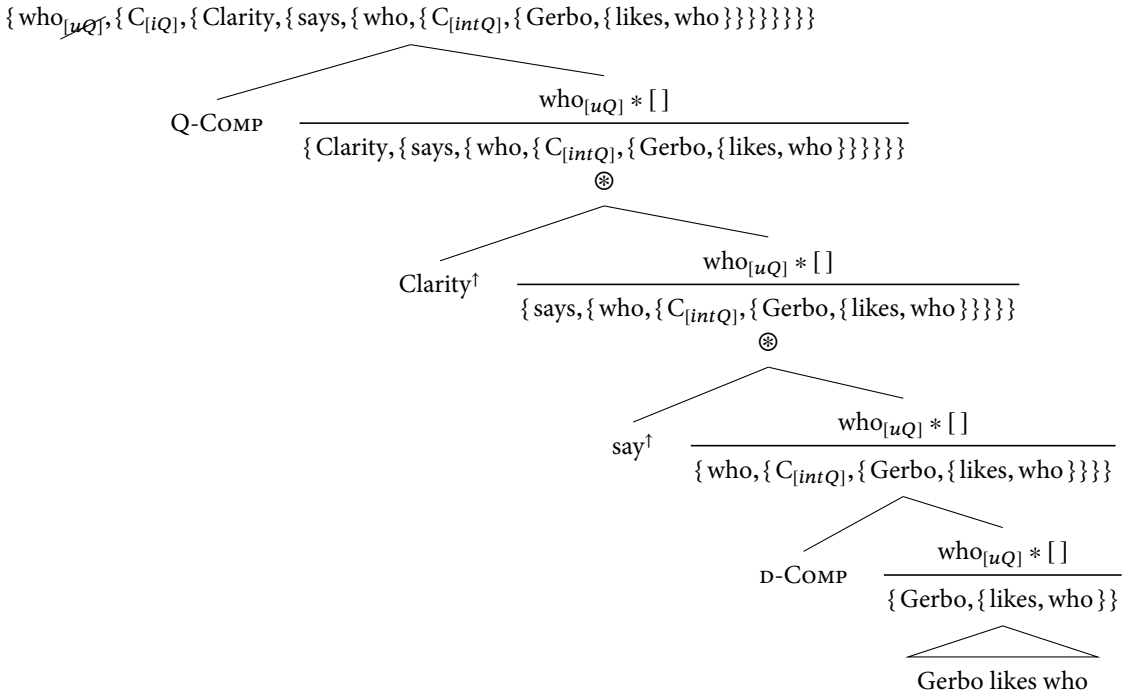
(29) feature-sensitive Merge (def.)

a. $\mathbb{X}_{[uQ]} * \{\mathbb{Y}_{[iQ]}, \_\} \coloneqq \{\mathbb{X}_{[\cancel{uQ}]}, \{\mathbb{Y}_{[iQ]}, \_\}\}$
b. $\mathbb{X}_{[uQ]} * \{\mathbb{Y}_{[intQ]}, \_\} \coloneqq \{\mathbb{X}, \{\mathbb{Y}_{[intQ]}, \_\}\}$
c. $\mathbb{X}_{[uQ]} * \mathbb{Y} \coloneqq \sharp$
d. $\mathbb{X} * \mathbb{Y} =\coloneqq \{\mathbb{X}, \mathbb{Y}\}$

(30) D-Comp $m \coloneqq \ddagger(C_{[intQ]} \circledast m)$

(31) Q-Comp $m \coloneqq (C_{[iQ]} \circledast m)^{\downarrow}$ $\quad \frac{\Updownarrow}{\Updownarrow} \to \Updownarrow$

(32) Who did Clarity say that Gerbo likes?



2.4 Generalized order-preservation

Order preservation effects are pervasive in syntax (Müller 2001), e.g., *superiority effects* in English and multiple *wh*-fronting languages such as Bulgarian.

(33) a. I wonder [who^{x} t_x bought what]

b. * I wonder [what^{y} who bought t_y]

(34) a. Whom^{x} did John persuade t_x to visit whom.

b. * Whom^{y} did John persuade whom to visit t_y.

