Chapter Six: Minimal Semantic Instructions

Most of the last chapter was devoted to describing the concepts that are fetched and assembled, on my view, when meanings are executed. In this chapter, I provide some compositional details regarding the instructions themselves. My plan is to sketch a familiar kind of "minimalist" account of how human linguistic expressions are generated, show how these expressions can be viewed as instructions for how to assemble concepts, and specify the instructions in enough detail to make it clear how they can be executed for some textbook constructions. At least for these constructions, I think my account preserves the main insights that have been obtained by developing truth-theoretic conceptions of linguistic meaning. My hope is that working through some elementary cases will also indicate how insights regarding other constructions—described in the vast literature that describes the meanings of human I-language expressions in truth-theoretic terms—can be redescribed in the "instructionist" terms described here.

In section one, I discuss untensed clauses like (1) and (2),

(1) a spy stab a colonel with a knife
(2) see Scarlet stab Mustard

in order to show how such clauses can be instructions for how to build monadic concepts of events or states that have participants, some which can be described as "external participants" (e.g., agents) or "internal participants" (e.g., patients). Section two focuses on tensed sentences like (3-5) and relative clauses like (6).

(3) Scarlet stabbed Mustard.
(4) Peacock saw me stab him.
(5) Scarlet didn't stab Mustard.
(6) who she stabbed

Sentences can be analyzed as instructions for how to build monadic concepts—of a special sort that interface with other cognitive systems in a distinctive way—without invoking concepts of truth values. This requires a little more than M-junction and Θ-junction. But given this account of sentences, and a slight tweak on standard treatments of pronouns as in (4), relative clauses can also be accommodated. And while there is no avoiding appeal to something like quantification over assignments of value(s) to variables, such quantification—in a massively monadic me(n)tal language—is much less powerful than standard appeals to lambda abstraction.

This paves the way for a treatment of quantificational constructions like (7-9).

(7) Every guest stabbed him.
(8) She stabbed every guest.
(9) Most guests stabbed every spy.

As discussed in section three, a phrase like 'every spy' can be analyzed as an instruction for how to build a *plural* monadic concept of things—perhaps things as simple as ordered pairs—whose internal participants are the relevant spies. I assume that at least many quantificational expressions are grammatically displaced (or "raised"), and that such expressions recombine with sentential expressions that contain a trace of the displacement. But in my view, displaced quantifiers *do not* combine with expressions of type <e, **t**> that are formed by abstraction on expressions of type <**t**>: this standard approach to examples like (7-9) obscures important parallels with (1-4); and it effectively abandons the idea that meanings compose in accord with substantive constraints, replacing this idea with the much weaker claim that expressions have recursively specifiable semantic properties. Correlatively, I think my proposal—which might be

described as a neo-Medieval treatment of quantification—offers a better description of the much discussed "conservativity" constraint on determiners like 'every' and 'most'. It also handles (10),

(10) every guest who stabbed him

which cannot be understood as a sentence with the meaning of (7). This *un*ambiguity is puzzling if 'every spy' is of type <<e, **t**>, **t**> and 'who stabbed him' is of type <e, **t**>. So not only can standard accounts of quantification be recoded. the proposed recoding is in some respects better.

1. Nouns, Verbs, and Simple Phrases

The generable expressions of a human I-language connect meanings with pronunciations. These expressions, formed by combining lexical items in certain ways, also exhibit syntactic properties that can be described independently of the associated semantic and phonological properties; though of course, the "autonomous" syntactic structures of human linguistic expressions constrain what the expressions can mean, and how they can be pronounced. Like many others, I think the meaning of a complex expression is tightly connected to the form of the expression, modulo the arbitrary character of lexical meanings. But in defending a conception of meaning, one must beware of positing implausibly powerful expression-generating capacities that are tailored to the conception. (Similarly, in defending a conception of syntax, one must beware of positing implausibly powerful operations of semantic composition.) So my aim is show that my proposed account of meaning is plausible given spare expression-generating capacities—and ideally, only the capacities assumed by any adequate conception of human languages; cp. Chomsky (1995, 2000a). As it happens, I think "minimalist" theories of syntax are independently plausible, on theoretical and empirical grounds; see, e.g., Hornstein (2001, 2009), Uriagereka *et. al*. (200x), Berwick *et.al*. (2011). But the main point here is that minimal expression-generating capacities suffice for my proposed semantics. Though if the net result is a descriptively adequate conception of syntax and semantics, with a spare syntax and semantics taking in each other's wash, that would be a nontrivial argument for the component theories.[1]

1.1 Building Simple Instructions

If human I-languages generate unboundedly many expressions from finitely many lexical items, then such I-languages are procedures that include at least one operation that somehow supports recursive combination of generable expressions. It must be possible to combine lexical items in a way that forms a "unit" that can combine with at least one lexical item to form another such unit. One can invent a language in which any two expressions can be "unified" in a way that is adequately represented with set formation: given any two expressions Σ and Σ', UNIFY(Σ, Σ') = $\{\Sigma, \Sigma'\}$; so given any expression Σ'', UNIFY($\{\Sigma, \Sigma'\}$, Σ'') = $\{\{\Sigma, \Sigma'\}, \Sigma''\}$; etc. But obviously, human I-languages are not so permissive. Words exhibit categories that correspond to restrictions on what can combine with what; see, e.g., Baker (2003). Moreover, phrases are *headed*. Combining 'brown' with 'cow', for example, yields a phrase that is importantly more like 'cow' than 'brown'. So I assume that at least some expressions/meanings are built up in way that goes beyond the "super spare" operation UNIFY.

[1] See Hornstein and Pietroski (2009) for extended discussion. Correlatively, I understand the Minimalist Program broadly, not merely as an attempt to simplify extant conceptions of syntax. Like many authors—e.g., Fox (1999), Borer (2005), Jackendoff (2002), Ramchand (2008)—I assume that goal is to simplify syntax *and* semantics, as opposed to minimizing one at the cost of complicating the other.

Following Chomsky (1995, 2000a) and others, one might posit a basic asymmetric operation MERGE such that MERGE(Σ, Σ′) is either {{Σ, Σ′}, Σ} or {{Σ, Σ′}, Σ′}; where the singleton member of the set is the label of the expression formed by combining Σ with Σ′. Everything I say in this chapter can be recast in these terms. But it will helpful to adopt a slightly different proposal that has some independent attractions. Instead of taking MERGE to be a basic operation, I adopt Hornstein's (2009) idea that Chomsky's posited operation reflects two simpler operations: a super spare operation like UNIFY, and an operation LABEL that can apply to results of the first operation.[2] The idea is that MERGE(Σ, Σ′) = LABEL(UNIFY(Σ, Σ′)) = LABEL({Σ, Σ′}); where this result is either {{Σ, Σ′}, Σ} or {{Σ, Σ′}, Σ′} depending on which constituent has the properties required to "head" the unit {Σ, Σ′}. If Σ is the noun 'cow', and Σ′ is the adjective 'brown', then LABEL({Σ, Σ′}) = {{'cow', 'brown'}, 'cow'}. But this set, identical with {{'brown', 'cow'}, 'cow'}, can also be described in a more familiar way: $[\text{brown}_A\ \text{cow}_N]_N$.

This leaves room for the hypothesis that lexical categories are themselves functional expressions that can combine with acategorial *roots*. Perhaps the noun 'cow' is a complex expression, $[\sqrt{\text{cow}}_{\ N}]_N$, whose label is literally its functional constituent.[3] And perhaps as Chametsky (199x) argues, there are cases of labeled expressions combining with unlabeled adjuncts. For simplicity, I'll assume that the indefinite article in 'a cow' and the plural marker in 'cows' are unlabeled atomic constituents of complex nouns, as in $[\text{a cow}_N]_N$ and $[\text{cow}_N\ \text{s}]_N$. But it won't matter if these expressions are described as determiner phrases like $[\text{a}_D\ \text{cow}_N]_D$ or number phrases like $[\text{cow}_N\ \text{s}_\#]_\#$; cp. Munn (199x)

The expression $[\text{brown}_A\ \text{cow}_N]_N$ connects a certain meaning, $\mu([\text{brown}_A\ \text{cow}_N]_N)$, with a certain pronunciation, $\pi([\text{brown}_A\ \text{cow}_N]_N)$. I take the meaning of the complex expression to be a complex instruction, leaving it open how the structure of a meaning (or "semantic instruction") is related to the structure of a pronunciation (or "phonological instruction"). Even two-word examples—e.g., 'brown toast', which is ambiguous as between the noun phrase $[\text{brown}_A\ \text{toast}_N]_N$ and the verb phrase $[\text{brown}_V\ \text{toast}_N]_V$—illustrate potential complexities. But as this kind of notation highlights, a nonatomic expression has at least three potentially significant components: two constituents and a label. So we can ask how the meaning of a complex expression is related to the cognitive significance of lexical items, unifying, and labeling.

Unsurprisingly, I think the significance of unifying is conjunctive. My suspicion is that labeling lets humans use old forms of conjunction in new cognitively fruitful ways; see Hornstein and Pietroski (2009). But speculations about natural history aside, it is useful to think about phrases as products of the operations UNIFY and LABEL. As we'll see, this provides a way of describing the conjunctive aspects of phrasal meanings while recognizing that often, the significance of phrasal syntax is not merely conjunctive.

---

[2] Hornstein speaks of concatenation rather than unification. The difference is purely terminological. But since I am not concerned with pronunciation here, I want to avoid any suggestion of *linear* as opposed to *compositional* order.

[3] Perhaps $_N$ is a device for fetching a functional monadic concept like INDEXABLE(_), while $_V$ is a device for fetching a concept like TENSABLE(_), thus allowing for a distinction between $\mu(\text{chase}_V)$ and $\mu(\text{chase}_N)$. I won't pursue this suggestion, but simply raise it to note the resources available *without* appeal to Fregean typology; see Hornstein and Pietroski (2010), drawing on Marantz (1984), Halle and Marantz (1993), Baker (2003), and Borer (2005). See also Harley (200x, 200x)

1.2 Specifying Simple Instructions

Recall that **M-join**(**I**, **I′**) is a "macro" instruction that is executed by executing the two subinstructions and M-joining the results. Executing **M-join**(**I**, **I′**) yields a concept of the form Φ(_)^Ψ(_). Likewise, executing **Θ-join**(**I**, **I′**) yields a concept of the form ∃[Θ(_, _)^Ψ(_)]. A subinstruction can be of the for ***fetch***@L; where executing such an instruction accesses a monadic or dyadic concept from the lexical address L. But a subinstruction can also be of the form **M-join**(**I**, **I′**) or **Θ-join**(**I**, **I′**). My proposal is that simple phrases are instructions for how to build monadic concepts in these simple ways. But the details depend on which concepts can be accessed with which lexical items, and exactly how labeling is related to meaning.

Some complex expressions at least approximate the classical/medieval ideal of a conjunctive monadic predicate. For example, $[\text{cow}_N\ \text{s}]_N$ arguably corresponds to the concept COW(_)^PLURAL(_); where as described at the end of the last chapter, COW(_) is a number-neutral concept. So $[\text{cow}_N\ \text{s}]_N$ can perhaps be described as the instruction **M-join**(μ(cow_N), μ(s)). And if 'cow that Fido chased' has the structured displayed with $[\text{cow}_N\ [\text{that Fido chased}]_C]_N$—ignoring structure within the relative clause labeled 'C'—then the meaning of this noun phrase can perhaps be described as **M-join**(μ(cow_N), μ($[\text{that Fido chased}]_C$)). Of course, this highlights the need for an account of how μ($[\text{that Fido chased}]_C$) can be an instruction for how to build a mental predicate. But it also highlights the fact that the significance of combining a noun with an adjective is rarely if ever *merely* conjunctive.

If we idealize away from the fact that a brown cow is a cow that is brown *for a cow*, we might say that executing μ(brown_A) accesses or assembles a monadic concept BROWN(_), which can be M-joined with COW(_). And perhaps μ($[\text{brown}_A\ \text{cow}_N]_N$) can be correctly described as the instruction **M-join**(μ(brown_A), μ(cow_N)). Though executing μ(brown_A) might access or assemble a dyadic concept BROWN-FOR(_, _); where this concept applies to one more things (e.g., a cow) and some things (e.g., the cows) iff the former is/are both among the latter and brown instances of the latter. So perhaps the syntax of 'brown cow' is more like $[[\text{brown-}\textit{one}]_A\ \text{cow}_N]_N$; where the meaning of the complex adjective $[\text{brown-}\textit{one}]_A$ is an instruction for how to build a concept like ∃[BROWN-FOR(_, _)^THE-COWS(_)]. In which case, μ($[[\text{brown-}\textit{one}]_A\ \text{cow}_N]_N$) could be the instruction **M-join**(μ($[\text{brown-}\textit{one}]_A$), μ($\text{cow}_N$)). But executing this instruction might yield a conjunctive but not merely conjunctive concept like ∃[BROWN-FOR(_, _)^THE-COWS(_)]^COW(_). Indeed, the concept formed by M-junction could have a constituent formed by Θ-junction.