(35) a. *Koj kakvo kupuva?*
who what buys

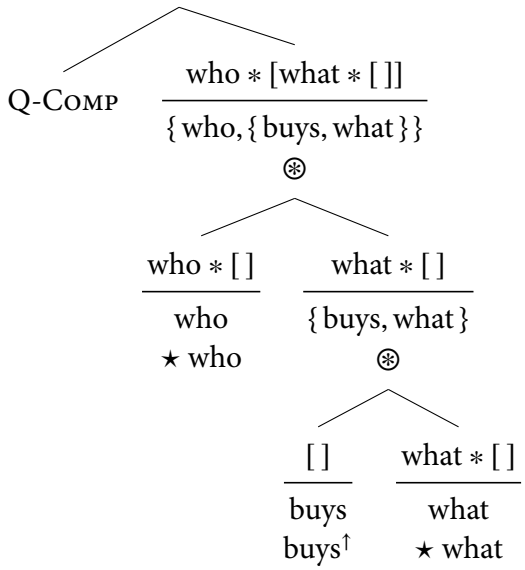
b. * *Kakvo koj kupuva?*
what who buys

"Who buys what?"

Order-preservation falls out as the unmarked case given the system outlined here. Consider again the rule for HIGHER ORDER MERGE. Movement effects are *sequenced* from left-to-right.

(36) $$\frac{f\,[\,]}{\mathbb{X}} \circledast \frac{g\,[\,]}{\mathbb{Y}} := \frac{f\,[\,g\,[\,]\,]}{\mathbb{X} \circledast \mathbb{Y}}$$

(37) $\{\text{who}, \{\text{what}, \{\text{who}, \{\text{buys}, \text{what}\}\}\}\}$



2.5 The interpretive component

As we'll see, this new view on structure-building will allow us to assign a single meaning to *who* which composes in tandem with syntactic operations, scoping the *wh*-expression at exactly the position it moves to.

2.5.1 Composing scopal meanings

The entry we'd like to suggest for *wh*-expressions is provided in (38). The *wh*-expression scopes over a question meaning and returns...a question meaning (see, e.g., Cresti (1995), Charlow (2017)).

(38) $[\![\text{which boy}]\!] := \lambda k\,.\,\bigcup_{\text{person } x} k\,x$ $\qquad ::= (\mathsf{e} \to \{\mathsf{t}\}) \to \{\mathsf{t}\}$

How do scopal meanings such as (38) compose with ordinary semantic values? First note that we can rewrite types of the form $(a \to \{t\}) \to \{t\}$ using Barker & Shan's tower notation.

(39) $\frac{\{\mathsf{t}\}}{\mathsf{a}} \coloneqq (\mathsf{a} \to \{\mathsf{t}\}) \to \{\mathsf{t}\}$

We can now define a function for lifting values into *trivially scopal* values (40), and additionally *scopal function application* (41).

(40) $x^{\rho} \coloneqq \lambda k \,.\, k\ x$

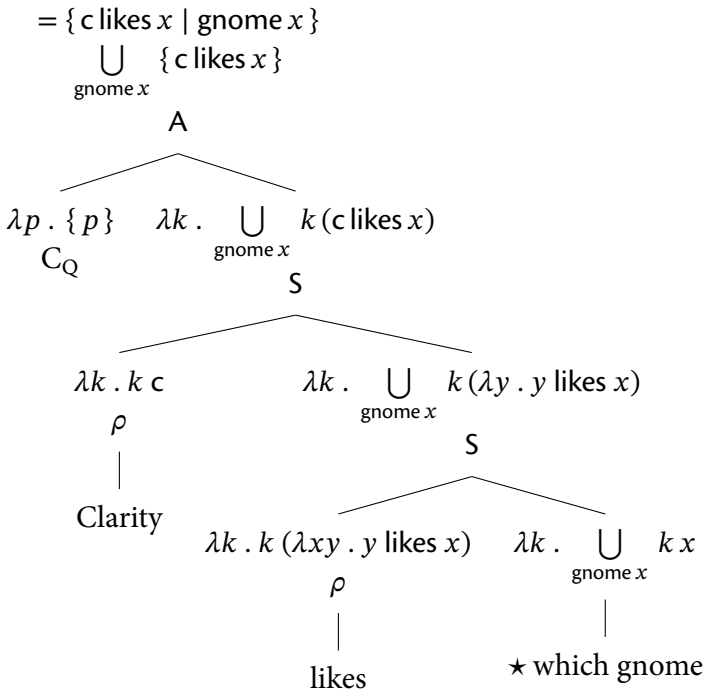
(41) $m\ \mathsf{S}\ n \coloneqq \lambda k \,.\, m\ (\lambda x \,.\, n\ (\lambda y \,.\, k\ (\mathsf{A}\ x\ y)))$

We take the interrogative complementizer to have its classical meaning (after Karttunen 1977) – it takes a proposition, and returns a question meaning.

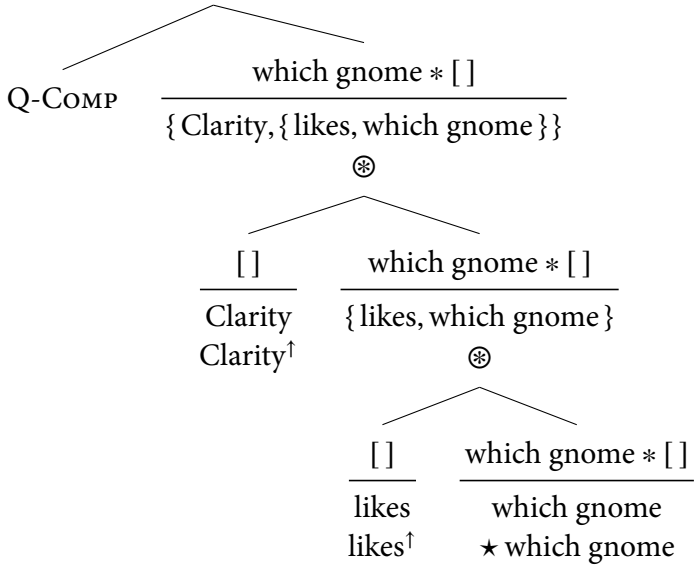
(42) $[\![\mathrm{C_Q}]\!] \coloneqq \lambda p \,.\, \{p\}$

Intuitively, ρ corresponds to $\uparrow$ on the syntactic side, and S corresponds to $\circledast$. Each derivational operation corresponds to a semantic operation. Here are the semantic and syntactic derivations of *wh*-movement, one after the other.

(43)



(44) $\{\text{which gnome}, \{C_{[uQ]}, \{\text{Clarity}, \{\text{likes}, \text{which gnome}\}\}\}\}$



References

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language* (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Charlow, Simon. 2014. *On the semantics of exceptional scope*.

Charlow, Simon. 2017. The scope of alternatives: Indefiniteness and islands. lingbuzz/003302.

Chomsky, Noam. 1995. *The minimalist program* (Current Studies in Linguistics 28). Cambridge Massachussetts: The MIT Press. 420 pp.

Cresti, Diana. 1995. Extraction and reconstruction. *Natural Language Semantics* 3(1). 79–122.

Elbourne, Paul. 2001. On the semantics of pronouns and definite articles. In *Proceedings of the 18th West Coast Conference on Formal Linguistics*, 164–177. Somerville, MA: Cascadilla Proceedings Project.

Fiengo, Robert & Robert May. 1994. *Indices and identity* (Linguistic Inquiry Monographs). Cambridge Massachussetts: The MIT Press. 340 pp.

Fox, Danny. 2002. Antecedent-contained deletion and the copy theory of movement. *Linguistic Inquiry* 33(1). 63–96.

Fox, Danny & Kyle Johnson. 2016. QR is restrictor sharing. In Kyeong-min Kim et al. (eds.), *Proceedings of the 33rd West Coast Conference on Formal Linguistics*, 1–16. Somerville, MA: Cascadilla Proceedings Project.

Heim, Irene. 1994. Lecture notes for semantics proseminar. Unpublished lecture notes.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar* (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.

Karttunen, Lauri. 1977. Syntax and semantics of questions. *Linguistics and Philosophy* 1(1). 3–44.

Mccloskey, James. 2002. Resumption, successive cyclicity, and the locality of operations. In *Derivation and explanation in the minimalist program*, 184–226. John Wiley & Sons, Ltd.

Müller, Gereon. 2001. Order preservation, parallel movement, and the emergence of the unmarked. In Géraldine Legendre et al. (eds.), *Optimality-theoretic syntax* (Language, speech, and communication). Cambridge, Mass: MIT Press.

Pesetsky, David & Esther Torrego. 2007. The syntax of valuation and the interpretability of features. In Simin Karimi, Vida Samiian & Wendy K. Wilkins (eds.), *Linguistik aktuell/linguistics today*, vol. 101, 262–294. Amsterdam: John Benjamins Publishing Company.

Sauerland, Uli. 1998. *On the making and meaning of chains*. Massachussetts Institute of Technology dissertation.

Sauerland, Uli. 2004. The interpretation of traces. *Natural Language Semantics* 12(1). 63–127.