Examples like 'big ant', 'toy truck', and 'fake diamond' are sometimes described as *non*-conjunctive. But this strikes me as misdescription. A big ant is still an ant that meets a further condition on its size. A fake diamond is a fake of a certain sort. And depending on the context, a toy truck is either a toy of a certain sort, or a truck of a certain sort. (See Higginbotham 198x.) So before concluding that any such example provides a special *difficulty* for the view presented here, we need to know what syntax is being presupposed and what alternative captures the meaning *better* given that syntax. If difficult cases raise questions about how to capture the not merely conjunctive aspects of meaning, then we may be better off trying to supplement an account of meaning that is conjunctive at its core, instead of starting with a non-conjunctive account and trying to capture all the apparently conjunctive aspects of meaning.

Given the history of semantics, it is tempting to describe the meaning of a verb phrase like ‘chase cows’ or ‘chase a cow’ in terms of saturation or function-application. But upon reflection, such phrases show signs of being conjunctive though not merely conjunctive. Events of chasing cows are chases, though of course, not chases that are also cows. And while the meaning of ‘chase a cow’ is presumably of the same type as the meaning of ‘chase Bessie’, we should not assume that ‘Bessie’ is of type <e>, given the considerations adduced in chapter five.

One can hypothesize that in ‘chase a cow’, ‘a cow’ is a quantificational expression that displaces, leaving a trace of type <e> that corresponds to the first saturater of a concept like CHASE(_, _, _) or PASSIVE-CHASE(_, _). But the availability of this hypothesis hardly shows that the meaning of ‘chase a cow’ is *non*-conjunctive. As discussed below, traces need not be described as expressions of type <e>, and ‘a’ need not be described as a quantifier. Moreover, just as a chase of a cow may be both a chase and one that was done by a dog, it may have been done to a cow. And even if there is no intuitive sense in which such a chase is “of a cow,” we can speak of the dog’s chase of the cow. If ‘chase a cow’ had the meaning that ‘a chase or a cow’ has, or the meaning of ‘eat a cow’, then we might well describe the significance of combining ‘chase’ with ‘a cow’ as non-conjunctive, much as the idiomatic significance of combining ‘kick’ with ‘the bucket’ is non-conjunctive. But in my view, ‘kick the bucket’ still has one meaning that is conjunctive though not merely conjunctive, and likewise for ‘chase a cow’.

Thinking about phrases as products of the operations UNIFY and LABEL suggests a way factoring the not merely conjunctive significance of combining a verb with a noun (phrase) into two components: a conjunctive operation invoked by UNIFY; and an “adaptive” operation—involving a thematic concept like PATIENT(_, _)—invoked by the formal relation that a verb bears to the noun it combines with and dominates. Again, the meaning of $[\text{chase}_V\ \text{cows}_N]_V$ is clearly *not* of the simple form **M-join**($\mu(\text{chase}_V)$, $\mu(\text{cows}_N)$), since executing $\mu(\text{cows}_N)$ yields a concept like COW(_)^PLURAL(_). But there is more than one potential alternative.

One option is to hypothesize, *a la* Kratzer, that executing $\mu(\text{chase}_V)$ accesses the dyadic PASSIVE-CHASE(_, _); cp. §2.4 of chapter five. On this view, $\mu([\text{chase}_V\ \text{cows}_N]_V)$ is the instruction **Θ-join**($\mu(\text{chase}_V)$, $\mu(\text{cows}_N)$), whose execution can yield a concept of events that are collectively chases of cows: ∃[PASSIVE-CHASE(_, _)^[COW(_)^PLURAL(_)]]. This treats verbs as relevantly like the prepositions in $[\text{above}_P\ \text{cows}_N]_P$. I assume that prepositions can be used to access dyadic concepts, and that the meaning of $[\text{above}_P\ \text{cows}_N]_P$ is an instruction whose execution can yield a concept like ∃[ABOVE(_, _)^[COW(_)^PLURAL(_)]]. But this doesn’t require a system that explicitly specifies the meanings of prepositional phrases in terms of Θ-junction and the meanings of plural nouns in terms of M-junction. One might characterize phrasal meanings as “neutral” instructions of the form **Join**(**I**, **I′**); where such an instruction can be executed by performing either of the more specific instructions **M-join**(**I**, **I′**) or **Θ-join**(**I**, **I′**).

If $\mu(\text{above}_P)$ can only be executed by fetching a dyadic concept, then the neutral instruction **Join**($\mu(\text{above}_P)$, $\mu(\text{cows}_N)$) can only be executed by Θ-joining the results of executing the two subinstructions. Likewise, If $\mu(\text{chase}_V)$ can only be executed by fetching a dyadic concept, then **Join**[$\mu(\text{chase}_V)$, $\mu(\text{cows}_N)$] can only be executed by Θ-joining the results of executing the two subinstructions. But if each subinstruction in **Join**($\mu(\text{cow}_N)$, $\mu(\text{s})$) can only be executed by fetching a monadic concept, then the macro-instruction can only be executed by M-joining the results of executing $\mu(\text{cow}_N)$ and $\mu(\text{s})$.

In the end, perhaps we will conclude that verbs do access dyadic concepts, and that the significance of combining a verb with direct object is *like* saturation to this extent: the result has a lower adicity than one of the inputs. But I don't want to *assume* that verbs access dyadic concepts, rather than monadic concepts like CHASE(_). Moreover, we eventually have to deal with 'saw a dog chase cows' and its constituent 'a dog chase cows', whose meaning is not an instruction of the form **Join**[μ($dogs_N$), μ([$chase_V$ $cows_N$]$_V$)]. So let me present another option, according to which μ([$chase_V$ [a cow_N]$_N$]$_V$) is an instruction whose execution can yield a concept like CHASE(_)^∃[PATIENT(_, _)^COWS(_)], which can be M-joined with ∃[AGENT(_, _)^DOG(_)].

Suppose that if expressions with the labels L and L* can be combined—yielding an expression of the form […$_L$, … $_{L*}$]$_L$—then the label-pair L/L* determines a concept like PATIENT(_, _) or perhaps IDENTICAL(_, _).[4] The idea is that the labeling in an expression of the form […$_L$, … $_{L*}$]$_L$ can serve as instruction to access a *third* concept, which can be described as the L/L*-adapter, in addition the concepts accessed or assembled by executing the constituent instructions μ(…$_L$) and μ(…$_{L*}$). Obviously, one wants to limit the number of such "adapter concepts." But for simplicity, let's suppose that each label-pair also serves as the address for some such concept, allowing for the possibility that a single concept has several addresses of the form 'L/L*'. Then one can posit a general composition principle (11)

(11) μ([…$_L$, … $_{L*}$]$_L$) = **Join**(μ(…$_L$), **Join**(***fetch*@**'L/L*', μ(…$_{L*}$)))

that has consequences like (12).

(12) μ([$chase_V$, $cows_N$]$_V$) = **Join**(μ($chase_V$), **Join**(***fetch*@**V/N, μ($cows_N$)))

From this perspective, combining expressions can call for M-junction *and* Θ-junction. Suppose the V/N-adapter is the dyadic concept PATIENT(_, _). Then the instruction **Join**(***fetch*@**V/N, μ($cows_N$)) can only be executed by Θ-joining PATIENT(_, _) with the result of executing μ($cows_N$), thereby creating a monadic concept like ∃[PATIENT(_, _)^COWS(_)]. So if μ($chase_V$) can only be executed by fetching a monadic concept, then μ([$chase_V$, $cows_N$]$_V$) must be executed via M-junction, yielding a concept like CHASE(_)^∃[PATIENT(_, _)^COWS(_)].

Of course, (11) also implies (13).

(13) μ([$above_P$, $cows_N$]$_P$) = **Join**(μ($above_P$), **Join**(***fetch*@**'P/N', μ($cows_N$)))

But this may not be a bad thing. Svenonius (2010, 201x) argues that at least in some languages, prepositions correspond to monadic concepts of states, and that the preposition/verb distinction is not so clear; see also Schein (2014). But (13) may be correct even if prepositions like $above_P$ fetch dyadic concepts like ABOVE(_, _). For one could say that in languages where prepositions access dyadic concepts, the label-pair P/N determines an "empty" concept like IDENTICAL(_, _) or SELF-IDENTICAL(_), so that executing **Join**(***fetch*@**'P/N', μ($cows_N$)) yields a monadic concept like ∃[IDENTICAL(_, _)^COWS(_)] or SELF-IDENTICAL(_)^COWS(_) or some other concept that is analytically equivalent to COWS(_). This allows for a kind of variation in *how* relationality is expressed—via certain lexical items, or via thematic adapters associated with certain labels.

Likewise, *if* verbs access dyadic concepts in some but not all human languages, then one might say that the label-pair P/N has no real (as opposed to merely formal) significance in some but not all human languages. I worry about introducing this kind of freedom and assuming that children somehow "figure out" how relationality is expressed in a particular language. But

[4] This does not preclude the possibility that L = L*.

there is whatever variation there is. And my point here is simply that for better or worse, my proposal allows for meanings/instructions that call for M-junction, Θ-junction, or both. So for better or worse, my proposal allows for more than one conception of verbs and prepositions.

If the significance of combining expressions is ever *merely* conjunctive, then positing at least one "empty" concept will be unavoidable given (11), which also implies (14).

(14) $\mu([\text{cow}_N\ [\text{that Fido chased}]_C]_N)$ =

Join($\mu(\text{cow}_N)$, **Join**(***fetch*@**N/C, $\mu([\text{that Fido chased}]_C)$))

But again, one can say the label-pair N/C determines a concept like IDENTICAL(_, _). If the plural marker in 'cows' has no label, then (11) does not apply. And we cay say that if at least one constituent of a complex expression is not labeled, the significance of the combination is mere M-junction, so that $\mu([\text{cow}_N\ \text{s}]_N)$ = **M-join**($\mu(\text{cow}_N)$, $\mu(\text{s})$). But if the plural marker has a label, as shown in (15), then perhaps the #/N-adapter is purely formal,

(15) $\mu([\text{cow}_N\ \text{s}_\#]_\#)$ = **Join**($\mu(\text{cow}_N)$, **Join**(***fetch*@**#/N, $\mu(\text{s})$))

Obviously, M-junction can be viewed as a special case of Θ-junction:
M-joining the concepts Φ(_) and Ψ(_) is equivalent to M-joining Φ(_) with the result of Θ-joining Ψ(_) with IDENTICAL(_, _). So one might think of labels as a way of introducing the *possibility* of mismatching labels, and thereby injecting some non-conjunctive significance into *many but not all* instances of grammatical combination. Imagine a mind that can M-join each of its monadic concepts with some others, but only in limited ways, given various typological restrictions. Merely adding a capacity to Θ-join some monadic concepts with some dyadic concepts might not have a large cognitive impact. But given a capacity to lexicalize concepts in a way that introduces *neutral* concepts that can be systematically M-joined/Θ-joined, then it might *very* useful to have a system that (i) labels the resulting lexical items, which can be used to fetch the neutral concepts, and (ii) supports recursive combination of labeled expressions in a way makes it possible to use label pairs like V/N as devices for accessing concepts like PATIENT(_, _). and AGENT(_, _); see Hornstein (200x), Hornstein and Pietroski (2009).

1.3 Internals and Externals

However, a substantive V/N-adapter need not be the specific concept PATIENT(_, _). This would imply that executing the meaning of any phrase of the form $[\ldots_V, \ldots_N]_V$ yields a concept of the form Φ(_)^∃[PATIENT(_, _)^Ψ(_)]. But one might think that executing $\mu([\text{see}_V, \text{cows}_N]_V)$ can yield SEE(_)^∃[THEME(_, _)^COWS(_)]; where this concept can apply to events without implying that any cows were *affected*. And plausibly, ∃[PATIENT(_, _)^COWS(_)] only applies to events in which cows were affected. So if verbs access monadic concepts like SEE(_) and CHASE(_), then given the general composition principle (11),

(11) $\mu([\ldots_L, \ldots_{L^*}]_L)$ = **Join**($\mu(\ldots_L)$, **Join**(***fetch*@**'L/L*', $\mu(\ldots_{L^*})$))

it might be better to say that the V/N-adapter is the more formalistic concept INTERNAL(_, _); where events of chasing can have internal participants that are patients of those events, events of seeing can have internal participants that are themes of those events, the ordered pair ⟨2, 3⟩—a.k.a. {2, {2, 3}}—has 3 as its internal participant, and so on.

On this view, both $\mu([\text{chase}_V, \text{cows}_N]_V)$ and $\mu([\text{see}_V, \text{cows}_N]_V)$ access the concept INTERNAL(_, _). But CHASE(_)^∃[INTERNAL(_, _)^COWS(_)] can still differ thematically from SEE(_)^∃[INTERNAL(_, _)^COWS(_)] if CHASE(_) carries an implication that it only applies to

events whose internal participants are patients, while SEE(_) does carry this implication. One can say that CHASE(_) and SEE(_) are neutral with regard to many aspects of conceptual typologies, but that CHASE(_) and SEE(_) are still typologically distinct in some respects; cp. Vendler's (196x) much discussed hierarchy.

A more pressing concern is that whatever one says about the V/N-adapter has to be compatible with the facts concerning *transitive* clauses like (16).[5]

(16) a dog chase cows

As discussed in chapter five, (16) cannot be understood as (an instruction for how to build) a predicate of chases whose agents are cows and whose patient is a dog. But if executing $\mu([\text{chase}_V, \text{cows}_N]_V)$ can yield CHASE(_)^∃[INTERNAL(_, _)^COWS(_)]—with CHASE(_) licensing replacement of INTERNAL(_, _) with PATIENT(_, _)—then so far, so good. However, this view requires that (16) not have the simple structure $[[\text{a dog}_N]_N\ [\text{chase}_V, \text{cows}_N]_V]_V$. Otherwise, executing the meaning of this expression would yield a concept like (17),

(17) ∃[INTERNAL(_, _)^DOG(_)]^CHASE(_)^∃[INTERNAL(_, _)^COWS(_)]

which applies to a chase only if (a) its internal participant is a dog, *and* (b) its internal participants are cows. And that won't do. So if (16) has the simple structure, (11) is wrong.

(11) $\mu([\ldots_L, \ldots_{L^*}]_L) = \textbf{Join}(\mu(\ldots_L), \textbf{Join}(\textbf{\textit{fetch}@}\text{'L/L*'}, \mu(\ldots_{L^*})))$

One alternative is to say that (16) has the structure $[[\text{a dog}_N]_N\ [\text{chase}_{V2}, \text{cows}_N]_{V1}]_V$; where numerals indicate the number of arguments that an expression can still combine with. But not only does this recapitulate talk of "verbal valence," in a way at odds with the spirit of chapter five, it requires a more sophisticated conception of labeling. Another alternative, explored in Pietroski (2005a), would be to replace (11) with (11a) and (11b);

(11a) $\mu([\ldots_{L(\text{atomic})}, \ldots_{L^*}]_L) = \textbf{Join}(\mu(\ldots_L), \textbf{Join}(\textbf{\textit{fetch}@}\text{'int'}, \mu(\ldots_{L^*})))$

(11b) $\mu([\ldots_{L(\text{phrasal})}, \ldots_{L^*}]_L) = \textbf{Join}(\mu(\ldots_L), \textbf{Join}(\textbf{\textit{fetch}@}\text{'ext'}, \mu(\ldots_{L^*})))$

where chase_V counts as atomic instance of category V, and $[\text{chase}_V, \text{cows}_N]_V$ counts as a phrasal instance of the same category V, while 'int' and 'ext' are addresses for the formalistic concepts INTERNAL(_, _) and EXTERNAL(_, _). Then executing $\mu([[\text{a dog}_N]_N\ [\text{chase}_V, \text{cows}_N]_V]_V)$ could yield (18), with CHASE(_) licensing replacement of EXTERNAL(_, _) with AGENT(_, _).[6]

(18) ∃[EXTERNAL(_, _)^DOG(_)]^CHASE(_)^∃[INTERNAL(_, _)^COWS(_)]

A third option is to say that (16) has the structure $[[\text{a dog}_N]_N\ [__v\ [\text{chase}_V, \text{cows}_N]_V]_v]_v$; where $[\text{chase}_V, \text{cows}_N]_V$ combines with a covert "small verb" to form a phrase that can combine with $[\text{a dog}_N]_N$. On this increasingly common view, chase_V itself only combines with a single grammatical argument. But one need not assign any significance to the covert verb, which can be viewed as a functional item whose role is to support the second grammatical argument: it may access no concept, or least none that applies any independent restriction; and the *v*/V-adapter may be the concept IDENTICAL(_, _). Then since (11) implies (19),

[5] I want to delay discussion of tense and complete sentences until section two.

[6] A variation on this suggestion is that "cyclicity" plays a role in choosing adapters: within a given "phase" of instruction *execution*, the innermost occurrence label 'V' accesses INTERNAL(E, X); the outermost occurrence of 'V' accesses EXTERNAL(E, X). If there are at most two grammatical arguments per phase/cycle/whatever, one might imagine a binary "switch" that gets "reset" to its initial state at the start of each cycle; cp. Boeckx (2008).

(19) $\mu([__v\ [\text{chase}_V, \text{cows}_N]_V]_v) =$

$\textbf{Join}(\mu(__v), \textbf{Join}(\textbf{\textit{fetch}}@v/V, \mu([\text{chase}_V, \text{cows}_N]_V)))$

executing $\mu([__v\ [\text{chase}_V, \text{cows}_N]_V]_v)$ could yield a concept like (20),

(20) $\exists[\text{IDENTICAL}(_, _)\text{^}\text{CHASE}(_)\text{^}\exists[\text{INTERNAL}(_, _)\text{^}\text{COWS}(_)]]$

which is equivalent to $\text{CHASE}(_)\text{^}\exists[\text{INTERNAL}(_, _)\text{^}\text{COWS}(_)]$. In this sense, the covert small verb might do nothing except provide a new label for $[\text{chase}_V, \text{cows}_N]_V$. But this is still doing something if the *v*/N-adapter is $\text{EXTERNAL}(_, _)$. For in that case, given that (11) implies (21),

(21) $\mu([[\text{a dog}_N]_N\ [__v\ [\text{chase}_V, \text{cows}_N]_V]_v]_v) =$

$\textbf{Join}(\mu([__v\ [\text{chase}_V, \text{cows}_N]_V]_v), \textbf{Join}(\textbf{\textit{fetch}}@v/N, \mu([\text{a dog}_N]_N)))$

executing $[[\text{a dog}_N]_N\ [__v\ [\text{chase}_V, \text{cows}_N]_V]_v]_v$ can yield (18).

Given a genuinely causative construction like (22),

(22) a child break vase

whose verb can appear in the intransitive construction (23),

(23) a vase break

one might posit a more substantive "light verb" as in (22a).

(22a) $[[\text{a child}_N]_N\ [\textit{termi}_v\ [\text{break}_V, [\text{a vase}_N]_N]_V]_v]_v$

Suppose the covert verb $\textit{termi}_v$ accesses a concept that applies to ⟨e1, e2⟩ iff e1 is a causal process *terminates in* event e2; where a process of breaking a vase can terminate in an event that is a vase breaking.[7] Then executing $\mu([\textit{termi}_v\ [\text{break}_V, [\text{a vase}_N]_N]_V]_v)$ might yield (23),

(23) $\exists[\text{TERMINATER}(_, _)\text{^}[\text{CHASE}(_)\text{^}\exists[\text{INTERNAL}(_, _)\text{^}\text{COWS}(_)]]]$

to which $\exists[\text{EXTERNAL}(_, _)\text{^}\text{CHILD}(_)]$ can be M-joined. So the meaning of (22) can be specified as $\textbf{Join}(\mu([\text{a child}_N]_N), \textbf{Join}(\mu([\textit{termi}_v\ [\text{break}_V, [\text{a vase}_N]_N]_V]_v))$.

With this in mind, let's return to the idea that executing $\mu(\text{chase}_V)$ accesses the dyadic concept $\text{PASSIVE-CHASE}(_, _)$. Initially, one might think this option—which lets one say that $\mu([\text{chase}_V, \text{cows}_N]_V) = \textbf{Join}(\mu(\text{chase}_V), \mu(\text{cows}_N))$—is simpler than any alternative according to which executing $\mu(\text{chase}_V)$ accesses a monadic concept. But if executing $\mu([\text{chase}_V, \text{cows}_N]_V)$ yields $\exists[\text{PASSIVE-CHASE}(_, _)\text{^}\text{COWS}(_)]$, then to accommodate (22) and (16),

(16) a dog chase cows

one still needs some principle according to which an external argument like 'dog' can correspond to a conjunct like $\exists[\text{EXTERNAL}(_, _)\text{^}\text{DOG}(_)]$. One can, of course, assign meanings of a suitably high Fregean type to the covert verbs; perhaps (16) indicates the function (24),

(24) $\lambda E.T \equiv \exists X[\text{EXTERNAL}(E, X)\text{^}\text{DOG}(X)]\text{^}\exists X[\text{PASSIVE-CHASE}(E, X)\text{^}\text{COWS}(X)]$

and the covert verb indicates a function that maps $\lambda E.T \equiv \exists X[\text{PASSIVE-CHASE}(E, X)\text{^}\text{COWS}(X)]$ to a higher-order function that maps $\lambda X.T \equiv \text{DOG}(X)$ to (24). Or one can simply posit a composition principle for external arguments that has the same effect; see Kratzer (1996) appeal on "event identification." But this is not simpler that uniform appeal to M-junction and Θ-junction.

Let me note, however, that appealing to M-junction and thematic concepts effectively permits "conversion" of concepts of dogs and cows into concepts of events in which dogs and cows participate. As noted in chapter one, this is some respects like more familiar appeals to type-shifting in a fundamentally "saturationist" account of semantic composition. One might say

[7] See Pietroski (2003b), drawing on a great deal of prior work, for presentation in a minimalist setting.

that in the absence of labeling, the meaning of a phrase with two constituents would be an instruction that is executed by executing the two subinstructions and saturating one result with the other. But suppose that $\mu(\text{brown}_A)$ and $\mu(\text{cow}_N)$ can only be executed by accessing the concepts $\lambda X.\top \equiv \text{BROWN}(X)$ and $\lambda X.\top \equiv \text{COW}(X)$. Then one might say that the labels in the phrase $[\text{brown}_A\ \text{cow}_N]_N$ matter, and that the label-pair N/A is an instruction to access the higher-order concept $\lambda\Psi.\lambda\Phi.\lambda X.\top \equiv \Psi(X)\ \&\ \Phi(X) = \top$. This concept of type <<e, t>, <<e, t>, <e, t>>> can be saturated by $\lambda X.\top \equiv \text{BROWN}(X)$ to yield $\lambda\Phi.\lambda X.\top \equiv \text{BROWN}(X)\ \&\ \Phi(X) = \top$; where this latter concept can be saturated by $\lambda X.\top \equiv \text{COW}(X)$ to yield $\lambda X.\top \equiv \text{BROWN}(X)\ \&\ \text{COW}(X) = \top$. So abstracting away from the brown-for-a-cow aspect of $\mu([\text{brown}_A\ \text{cow}_N]_N)$, one might specify this meaning as follows: **Saturate**(**Saturate**(***fetch*@**'N/A', $\mu(\text{brown}_A)$), $\mu(\text{cow}_N)$).

If only because this analysis of $[\text{brown}_A\ \text{cow}_N]_N$ invokes a concept at Level Three of the Fregean hierarchy, in addition to a concept of conjunction, it is not simpler than an analysis of $[\text{chase}_V\ \text{cow}_N]_V$ that invokes Θ-junction. But my proposal is analogous, in that COW(_) can be "Θ-adapted" to form the concept ∃[INTERNAL(_, _)^COW(_)]. So while $\mu([\text{chase}_V\ \text{cow}_N]_V)$ is executed by M-joining two monadic concepts, I don't want to disguise the appeal to Θ-junction.

That said, it may be possible to impose a further and rather severe restriction on (11).

(11) $\mu([\ldots_L, \ldots_{L*}]_L) = \textbf{Join}(\mu(\ldots_L), \textbf{Join}(\textbf{\textit{fetch}@}\text{'L/L*'}, \mu(\ldots_{L*})))$

Perhaps the only "Θ-adapters" are INTERNAL(_, _) and EXTERNAL(_, _) and IDENTICAL(_, _).

1.4 Adverbials and Perceptuals

Given that event analyses were designed to treat adverbs and prepositional phrases as conjuncts in complex predicates of events, it is not surprising that such expressions can be viewed as instructions for how to build concepts of events. If the meaning of (16)

(16) a dog chase cows

can be an instruction whose execution yields (18),

(18) ∃[EXTERNAL(_, _)^DOG(_)]^CHASE(_)^∃[INTERNAL(_, _)^COWS(_)]

then the meaning of (25) can be an instruction whose execution yields (26).

(25) a dog chase a cow today

(26) ∃[EXTERNAL(_, _)^DOG(_)]^CHASE(_)^∃[INTERNAL(_, _)^COW(_)]^TODAY(_)

As noted in §1.2 above, the meaning of $[\text{above}_P\ \text{cows}_N]_P$ can be an whose execution yields a concept like ∃[ABOVE(_, _)^COWS(_)]. Likewise, executing $\mu([\text{with}_P\ [\text{a knife}_N]_N]_P)$ can yield ∃[INSTRUMENT(_, _)^KNIFE(_)]. There need not be a single concept that the polysemous preposition 'with' accesses; cp. 'France' and 'book'. But if prepositions can be used to access thematic adapters, then clauses like (1) can be used to build concepts like (1a);

(1) a spy stab a colonel with a knife

(1a) ∃[EXTERNAL(_, _)^SPY(_)]^
STAB(_)^∃[INTERNAL(_, _)^COLONEL(_)]^∃[INSTRUMENT(_, _)^KNIFE(_)]

where STAB(_) licenses replacement of the formalistic EXTERNAL(_, _) and INTERNAL(_, _) with the more specific thematic concepts AGENT(_, _) and PATIENT(_, _). And if proper nouns are like phrases headed by common nouns—in accessing monadic concepts, albeit concepts that apply to only one thing—then clauses like (27) can be used to build concepts like (27a).

(27) Scarlet stab Mustard

(27a) ∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]

At this point, accommodating perceptual idioms like (2) is easy.

(2) see Scarlet stab Mustard

If (27) is the complement of 'see' in (2), then (2) is relevantly like (28), which implies (29),

(2) see a stab of Mustard that was done by Scarlet

(29) see a stab

The meaning of (29) can be described as an instruction whose execution can yield (29a).

(29a) SEE(_)^∃[INTERNAL(_, _)^STAB(_)]

Likewise, the meaning of (2) can be described as an instruction whose execution can yield (2a).

(2a) SEE(_)^∃[INTERNAL(_, _)^
∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]]

This concept applies to events of seeing whose internal participants (or themes) are events of Scarlet stabbing Mustard.

2. Tense, Sentences, and Relative Clauses

It still needs showing that the proposed account of untensed clauses can be extended to accommodate sentences like (3-6) and relativel clauses like (6).

(3) Scarlet stabbed Mustard.

(4) Peacock saw me stab him.

(5) Scarlet didn't stab Mustard.

(6) who she stabbed

This is not entirely trivial. But it does not require appeal to truth values, or to concepts that cannot be inputs to the basic composition operations of M-junction and Θ-junction.

2.1 Adding Tense

Adding a tense marker to (27)

(27) Scarlet stab Mustard

can be viewed as instruction to build a temporal concept and M-join it with a concept like (27a).

(27a) ∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]

Following Hornstein (198x) and others, I suspect that a concept of events like PAST(_) is best viewed as a complex Reichenbach-style concept, perhaps ∃[BEFORE(_, _)^SPEECH-TIME(_)]; where the concept SPEECH-TIME(_) is context sensitive concept and perhaps itself of the form ∃[Θ(_, _)^Φ(_)]. In my view, Hornstein's analysis highlights some reasons for wanting a fundamentally conjunctive (as opposed to saturationist) account of human tenses. But in any case, his account lends itself to expresssion in terms of M-junction and Θ-junction.[8]

Let's assume that (27) is a verb phrase of some kind, and describe it simply as follows: [Scarlet stab Mustard]$_V$; where the subscript may reflect both 'stab' and a covert small verb. And lets assume that a tense morpheme can be combined with (27) to form a tense phrase like [-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$; where for purposes of pronunciation, the head of this phrase can be "lowered" onto the verb 'stab' (see Pollack 199x). Then the meaning of the past tense morpheme can be described in much the same way as the meaning of the temporal adverb in (30)

(30) Scarlet stab Mustard yesterday

[8] Grammatical aspect is its own complex topic that I will not try to address here. But see Hacquard (2009, 2010). Her account of is encoded in traditional type-theoretic terms. But as Hacquard's proposed logical forms make clear, her event-based conception of aspect (and its relation to tense) is underlyingly conjunctive, and her leading ideas can be captured terms of Θ-junction.

or perhaps the adverbial phrase in (31).

(31) Scarlet stab Mustard before midnight

That is, $\mu([\text{-ed}_T\ [\text{Scarlet stab Mustard}]_V]_T)$ can be described as the following instruction: **Join**($\mu(\text{-ed}_T)$, **Join**(***fetch@***'T/V', $\mu([\text{Scarlet stab Mustard}]_V)$)). If executing $\mu(\text{-ed}_T)$ yields a monadic concept—say, ∃[BEFORE(_, _)^SPEECH-TIME(_)]—there is no need for a T/V-adapter. In which case, $\mu([\text{-ed}_T\ [\text{Scarlet stab Mustard}]_V]_T)$ has the same effect as the formally simpler instruction **M-join**($\mu(\text{-ed}_T)$, $\mu([\text{Scarlet stab Mustard}]_V)$).

This captures the subtle ambiguity of (32),

(32) Peacock hear Scarlet shoot Green yesterday

which is structurally homophonous as bewteen (32′) and (32″),

(32′) $[\text{Peacock hear } [[\text{Scarlet shoot Green}]_V \text{ yesterday}_A]_V]_V$

(32″) $[[\text{Peacock hear } [\text{Scarlet shoot Green}]_V]_V \text{ yesterday}_A]_V$

either of which can be a constituent of a question like (33); see Higginbotham (198x).

(33) Did Peacock hear Scarlet shoot Green today?

If the shot occurred just before midnight, but Peacock heard it just after midnight, then the answer to (33) might be positive on the (32′)-reading but negative on the (33′)-reading; cp. the discussion in §x.x of chapter three. But adding a tense morpheme to (32), on either reading, would unambiguously add the temporal restriction to HEAR(_) rather than SHOOT(_).

However, if adding a tense morpheme corresponds to just another case of M-junction, then executing $\mu([\text{-ed}_T\ [\text{Scarlet stab Mustard}]_V]_T)$ yields a concept like (34).

(34) ∃[BEFORE(_, _)^SPEECH-TIME(_)]^
∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]

And while $\mu([\text{-ed}_T\ [\text{Scarlet stab Mustard}]_V]_T)$ can be paired with the pronunciation of (34),

(34) Scarlet stabbed Mustard

which can be viewed as tensed clause, this clause differs somehow from the "complete sentence" (3), whose period corresponds to something like an existential closure of the open slot in (34).

(3) Scarlet stabbed Mustard.

This can make it tempting to describe $\mu([\text{Scarlet stab Mustard}]_V)$ and $\mu(\text{-ed}_T)$ as meanings of types <e, **t**> and <<e, **t**>, **t**>. For example, one might specify $\mu(\text{-ed}_T)$ as the function λΦ.⊤≡ ∃E[Φ(E) & PAST(E)]. But this stipulates that a tense morpheme, unlike the adverb in (30),

(30) Scarlet stab Mustard yesterday

does two jobs: add temporal information, like a *conjunctive adjunct*; and close a variable, like a *quantificational argument*. Alternatively, one might say that (3) is the result of adding a covert existential quantifier—of type <<e, **t**>, **t**>—to $[\text{-ed}_T\ [\text{Scarlet stab Mustard}]_V]_T$. But this raises the question of why such a quantifier cannot be pronounced, with (35) having the meaning of (3).

(35) Some event Scarlet stabbed Mustard.

I don't doubt that some such account can be sustained. But there is another way of describing the difference between (3) and (30), according to which even (3) is an instruction for how to build a monadic concept.

2.2 T-Concepts

A monadic concept of events is neither true or false. One can say that (34)

(34) ∃[BEFORE(_, _)^SPEECH-TIME(_)]^
∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]

is satisfied by an assignment $\mathcal{A}$ of value(s) to variables iff $\mathcal{A}$ assigns to the "zeroth" variable—where by stipulation, this is the variable associated with the open slot in (34)—one or more things that are *both* events of Scarlet stabbing Mustard *and* things that occurred prior to whatever time $\mathcal{A}$ assigns to the variable associated with 'now'. Correlatively, one can say that relative to any given time, (34) is *true of* stabbings of Mustard that were done by Scarlet prior to that time. But (34) is not itself true or false, not even relative to a time. However, it doesn't follow that if a concept has a satisfaction condition, it cannot be true or false. On the contrary, Tarski (1933) showed how to describe truth in terms of satisfaction. As discussed in chapter two, the closed sentences of a Tarskian language are not expressions of a special type <**t**>. They are, however, sentences that are satisfied by each assignment or by none. Likewise, (34) can be "closed" in a way that yields another monadic concept, albeit one that applies to everything or nothing.

Consider a pair of operators, ⇑and ⇓, that create monadic concepts from monadic concepts; where for any one or more things, ⇑Φ(_) applies to them iff Φ(_) applies to one or more things, and ⇓Φ(_) applies to them iff Φ(_) applies to nothing.[9] More briefly, and without worrying about number neutrality: for each entity, ⇑Φ(_) applies to it iff Φ(_) applies to something; and ⇓Φ(_) applies to it iff Φ(_) applies to nothing. One can think of ⇑ and ⇓ as *polarizing* operators—cp. §2.2 of chapter one—that convert any monadic concept into a concept of everything or nothing, perhaps akin to EXIST(_) and ~EXIST(_). For example, given any entity, ⇑COW(_) applies to it iff COW(_) applies to something; so ⇑COW(_) applies to you, and likewise to me, iff there is a cow. By contrast, ⇓COW(_) applies to you (and to me) iff nothing is a cow. And for each thing, either ⇑COW(_) or ⇓COW(_) applies to it—since each thing is either such that there is a cow, or such that there is no cow. Correlatively, nothing is such that both ⇑COW(_) and ⇓COW(_) apply to it. Hence, nothing is such that ⇑COW(_)^⇓COW(_) applies to it.

Given a suitable metalanguage, we can say: ∀X{⇑Φ(X) ≡ ∃Y[Φ(Y)]}; and similarly, ∀X{⇓Φ(X) ≡ ~∃Y[Φ(Y)]}. But the idea is not that '⇑' and '⇓' are abbreviations for Tarski-style existential closure and its negation. For example, '⇑BETWEEN(X, Y, Z)' is gibberish, as is '⇓AGENT(E, X)'. The idea is rather that certain human linguistic meanings, perhaps associated with tense and/or negation, invoke "closure" operations that convert a monadic concept (say, of events) into a concept of all or none. So let's say that any concept of the form ⇑Φ(_) or ⇓Φ(_) is a *T-concept*, with 'T' connoting Tarski, truthy, and totality.

Note that for any concept Φ(_) and any entity e, ⇑⇑Φ(_) applies to e iff ⇓⇓Φ(_) does, since each of these concepts applies to e iff ⇑Φ(_) does—i.e., iff Φ(_) applies to something. Likewise, ⇑⇓Φ(_) applies to e iff ⇓⇑Φ(_) does, since each of these concepts applies to e iff ⇓Φ(_) does—i.e., iff Φ(_) applies to nothing. And while ⇑[Φ(_)^Ψ(_)] applies to e iff something falls under the conjunctive concept Φ(_)^Ψ(_), which applies to e iff e falls under both conjuncts, ⇑Φ(_)^⇑Ψ(_)] applies to e iff (e is such that) something falls Φ(_) *and* something falls

[9] Or if you prefer, for any one or more things: [⇑Φ(_)](_) applies to them iff Φ(_) applies to one or more things; and [⇓Φ(_)](_) applies to them iff Φ(_) applies to nothing. But omitting the extra slot and brackets turns out to be at least as perspicuous. For these purposes, let me ignore mass nouns and assume countable values of variables.

under Ψ(_). Thus, ⇑[BROWN(_)^COW(_)] is more restrictive than ⇓BROWN(_)^⇑COW(_), much as the more familiar (36) implies (37), but not conversely.

(36) ∃X[BROWN(X) & COW(X)]

(37) ∃X[BROWN(X)] & ∃X[COW(X)].

Correlatively, ⇓ [BROWN(_)^COW(_)] applies to e iff nothing is both brown and a cow, while ⇓BROWN(_)^⇓COW(_) applies to e iff (e is such that) nothing is brown *and* nothing is a cow. So ⇓BROWN(_)^⇓COW(_) is more restrictive than ↓[BROWN(_)^COW(_)]. And ↓COW(_) is more restrictive than ⇓[BROWN(X)^COW(X)]; again, cp. the earlier discussion of "medieval polarity."

Note that when the concepts conjoined are themselves T-concepts, which apply to all or none, "closing up" has no effect. If **P** and **Q** are T-concepts, and so each is of the form ⇑Φ(_) or ⇓Φ(_), then ⇑[**P**^**Q**] is logically equivalent to **P**^**Q**: ⇑[**P**^**Q**] applies to e iff something/everything falls under both **P** and **Q**; **P**^**Q** applies to e iff e/everything falls under both **P** and **Q**. By contrast, ⇓ [⇓**P**^⇓**Q**] applies to e iff: nothing falls under both ⇓**P** and ⇓**Q**; i.e., nothing is such that both **P** and **Q** are empty; i.e., something falls under **P** *or* something falls under **Q**. So propositional disjunction can be characterized, *a la* de Morgan, given T-concepts.

So we should pause before assuming that human I-languages generate expressions of the Fregean type <**t**>, as opposed to expressions that can be used to build T-concepts, which can bear an intimate relation to existential *thoughts* of type <**t**>. Let's abbreviate (34) as (34a),

(34) ∃[BEFORE(_, _)^SPEECH-TIME(_)]^
∃[EXTERNAL(_, _)^SCARLET(_)]^STAB(_)^∃[INTERNAL(_, _)^MUSTARD(_)]

(34a) PAST-STAB-OFMUSTARD-BYSCARLET(_)

and assume that this is the concept built by executing μ([-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$). If (3)

(3) Scarlet stabbed Mustard.

is the result of combining the homophonic tensed clause with a silent polarity head as in (3′)

(3′) [+$_{Pol}$ [-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$]$_{Pol}$

then (3) can analyzed as the following instruction: **CloseUp**(μ([-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$)); where this instruction is executed by executing the embedded instruction, thereby creating (34a), and applying the operator ⇑ to form the corresponding T-concept, (3a).

(3a) ⇑PAST-STAB-OFMUSTARD-BYSCARLET(_)

This clearly invokes a non-conjunctive (polarizing) operation. So principle (11)

(11) μ([…$_L$, … $_{L*}$]$_L$) = **Join**(μ(…$_L$), **Join**(***fetch***@'L/L*', μ(…$_{L*}$)))

must be limited to tensed clauses (including aspect in the tense system for these purposes). From this perspective, the polarity head marks a "new phase" of composition in which an instruction that conforms to (11) is "topped off" in certain ways, perhaps in ways that correspond to preparing a constructed monadic concept for certain uses; see Lohndal and Pietroski (2011) for related discussion drawing on Chomsky (2005). But even if the meaning of (3) is an instruction to perform a closure operation on a certain concept (obtained by executing a tensed instruction), the net result is still monadic concept. And I don't think that this analysis of (3) is any less simple than positing a covert quantifier of type <<e, **t**>, **t**>. If (5) has the structure shown in (5′),

(5) Scarlet didn't stab Mustard.

(3′) [not$_{Pol}$ [-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$]$_{Pol}$

with "sentential" negation characterized as an overt negative polarity item, then (5) can analyzed

as the following instruction: **CloseDown**(μ([-ed$_T$ [Scarlet stab Mustard]$_V$]$_T$)); where this instruction is executed by executing the embedded instruction, thereby creating (34a), and applying the operator to form the corresponding T-concept, (5a).

(5a) ⇓PAST-STAB-OFMUSTARD-BYSCARLET(_)

A mind that can form monadic concepts like (3a) and (5a) might also be able to form the corresponding *thoughts*, (3b) and (5b),

(3b) ∃E[PAST-STAB-OFMUSTARD-BYSCARLET(E)]

(5b) ~∃E[PAST-STAB-OFMUSTARD-BYSCARLET(E)]

or perhaps subject-predicate analogs of these existential generalizations. Such a mind might treat (3a) and (5a) as cognitively equivalent to (3b) and (5b), at least for many purposes. But while "pre-linguistic" and "post-linguistic" cognition presumably traffics in complete thoughts, at least often, human I-languages may *interface with* such cognition via formally monadic T-concepts. Indeed, one can view (3a) and (5a) as formal variants of (3b) and (5b), which cannot themselves be outputs of (or inputs to) M-junction or Θ-junction.

2.3 Abstraction

Given the operators ⇑ and ⇓, a mind can convert a concept that applies to some but not all things into a T-concept that applies to all or none. But this does not yet explain the converse capacity, much less the capacity to form complex concepts like WHO-SCARLET-STABBED(_). And this is arguably the most interesting respect in which human thought is recursive. Reformulating a point from chapter one: given **P^Q**, the capacity to form **P**^[**P^Q**] is not that impressive; and in important respects, the recursion exhibited by 'who saw a colonel that Scarlet stabbed' is more interesting than that exhibited by 'thinks that Peacock said that Scarlet stabbed Mustard'. Embedding one sentence in another is a good trick. Using a sentence to create a concept that applies to some but not all things is a great trick. Clearly, this requires more than mere conjunction. Though as Tarski (1933) showed, the requisite machinery is not especially complicated: it involves a distinctive operation—in effect, quantification over ways of assigning values to variables—but not saturation. So if as I suspect, appeal to the Tarskian operation is unavoidable in theory of meaning for a human I-language, then we need ask whether such a theory of should *also* appeal to a Frege-Church operation along with a hierarchy of types.

Let's assume that humans have *conceptual indices* that can be used as devices for temporarily tracking salient things (cp. Pylylshyn [2007]). Some of these indices may be singular context-sensitive denoters of type <e>. But suppose that human I-languages allow for *grammatical indices* like those indicated with the numerical subscripts in [she$_1$ stab him$_2$]$_V$ and [them$_3$ stab it$_4$]$_V$. And suppose these indices can be used to access context-sensitive mental predicates like FIRST(_) and SECOND(_). The idea is that in any context, FIRST(_) applies to whatever is tracked with the first index used in that context, and likewise for SECOND(_).

If it helps, we can say that SECOND(_) is satisfied by an assignment $\mathcal{A}$—of values to potential variables—iff the one or more things that $\mathcal{A}$ assigns to the (zeroth) variable associated with the open slot are the one or more things that $\mathcal{A}$ assigns to the second potential variable; likewise for FIRST(_). Then one can view the meanings of [-ed$_T$ [she$_1$ stab him$_2$]$_V$]$_T$ and [-ed$_T$ [them$_3$ stab it$_4$]$_V$]$_T$ as instructions for how to build concepts like (38) and (39).

(38) PAST(_)^∃[EXTERNAL(_,_)^[FIRST(_)^FEMALE(_)^~PLURAL(_)]^
STAB(_)^∃[INTERNAL(_, _)^[SECOND(_)^MALE(_)^~PLURAL(_)]

(39) PAST(_)^∃[EXTERNAL(_,_)^[FIRST(_)^PLURAL(_)]^
STAB(_)^∃[INTERNAL(_, _)^[SECOND(_)^INANIMATE(_)^~PLURAL(_)]

Similarly, one can view μ([me stab him_1]$_V$) as an instruction for how to build a concept like (40).

(40) PAST(_)^∃[EXTERNAL(_,_)^[SPEAKER(_)^~PLURAL(_)]^
STAB(_)^∃[INTERNAL(_, _)^[FIRST(_)^MALE(_)^~PLURAL(_)]

So sentences like (4) can also be accommodated, given the proposed account of clauses like (2).

(4) Peacock saw me stab him.

(2) see Scarlet stab Mustard

Note that if grammatical indices are analyzed as expressions of type <e>, then one needs to account for the contrasting predicative aspects of 'she_1' and 'him_2' without supposing that these pronouns are results of combining an expression of type <e, **t**> with an index. But let's ignore gender for simplicity, and abbreviate (38-40) as (38a-40a), suppressing thematic structure.

(38a) PAST-2-STAB-1(_)

(39a) PAST-3-STAB-4(_)

(40a) PAST-2-SPEAKER-STAB-1(_)

Corresponding to (38a), there are two T-concepts. The concept ⇑PAST-2-STAB-1(_) applies to everything or nothing, relative to assignment $\mathcal{A}$, depending on whether or not: prior to whatever time $\mathcal{A}$ assigns to the variable associated with 'now', there were some stabbings of whatever $\mathcal{A}$ assigns to the second index done by whatever $\mathcal{A}$ assigns to the first index. The concept ⇓PAST-2-STAB-1(_) applies to nothing iff ⇑PAST-2-STAB-1(_) applies to everything. Let's abbreviate these T-concepts, in bold, as follows: **2-STABBED-1**(_); **NOT-2-STABBED-1**(_). Likewise, let's abbreviate the T-concepts corresponding to (39a) and (39b) as follows: **3-STABBED-4**(_); **NOT-3-STABBED-4**(_); **SPEAKER-STABBED-1**(_); **NOT-SPEAKER-STABBED-1**(_).

These are analogs of conditional assignment-relative specifications of truth values like '⊤ if there was a stabbing by $\mathcal{A}$(1) of $\mathcal{A}$(1), and ⊥ otherwise'. But T-concepts apply to everything or to nothing; they do not specify truth values. One can say that this is notational variation. But then appealing to truth values and expressions of type <**t**> is a potentially misleading notational variant of appeal to instructions for how to build T-concepts. In any case, a mind that can form T-concepts like **2-STABBED-1**(_)—perhaps via introduced concepts that can be M-joined and Θ-joined to form monadic concepts that can be "closed" with the operator ⇑—does not need the full power of Church-style lambda abstraction to form concepts like WHO-SCARLET-STABBED(_).

As discussed in chapter two, the invented language CPL generates denoters of every Fregean type. But for any given variable type, lambda abstraction is equivalent to Tarski-style quantification over assignments of values to variables of that type. So for example, relative to any assignment $\mathcal{A}$: the complex denoter λX.⊤ ≡ SPY(X) denotes the smallest function F of type <e, **t**> such that for each assignment $\mathcal{A}^*$ such that $\mathcal{A}^* \approx_X \mathcal{A}$, F maps $\mathcal{A}^*$(X) to ⊤ or ⊥, depending on whether or not $\mathcal{A}^*$ satisfies the open sentence SPY(X); and likewise, the complex denoter 'λX.⊤ ≡ ABOVE(SCARLET, X) & ABOVE(X, X′)' denotes the smallest function F of type <e, **t**> such

that for each assignment $\mathcal{A}^*$ such that $\mathcal{A}^* \approx_x \mathcal{A}$, F maps $\mathcal{A}^*$(X) to ⊤ or ⊥, depending on whether or not $\mathcal{A}^*$ satisfies the open sentence ABOVE(SCARLET, X) & ABOVE(X, X′).

As I said in chapter two, lambda abstraction can be viewed as a way of unleashing the power of quantifying over sequence variants within a language that extends the Tarskian language PL in a Fregean way. But PL already allowed for open sentences with arbitrarily many variables. So given the Fregean hierarchy of types, CPL generates concepts all of those types, starting with the more than two million types at Level Four or lower, the more than 5 x 10^{12} types at Level Five, etc. One might hope that sparer resources would suffice to let a mind form concepts like WHO-SCARLET-STABBED(_) and WHO-STABBED-MUSTARD(_). And indeed, a mind that can form T-concepts like **2-STABBED-1**(_) only needs the basic Tarskian trick—of quantifying over assignment variants—to use a relative clause like (6)

(6) who she stabbed

as an instruction whose execution can yield a concept like who-Scarlet-stabbed(_).

Given any index *i* and T-concept **P**, which can evaluated relative to any assignment **A**, let **TARSKI**{*i*, **P**} be a "semantic" concept whose content can be specified with (41);

(41) $\exists\mathcal{A}^*{:}\mathcal{A}^* \approx_i \mathcal{A}$\{ASSIGNS($\mathcal{A}^*$, _, *i*) & SATISFIES($\mathcal{A}^*$, **P**)\}

where ASSIGNS($\mathcal{A}^*$, _, *i*) applies to whatever $\mathcal{A}^*$ assigns to the *i*th index, and '$\mathcal{A}^* \approx_i \mathcal{A}$' means that $\mathcal{A}^*$ differs from $\mathcal{A}$ at most with regard to what it assigns to the *i*th index. To be sure, any natural concept of satisfaction is likely to differ from Tarski's; and **TARSKI**{*i*, **P**} is posited as a natural concept, presumably less intellectualized than (41). But my suggestion is that we can use a T-concept like **2-STABBED-1**(_) to form a concept that applies to one or more things (relative to an assignment of values to indices) iff making them the values of the first index (and holding everything else constant) satisfies the T-concept. One can think of this as a restricted number-neutral version of lambda abstraction. But the idea is not that we apply a general capacity for lambda abstraction to (6). The idea is human I-languages support Tarski-style abstraction on T-concepts that contain indices corresponding to concepts like FIRST(_) and SECOND(_).

Relative to any assignment $\mathcal{A}$, and ignoring the tense for simplicity, the concept **TARSKI**{1, **2-STABBED-1**(_)} applies to one or more things iff they were stabbed by whatever $\mathcal{A}$ assigns to '2'. We can specify this application condition more formally with (42);

(42) $\exists\mathcal{A}^*{:}\mathcal{A}^* \approx_1 \mathcal{A}$\{ASSIGNS($\mathcal{A}^*$, _, 1) & SATISFIES($\mathcal{A}^*$, **2-STABBED-1**(_)\}

if $\mathcal{A}^* \approx_1 \mathcal{A}$, $\mathcal{A}^*$ assigns whatever $\mathcal{A}$ assigns to '2'. And we can abbreviate (42) with (42a).

(42a) λ1.**2-STABBED-1**

But we shouldn't let this *abbreviation* make us think that **TARSKI**{1, **2-STABBED-1**(_)} is a concept of a function that maps entities to truth values. Similarly, **TARSKI**{2, **2-STABBED-1**(_)} applies to one or more things iff they stabbed whatever **A** assigns to '1'. We can specify this application condition more formally with (43), which we can abbreviate with (43a).

(43) $\exists\mathcal{A}^*{:}\mathcal{A}^* \approx_2 \mathcal{A}$\{ASSIGNS($\mathcal{A}^*$, _, 2) & SATISFIES($\mathcal{A}^*$, **2-STABBED-1**(_)\}

(43a) λ2.**2-STABBED-1**

But *abbreviation* doesn't show that **TARSKI**{2, **2-STABBED-1**(_)} maps entities to truth values, much less that (44) denotes a truth value relative to each assignment of values to indices in (44).

(44) She stabbed him.

If **2-STABBED-1**(_) doesn't denote a truth value, **TARSKI**{2, **2-STABBED-1**(_)} does not denote a function of type <e, **t**>. And one can't establish that **2-STABBED-1**(_) denotes a truth value by insisting that **TARSKI**{2, **2-STABBED-1**(_)} denotes a function of type <e, **t**>. But I agree that using **2-STABBED-1**(_) to build a "relative clause concept," with an application condition like that of (42) or (43), is an operation considerably more sophisticated than conjunction.

Indeed, this kind of concept construction violates certain compositionality constraints that M-junction and Θ-junction respect. Relative to a particular assignment $\mathcal{A}$, the **2-STABBED-1**(_) may apply to nothing, while **TARSKI**{1, **2-STABBED-1**(_)} applies to many things; see Salmon (2006). Suppose that whatever $\mathcal{A}$ assigns to '2', it/they stabbed many things, but *not* whatever $\mathcal{A}$ assigns to '1'. Then relative to $\mathcal{A}$: **2-STABBED-1**(_) is false of each thing, and in that sense false; yet **TARSKI**{1, **2-STABBED-1**(_)} is true of many things, and so ⇑**TARSKI**{1, **2-STABBED-1**(_)} is true of each thing. And if whatever $\mathcal{A}$ assigns to '2' stabbed nothing, then **2-STABBED-1**(_) and ⇑**TARSKI**{1, **2-STABBED-1**(_)} are false relative to $\mathcal{A}$. Like it or not, this "non-truth-functional" composition is available to a mind that is equipped to perform the Tarski trick. And my claim is *not* that this kind of abstraction can be reduced to anything else. On the contrary, I see no way to avoid positing a capacity for construction in minds that can understand relative clauses like (6).

(6) who she stabbed

But in offering plausible theories of meaning for human I-languages, I think we also need to posit—one way or another—the more mundane operations of M-junction and Θ-junction. So in my view, the question is whether we *also* need Fregean typology and an operation of saturation.

I haven't yet specified the meaning of (6). But let's assume, standardly, that the 'wh'-expression is a displaced analog of 'him$_2$' in the tensed phrase [-ed$_T$ [she$_1$ stab him$_2$]$_V$]$_T$; where the displaced position is associated with a covert complementizer that takes a sentential complement. If sentencehood is associated with a polarity head, as suggested above, then (6) has a grammatical structure along the lines of (6′);

(6′) [who$_1$ [_$_C$ [+$_{Pol}$ [-ed$_T$ [she$_2$ stab$_V$ ~~who~~$_1$]$_V$]$_T$]$_{Pol}$]$_C$]$_C$

For simplicity, let's assume that the covert complementizer is semantically null—that it is there simple to support the extraction of 'who'—and that μ([_$_C$ [+$_{Pol}$ [-ed$_T$ [she$_2$ stab$_V$ ~~who~~$_1$]$_V$]$_T$]$_{Pol}$]$_C$]) just is the meaning of the polarity phrase. For any index *i*, let an instruction of the form **ABSTRACT**(*i*, μ([…]$_{Pol}$])) be executed by executing the instruction μ([…]$_{Pol}$])—thereby forming a T-concept like **2-STABBED-1**(_)—and forming the corresponding concept of the form **TARSKI**{*i*, **P**}. Then the meaning of (6) can be specified as the following instruction: **ABSTRACT**(1, μ([+$_{Pol}$ [-ed$_T$ [she$_2$ stab$_V$ ~~who~~$_1$]$_V$]$_T$]$_{Pol}$])).

Executing this meaning can yield **TARSKI**{1, **2-STABBED-1**(_)}; cp. (42/42a).

(42) $\exists\mathcal{A}^*{:}\mathcal{A}^*\approx_1\mathcal{A}$\{ASSIGNS($\mathcal{A}^*$, _, 1) & SATISFIES($\mathcal{A}^*$, **2-STABBED-1**(_)\}

(42a) λ1.**2-STABBED-1**

Likewise, executing the meaning of (45) can yield **TARSKI**{2, **2-STABBED-1**(_)}; cp. (43/43a).

(45) who stabbed him

(43) $\exists\mathcal{A}^*{:}\mathcal{A}^*\approx_2\mathcal{A}$\{ASSIGNS($\mathcal{A}^*$, _, 2) & SATISFIES($\mathcal{A}^*$, **2-STABBED-1**(_)\}

(43a) λ2.**2-STABBED-1**

In general, μ([wh$_i$ …]$_C$) can be specified as **ABSTRACT**(*i*, μ(…)). One can declare this to be notational variant of lambda abstraction in a system that posits truth values and the possibility of converting expressions of type <**t**> into expressions of type <e, **t**>. But then such a system is a notational variant one that posits a capacity for using polarized phrases to build T-concepts and relative clauses (along with indices) to build corresponding concepts of the form **TARSKI**{*i*, **P**}.[10]

3. Quantification

The meanings of quantificational constructions like (7) and (8) can now be accommodated.

(7) Every guest stabbed him.

(8) She stabbed every guest.

I assume that quantifiers like 'every spy' displace as in the structures shown below.

[[every guest]$_{Q2}$ [+$_{Pol}$ [[~~every guest~~]$_2$ stabbed$_V$ him$_1$]$_T$]$_{Pol}$]$_Q$

[[every guest]$_{Q1}$ [+$_{Pol}$ [she$_2$ stabbed$_V$ [~~every guest~~]$_1$]$_T$]$_{Pol}$]$_Q$

For present purposes, the details regarding labeling do not matter. The idea is simply that in (7), the indexed quantifier [every guest]$_{Q2}$ combines with a polarity phrase that can be used to build the T-concept **2-STABBED-1**(_); while in (8), the indexed quantifier [every guest]$_{Q1}$ combines with a polarity phrase that can be used to build the same T-concept. But the idea is *not* that the indexed quantifiers combine with expressions that have the meanings of relative clauses like (45)

(45) who stabbed him

On the contrary, I want to explain why (10) cannot be understood as synonymous with (7).

(10) every guest who stabbed him

---

[10] As Heim and Kratzer's (1998) system highlights, even if one appeals to saturation as a composition operation, one still needs to posit Tarskian abstraction—often encoded as lambda abstraction—as a distinct operation, even given a Fregean typology. Suppose we treat indices and traces as constituents, as in [2 [she$_1$ chased$_V$ t$_2$]$_S$], with the embedded sentence as an expression of type <**t**> and the larger expression as of type <e, **t**>. From an I-language perspective, one can say (modulo tense and gender) that relative to any assignment $\mathcal{A}$: the concept formed by executing [she$_1$ chased t$_2$]$_S$ denotes truth iff whatever $\mathcal{A}$ assigns to 1 chased whatever $\mathcal{A}$ assigns to 2; and correlatively, the concept formed by executing [2 [she$_1$ chased$_V$ t$_2$]$_S$] applies to entity e iff whatever $\mathcal{A}$ assigns to 1 chased e. But the idea *isn't and can't be* that the index denotes a function-in-extension of type <**t**, <e, **t**>>, which maps the truth value of [she$_1$ chased t$_2$] onto a function of type <e, **t**>. Rather, '2' has to indicate a hypothesized (syncategorematic) instruction to convert a *representation* of one sort into a *representation* of another sort. (Kobele [200x] shows how to pretend otherwise, for certain purposes, by positing a domain that includes assignments as entities in the basic domain. But in my view, this highlights the point.)

Heim and Kratzer's third composition rule, in addition to rules for saturation and M-junction, makes this vivid. Their idea is that the higher copy of the lower index triggers quantification over assignment variants:

$$\|[2\text{^}[\text{she}_1 \text{ chased}_V \text{ t}_2]_S]\|^{\mathcal{A}} =$$

$$\lambda x.\ \top \text{ iff } \exists \mathcal{A}^*{:}\mathcal{A}^* \approx_2 \mathcal{A}[x = \mathcal{A}^*(2)\ \&\ \|[\text{she}_1 \text{ chased}_V \text{ t}_2]_S]\|^{\mathcal{A}^*} = \top]$$

This has the desired result, taking the lambda-expression to be a theorist's representation of the hypothesized *concept* obtained in two stages: execute the sentential instruction, obtaining a concept that is doubly sequence-sensitive, and modify the resulting concept as directed by the *index* '2'. One can remain agnostic about the detailed forms of the concepts constructed. And from an E-language perspective, one can take the lambda-expression to be (only) a theorist's representation of the hypothesized *satisfaction condition*; cp. Kobele (200x). But from an I-language perspective, the goal is to say how competent speakers represent the alleged satisfaction condition. And while theorists can abbreviate—as in $\|[2\text{^}[\text{she}_1 \text{ chased}_V \text{ t}_2]_S]\|^{\mathcal{A}} = \lambda x.$**CHASE(1, X)**—we should remember that the corresponding *psychological* hypothesis presupposes some version of the Tarski trick.

My suggestion is that in (7), [every guest]$_{Q2}$ combines with a polarity phrase, not a relative clause. And while polarity phrases are used to build monadic concepts, of a special sort, the formal difference matters in ways that are mirrored by the meanings of quantifiers like 'every'.

3.1 Quantifiers as Plural Predicates

The leading idea is simple: a quantificational determiner like 'every' ('some', 'no', 'most', etc.) accesses a number neutral concept of ordered pairs; where by stipulation, the ordered pair ⟨α, β⟩ has β as its internal participant and α as its external participants. More specifically, let's say that EVERY(_) applies to some ordered pairs iff: every one of their internal participants is one of their external participants; or put another way, (all of) their internals are among their externals. Likewise, MOST/SOME/NO(_) applies to some ordered pairs iff most/some/none of their internals are among their externals.

This is *analogous* to saying that quantificational expressions are true of certain ordered pairs of *sets*. But EVERY(_) applies, plurally, to the following six ordered pairs: ⟨1, 1⟩; ⟨2, 1⟩; ⟨3, 3⟩; ⟨4, 3⟩; ⟨5, 5⟩; ⟨6, 5⟩. Each of the three internal participants—1, 3, and 5—is one of the external participants. Likewise, EVERY(_) applies, plurally, to the boundless many ordered pairs ⟨x, y⟩ such that x is a positive integer and y is a positive odd integer. But EVERY(_) does not apply to ⟨{x: x is a positive integer}, {x: x is a positive odd integer}⟩. Nor does EVERY(_) apply to this ordered pair of sets and the first six ordered pair of numbers, since one of internal participants—{x: x is a positive odd integer}— is not one of the external participants. But EVERY(_) does apply, plurally, to the following three ordered pairs: ⟨1, 5⟩; ⟨3, 3⟩; ⟨5, 1⟩.

Let's say that for any concept Φ(_), the concept MAX:Φ(_) applies to some things iff they are (all and only) *the* things to which Φ(_) applies. Put another way, MAX:Φ(_) applies to some things, the Xs, iff any one more things, the Ys, that Φ(_) applies to are among the Xs. Then the three-conjunct concept (46) applies to some ordered pairs iff:

(46) EVERY(_)^∃[INTERNAL(_, _)^MAX:GUEST(_)]^
∃[EXTERNAL(_, _)^MAX:ARRIVED(_)]

each of their internal participants is one of their external participants; their internal participants are the guests; and their internal participants are the things that arrived. More briefly, (46) applies to some ordered pairs iff each of the guests is one of the things that arrived. Or still more briefly, (46) applies to some ordered pairs iff every guest arrived.

So using the operator ⇑ to "close up" (46) yields a T-concept that applies to everything or nothing, depending on whether or not every guest arrived. This should already make it clear that we have the resources needed to specify the meanings of (7) and (8),

(7) Every guest stabbed him.
(8) She stabbed every guest.

at least if they have the structures shown in (7′) and (8′).

(7′) [$+_{Pol}$ [[every$_Q$ guest$_N$]$_{Q2}$ [$+_{Pol}$ [[~~every guest~~]$_2$ stabbed$_V$ him$_1$]$_T$]$_{Pol}$]$_Q$]$_{Pol}$
(8′) [$+_{Pol}$ [[every$_Q$ guest$_N$]$_{Q1}$ [$+_{Pol}$ [she$_2$ stabbed$_V$ [~~every guest~~]$_1$]$_T$]$_{Pol}$]$_Q$]$_{Pol}$

What remains are some compositional details. In spelling these out, let's assume that the concept INTERNAL(_, _) is the Q/N-adapter, while the concept EXTERNAL(_, _) is the Q/Pol-adapter. This corresponds to the familiar idea that a lexical quantifier combines with a noun to form a unit that

displaces, in order to combine with the sentence from which that unit displaces.

3.2 Building the Plural Predicates

If principle (11) applied to all labeled expressions,

(11) $\mu([\ldots_{L}, \ldots_{L*}]_{L})$ = **Join**($\mu(\ldots_{L})$, **Join**(***fetch*@**'L/L*', $\mu(\ldots_{L*})$))

then $\mu([\text{every}_{Q}, \text{guest}_{N}]_{Q})$ would be the instruction (47),

(47) **Join**($\mu(\text{every}_{Q})$, **Join**(***fetch*@**'Q/N', $\mu(\text{guest}_{N})$))

whose execution would yield a concept like EVERY(_)^∃[INTERNAL(_, _)^GUEST(_)]. That is close to what we want. But we 'every guest' does not have the meaning of 'every one of *some* guests', it has the meaning of 'every one of *the* guests'. One possible response is to posit a covert maximizing operator in the 'every guest' itself. But if we already need to say that (11) does not apply "above tense," then another option is to posit the slight variation on (11) shown in (48);

(48) $\mu([\ldots_{Q}, \ldots_{N}]_{Q})$ = **Join**($\mu(\ldots_{Q})$, **Join**(***fetch*@**'Q/N', **Maximize**:$\mu(\ldots_{N})$))

where for any instruction **I**, the macro instruction **Maximize:I** is executed by executing **I**—thereby forming a monadic concept Φ(_)—and forming the corresponding concept MAX:Φ(_). Then as desired, executing $\mu([\text{every}_{Q}, \text{guest}_{N}]_{Q})$ would yield a concept like (49).

(49) EVERY(_)^∃[INTERNAL(_, _)^MAX:GUEST(_)]

Let's continue to assume that executing $\mu([+_{Pol}\ [[\cancel{\text{every guest}}]_{2}\ \text{stabbed}_{V}\ \text{him}_{1}]_{T}]_{Pol}])$ yields the concept **2-STABBED-1**(_); where this T-concept applies to everything or nothing, relative to any assignment $\mathcal{A}$, iff $\mathcal{A}(2)$ stabbed $\mathcal{A}(1)$. We have seen how **2-STABBED-1**(_) can be a constituent of the concept **TARSKI**{2, **2-STABBED-1**(_)}—*abbreviatable* as λ2.**2-STABBED-1**—which applies, relative to any assignment $\mathcal{A}$, to one or more things iff they stabbed $\mathcal{A}(1)$. So relative to $\mathcal{A}$, concept (50) applies to things whose externals are the things that stabbed $\mathcal{A}(1)$.

(50) ∃[EXTERNAL(_, _)^MAX:**TARSKI**{2, **2-STABBED-1**(_)}]

And concept (50) can be built by executing instruction (51),

(51) **Join**(***fetch*@**'Q/Pol',
Maximize:ABSTRACT(2, $\mu([+_{Pol}\ [[\cancel{\text{every guest}}]_{2}\ \text{stabbed}_{V}\ \text{him}_{1}]_{T}]_{Pol}])$))

So rule (52) will do the job.

(52) $\mu([\ldots_{Qi}\ \ldots_{Pol}]_{Q})$ =
Join($\mu(\ldots_{Qi})$, **Join**(***fetch*@**'Q/Pol', **Maximize:ABSTRACT**(i, $\mu(\ldots_{Pol})$))

Then the meaning of the full Q-phrase in (7′) can be specified as the following instruction:

(7′) $[+_{Pol}\ [[\text{every}_{Q}\ \text{guest}_{N}]_{Q2}\ [+_{Pol}\ [[\cancel{\text{every guest}}]_{2}\ \text{stabbed}_{V}\ \text{him}_{1}]_{T}]_{Pol}]_{Q}]_{Pol}$

join the meaning of $[\text{every}_{Q}\ \text{guest}_{N}]_{Q2}$ with the meaning specified in (51). Executing this instruction can yield the M-junction of (49) with (50); and the ⇑-closure of this M-junction applies to every or nothing, relative to $\mathcal{A}$, depending on whether or not every guest stabbed $\mathcal{A}(1)$. Likewise, the meaning of the full Q-phrase in (8′) can be specified as the following instruction:

(8′) $[+_{Pol}\ [[\text{every}_{Q}\ \text{guest}_{N}]_{Q1}\ [+_{Pol}\ [\text{she}_{2}\ \text{stabbed}_{V}\ [\cancel{\text{every guest}}]_{1}]_{T}]_{Pol}]_{Q}]_{Pol}$

join the meaning of $[\text{every}_{Q}\ \text{guest}_{N}]_{Q1}$ with the meaning specified in (53).

(53) **Join**(***fetch*@**'Q/Pol',
Maximize:ABSTRACT(1, $\mu([+_{Pol}\ [\text{she}_{2}\ \text{stabbed}_{V}\ [\cancel{\text{every guest}}]_{1}]_{T}]_{Pol}])$))

Executing this instruction can yield the M-junction of (49) with (54);

(54) ∃[EXTERNAL(_, _)^MAX:**TARSKI**{1, **2-STABBED-1**(_)}]

and the ⇑-closure of this M-junction applies to every or nothing, relative to $\mathcal{A}$, depending on whether or not $\mathcal{A}$(2) stabbed every guest.

If (9) has the structure shown in (9′),

(9) Most guests stabbed every spy.

(9′) $[+_{Pol}\ [[\text{most}_Q\ \text{guests}_N]_{Q2}$

$[+_{Pol}\ [[\text{every}_Q\ \text{spy}_N]_{Q1}$

$[+_{Pol}\ [\cancel{\text{most guests}}]\ [_2\ \text{stabbed}_V\ [\cancel{\text{every spy}}]_1]_T]_{Pol}]_Q]_{Pol\ Q}]_{Pol}$

then (9) presents no special difficulty. Executing the meaning of (9) yields a T-concept that applies to everything or nothing, depending on whether are not there are some ordered pairs that meet *both* the $[\text{most}_Q\ \text{guests}_N]_{Q2}$ condition—i.e., the internal participants are the guests, and most of them are also external participants—*and* the following condition: their external participants are the things that (distributively) stabbed every spy; i.e, their external participants are the things such that for every spy, they (each) stabbed that spy. Spelling this condition out, in terms relativization to assignments—and quantification over assignment variants, so that it doesn't matter (modulo tense) which assignment (9) is relativized to—makes the condition look complicated. But the meaning isn't complicated. Or at least it is not more complicated than the meaning posited by more familiar treatments of (9) that appeal to saturation, Fregean typology, and rules for converting assigment-relatived expressions of type <**t**> into assigment-relatived expressions of type <e, **t**>; cp. note 10.

From this perspective, the compositional difference between $[\text{every}_Q\ \text{guest}_N]_{Q2}$ and a verb phrase like $[\text{see}_V\ \text{guests}_N]_V$ is small. The quantificational phrase adds a maximality condition to the internal argument. Of course, one would like to reduce (48)

(48) $\mu([\ldots_Q, \ldots_N]_Q) = \textbf{Join}(\mu(\ldots_Q), \textbf{Join}(\textbf{\textit{fetch}@}\text{'Q/N'}, \textbf{Maximize}{:}\mu(\ldots_N)))$

to a generalization of (11) that does apply "above tense" and a small twist regarding quantifiers,

(11) $\mu([\ldots_L, \ldots_{L*}]_L) = \textbf{Join}(\mu(\ldots_L), \textbf{Join}(\textbf{\textit{fetch}@}\text{'L/L*'}, \mu(\ldots_{L*})))$

(48) is not too ugly even as it stands. The proposal is not to invoke an entirely new principle of composition for quantifiers. On the contrary, the idea is to retain the idea that $[\text{every}_Q\ \text{guest}_N]_{Q2}$ and $[\text{see}_V\ \text{guests}_N]_V$ are *almost* identical compositionally. To be sure, specifying the meaning of $[\text{every}_Q\ \text{guest}_N]_{Q2}$ as in (55) already builds the relevant maximization in via lambda abstraction.

(55) $\lambda\Phi.\lambda\Psi.\top \equiv \{x{:}\Phi(x) = \top\} \subseteq \{x{:}\Psi(x) = \top\}(\lambda x.\top \equiv \text{GUEST}(x))$

Function (55), identical to $\lambda\Psi.\top \equiv \{x{:}\text{GUEST}(x)\} \subseteq \{x{:}\Psi(x) = \top\}$, maps a function F of type <e, **t**> to ⊤ iff the extension of F includes every one of *the guests*. But precisely because lambda abstraction already includes the relevant kind of maximization, specifying the meaning of $[\text{see}_V\ \text{guests}_N]_V$ in terms $\lambda x.\top \equiv \text{GUEST}(x)$ requires something extra—e.g., positing a covert existential quantifier, to get the meaning of 'see *some* guests' rather than 'see (all) *the* guests'.

This doesn't show that my proposal, which locates the extra twist in the quantificational phrase, is correct. But there is no a priori reason for thinking that it is better to locate the extra twist in the "bare plural" verb phrase.[11] And I see no reason for insisting that the indefinite article in $[\text{see}_V\ [\text{a guest}_N]_N]_V$ is an existential quantifier; see Higginbotham (1987). The difference

[11] See Izumi (201x) for relevant comparison of English and Japanese. Cite Dayal, Chierchia (1998).

between [every$_Q$ guest$_N$]$_{Q2}$ and [see$_V$ guests$_N$]$_V$ has to be encoded somehow. One option is to encode it in a way that builds maximalization in via lambda abstraction, and de-maximalizes for certain cases. But another option is to adopt a principle like (11), which assigns weaker (un-maximalized) interpretations by default, and then add maximalization as needed—e.g., for strong quantifiers like 'every', which end up raising above tense.

Principle (52) is a little more *ad hoc.*

(52) $\mu([\ldots_{Qi}\ \ldots_{Pol}]_Q) =$

Join($\mu(\ldots_{Qi})$, **Join**(***fetch@***'Q/Pol', **Maximize:ABSTRACT**(i, $\mu(\ldots_{Pol})$))

Compared with (48), it introduces the further operation of abstraction. The idea is that when [every$_Q$ guest$_N$]$_{Q2}$ combines a polarized phrase—which becomes the external argument of the raised quantifier every$_Q$—the *index* is also relevant to the meaning, because the index of the quantifier tracks which aspect of polarized phrase is relevant to the abstraction. And because the index matters in this way, (52) is as syncategorematic as any other plausible account of quantification direct objects; see Jacobson (199x, 200x) and note 10 above.

So while (52) looks a little uglier than (48), which is not *quite* as simple as (11), I think (48) and (52) have the virtue of making the relevant assumptions about $[[\ldots_Q\ \ldots_N]_{Qi}\ \ldots_{NPol}]_{Qi}$ explicit: the arguments of a quantifier are subject to a kind of maximization; the external argument is also subject to a kind of abstraction, dependent on the index. Moreover, the apparent complexity of (52) allows for a certain kind of simplification.

3.2 Restriction and Conservativity

Let's return to (7).

(7) Every guest stabbed him.

(7′) $[+_{Pol}$ [[every$_Q$ guest$_N$]$_{Q2}$ $[+_{Pol}$ [[~~every guest~~]$_2$ stabbed$_V$ him$_1$]$_T$]$_{Pol}$]$_Q$]$_{Pol}$

In terms of this example, we don't need to specify the relevant external participants as *all the things* that stabbed $\mathcal{A}(1)$. For purposes of capturing the thought built/expressed via (7), it is enough (and perhaps better) to specify the external participants to as the internal participants that stabbed $\mathcal{A}(1)$—*i.e.*, *all the guests* that stabbed $\mathcal{A}(1)$. If poor Colonel Mustard was also stabbed by some non-guests, that is irrelevant. This kind of point is often stressed when discussing the *conservativity* of determiners; see Barwise and Cooper (1981), Higginbotham and May (1981).

Sentence (7) seems synonymous with (56).

(56) Every guest is a guest who stabbed him.

This is interesting, especially given that (7) is not synonymous with (10),

(10) every guest who stabbed him

which cannot be understood as a complete sentence. Specifying the meaning of [every$_Q$ guest$_N$]$_{Q2}$ as in (55), simplified as (55a),

(55) $\lambda\Phi.\lambda\Psi.\top \equiv \{x{:}\Phi(x) = \top\} \subseteq \{x{:}\Psi(x) = \top\}(\lambda x.\top \equiv \text{GUEST}(x))$

(55a) $\lambda\Psi.\top \equiv \{x{:}\text{GUEST}(x)\} \subseteq \{x{:}\Psi(x) = \top\}$

captures the equivalence of (7) and (56): the set of guests is a (perhaps improper) subset of the set of things that stabbed Mustard iff the set of guests is a (perhaps improper) subset of the set of guests that stabbed Mustard. But one wants to know why (10) fails to have an equivalent reading—in addition to its actual reading in which the relative clause restricts 'guest'—and why the "inverted" function specified in (57) is a meaning of any known quantificational determiner.

(57) $\lambda\Phi.\lambda\Psi.\top \equiv \{x{:}\Phi(x) = \top\} \supseteq \{x{:}\Psi(x) = \top\}$

One can say that (10) has no type <**t**> meaning because for *grammatical* reasons, a quantifier like [every$_Q$ guest$_N$]$_{Q2}$ requires an external argument that it is "smaller" than a full relative clause, even though this smaller argument is treated as if it is a relative clause for *semantic* reasons. In my view, there is something right about this idea. But I think it is better expressed by saying that a quantifier like [every$_Q$ guest$_N$]$_{Q2}$ takes a polarized phrase as its external argument, and that the relevant abstraction—on a T-concept like **2-STABBED-1**(_)—reflects a syncategorematic mode of concept composition described with (52).

(52) $\mu([\ldots_{Qi}\ \ \ldots_{Pol}]_Q) =$

Join($\mu(\ldots_{Qi})$, **Join**(***fetch@***'Q/Pol', **Maximize:ABSTRACT**(i, $\mu(\ldots_{Pol})$))

Characterizing the meanings of (7), (10), and (56) *truth-theoretically* seems like a way of missing what is important about these examples, especially if one describes them all in terms of a a fundamentally "saturationist" conception of composition. But whatever one says about the difference between (7) and (10), one wants it to fit with what one says about why there is no analog of every$_Q$ that indicates function (57).

Put another way, if (58) is a possible meaning for a lexical item of a human I-language,

(58) $\lambda\Phi.\lambda\Psi.\top \equiv \{x{:}\Phi(x) = \top\} \subseteq \{x{:}\Psi(x) = \top\}$

one wants to know why (57) is not attested. We can imagine a quantificational determiner gronk$_Q$ that is like every$_Q$, except that (59) has the meaning of (60).

(59) Gronk cow is brown.

(60) Every brown thing is a cow.

The absence of gronk$_Q$ is especially striking given (61),

(61) Only cows are brown.

which means roughly that all brown things are cows. It turns out to be independently plausible that 'only' is a focus operator, and not a quantifier that indicates function (57).[12] But it seems unlikely that children allow for the possibility of such a quantifier, yet always end up concluding that there is no such quantifier.

Likewise, if (58) is a possible meaning for a lexical quantifier, one wants to know why (62) and (63) are not attested, especially given quantifiers like most$_Q$; see note x of chapter one.

(62) $\lambda\Phi.\lambda\Psi.\top \equiv \text{IDENTICAL}[\{x{:}\Phi(x) = \top\} = \{x{:}\Psi(x) = \top\}]$

(63) $\lambda\Phi.\lambda\Psi.\top \equiv \text{EQUINUMEROUS}[\{x{:}\Phi(x) = \top\}, \{x{:}\Psi(x) = \top\}]$

Why don't we have lexical items ident$_Q$ and equi$_Q$ such that (64) and (65)

(64) Ident cows are brown.

(65) Equi cows are brown.

have the meanings of (64a) and (65a), respectively?

(64a) The cows are the brown things.

(65a) The cows are equinumerous with the brown things.

The general constraint is often described in terms of Barwise and Cooper's (1981) notion of conservativity, defined for set-theoretic relations like inclusion: **R**(_, _) is a conservative relation iff: $\forall s \forall s'[\mathbf{R}(s', s) \equiv \Re(s' \cap s, s)$; cp. Higginbotham and May's (1981) notion of intersectivity. One can go on to define the corresponding functions of type <e**t**, <e**t**, **t**>> as

[12] See, e.g., Herburger (2000). Note that 'only' can be inserted anywhere in 'He said that she likes him'.

conservative, and say that no determiners express nonconservative relations. But however the constraint on *sentence meanings* is initially formulated, one need not adopt a lexicalist conception of it. That is, one need not posit a filter that somehow excludes otherwise available lexical meanings like (62) and (63). I have suggested that executing the meaning of a quantifier like every$_Q$ accesses a concept like EVERY(_), which applies to ordered pairs such that (all of) their internal participants are among their external participants. If meanings are described in these terms, one wants to know why there is no analagous though "inverted" quantifier gronk$_Q$ that accesses a concept of ordered pairs whose external participants are among their internal participants. But there may be an answer.

It is often observed that expressions like [every$_Q$ guest$_N$]$_Q$ are understood as *restricted* quantifiers. So one might stipulate that *all* expressions of the form …$_Q$ access concepts of ordered pairs whose external participants are among (or selected from) their internal participants. Then one could say that meaning every$_Q$ accesses the concept like IDENT(_), which applies to ordered pairs such that their external participants *are* (i.e., are all of) their internal participants. Given a global restriction to concepts of ordered pairs whose externals are selected from their internals, adding the specific restriction that the selection be total—or put another way, taking the maximal permitted selection—is equivalent to imposing the restriction that the internals be among the externals. If the guests are among the things that stabbed Mustard, then the guests *are* the guests who stabbed Mustard. That is one way of getting at the equivalence of (7) and (56). But this way of describing the equivalence employs rather than eschews a notion of identity.

(7) Every guest stabbed him.

(56) Every guest is a guest who stabbed him.

In one sense, it is a stipulation to say that lexical quantifiers access concepts of ordered pairs whose external participants selected from their internal participants, as opposed to concepts of ordered pairs whose internal participants selected from their external participants. But if the internal argument of every$_Q$ is a noun corresponding to a monadic concept of the usual sort—one that can, at least in principle, apply to some but not all things—while the external argument of every$_Q$ is a polarized phrase that corresponds to a T-concept, that may create pressure in favor of using the noun as the restrictor on external participants, as opposed to the restricted specifier of external participants; see Pietroski (2005a, 2006) for related discussion in a framework that treats sentences as expressions of type <**t**>.

Moreover, the index on a phrase like [every$_Q$ guest$_N$]$_{Q2}$—corresponding to the (displaced) aspect of a polarized phrase that is relevant to the abstraction associated with the external argument of the quantifier—suggests an easy way of encoding the restriction that external participants be among internal participants. We can restrict the permissible assignment variants. Let's say that for any assignments $\mathcal{A}$ and $\mathcal{A}$*, and any index *i*, $\mathcal{A}^* \subseteq_i \mathcal{A}$ iff: $\mathcal{A}$* differs from $\mathcal{A}$ at most in that $\mathcal{A}$* does not assign to *i* everything that $\mathcal{A}$ assigns to *i*; whatever $\mathcal{A}$ assigns to *i*, $\mathcal{A}$* assigns *one or more but perhaps not all* of those things to *i*. Given an assignment $\mathcal{A}$ that assigns (all and only) the guests to the first index, every *restricted* variant of $\mathcal{A}$ assigns one or more of those guests to the first index. This puts number-neutrality to work in a perhaps unexpected way.

In section two, I posited the concept **TARSKI**{*i*, **P**} as a natural "semantic" concept whose content can be specified formalistically with (41).

(41) $\exists\mathcal{A}^*{:}\mathcal{A}^*\approx_i \mathcal{A}\{\text{ASSIGNS}(\mathcal{A}^*, _, i)\ \&\ \text{SATISFIES}(\mathcal{A}^*, \mathbf{P})\}$ where $\text{ASSIGNS}(\mathcal{A}^*, _, i)$

But perhaps the natural concept is **RESTRICTEDTARSKI**$\{i, \mathbf{P}\}$. And perhaps (57)

(57) $\exists\mathcal{A}^*{:}\mathcal{A}^*\subseteq_i \mathcal{A}\{\text{ASSIGNS}(\mathcal{A}^*, _, i)\ \&\ \text{SATISFIES}(\mathcal{A}^*, \mathbf{P})\}$ where $\text{ASSIGNS}(\mathcal{A}^*, _, i)$

is a better description of this concept's content. In any case, the instruction **ABSTRACT**(i, μ) can characterized in terms of **RESTRICTEDTARSKI**$\{i, \mathbf{P}\}$ instead of **TARSKI**$\{i, \mathbf{P}\}$. And given this characterization, the proposed meaning specifications have the consequence that phrases like $[\text{every}_Q\ \text{guest}_N]_{Q2}$ are restricted quantifiers, and that quantificational expressions exhibit the "conservativity pattern" illustrated by the equivalence of (7) and (56).

(7) Every guest stabbed him.

(56) Every guest is a guest who stabbed him.

This still accommodates the meanings of (7-9) without generating an unwanted meaning for (10).

(8) She stabbed every guest.

(9) Most guests stabbed every spy.

(10) every guest who stabbed him

So by *not* thinking of quantification in terms of Fregean typology, we can preserve the insights of standard accounts while perhaps doing a little *better* in terms of explaining the source of *constraints* on how human I-languages generate meanings.

4. Conclusion

A pattern emerges from the exercise conducted in this chapter. If one adheres to the idea that combining expressions is fundamentally an instruction to construct *conjunctive* concepts, along with the idea that open class lexical items are instructions to fetch concepts with independent content, one is led to say that certain aspects of syntax and various functional items are instructions to *convert* fetchable/constructable concepts into concepts that can be systematically conjoined with others. Perhaps this is the *raison d'etre* of syntax that goes beyond mere recursive concatenation: grammatical relations, like being the internal/external argument of a verb or determiner, can carry a kind of significance that is intriguingly like the kind of significance that prepositions have. These old ideas can be combined in a Minimalist setting devoted to asking which conversion operations are required by a spare conception of the recursive composition operations that human I-languages can invoke in directing concept assembly. The list of operations surveyed here is surely both empirically inadequate, and yet already too rich. My aim has been to offer a specific proposal as one illustration of minimalist thinking in semantics.