



Chung, Woojin. 2023. Local context is calculated domain by domain, for each maximal projection. *Glossa: a journal of general linguistics* 8(1). pp. 1–34. DOI: <https://doi.org/10.16995/glossa.5741>



Open Library of Humanities

## Local context is calculated domain by domain, for each maximal projection

Woojin Chung, Seoul National University, KR, [woojin@snu.ac.kr](mailto:woojin@snu.ac.kr)

---

Parsing-based theories of local contexts that either rely on strict linear order (Schlenker 2009) or c-command (Ingason 2016; Romoli & Mandelkern 2017) face a number of challenges, most notably in accounting for nominal modification (Schlenker 2020) and partitives (Anvari & Blumberg 2021) but also, depending on the theory, belief reports and coordination in head-final languages. To resolve such issues, I propose a hybrid theory which takes into account both linear order and syntactic structure: local context is calculated domain by domain, and for each maximal projection.

---

*Glossa: a journal of general linguistics* is a peer-reviewed open access journal published by the Open Library of Humanities. © 2023 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

**OPEN ACCESS**



## 1 Introduction

Local context has been one of the most thoroughly-studied topics in semantics and pragmatics in the past few decades, as it provides theoretical grounds for capturing a wide range of phenomena such as presupposition projection (Stalnaker 1974; Karttunen 1974; Heim 1983; Schlenker 2009), redundancy effects (Stalnaker 1974; Katzir & Singh 2014; Mayr & Romoli 2016), and antipresuppositions (Percus 2006; Chemla 2008; Sauerland 2008; Singh 2011). Despite its key role, there is no consensus as to how the local context of a linguistic expression should be determined. Traditional approaches such as Karttunen (1974) and Heim (1983) stipulated the order in which local contexts are updated by enriching the lexical semantics of linguistic expressions. However, such theories would be too strong to be sufficiently explanatory because one can encode an arbitrary update behavior into any given operator. For example, as Soames (1989) points out, one can come up with a deviant conjunction *and\** which updates the context in the opposite order of the ordinary conjunction *and*. The dynamic approaches in principle cannot rule out this possibility, leaving the *explanatory problem* for dynamic semantics (Soames 1982; Heim 1990; Schlenker 2009) unresolved.

- (1) The explanatory problem (as stated in Schlenker 2009)

Find an algorithm that predicts how any operator transmits presuppositions once its syntax and its classical semantics have been specified.

To address the explanatory problem, Schlenker (2009; 2010; 2011a; b) proposes a parsing-based account of local contexts that derives the local context of an expression on the basis of classical truth-conditional semantics. Informally speaking, the local context of an expression *E* is the strongest yet innocuous restriction that can be constructed based on the expressions that precede *E*. The key aspect of Schlenker’s theory is that local context is typically calculated *incrementally*: the interpreter traverses a string of expressions from left to right. At the point at which she calculates the local context of *E*, she is completely blind to everything that follows *E*. As a consequence, she needs to take into account every possible continuation that results in a well-formed sentence. This “left-to-right” bias is built into the computation algorithm of local contexts defined in (2).

- (2) Local context (Schlenker 2011, incremental version)<sup>1</sup>

The local context of an expression *d* of propositional or predicative type which occurs in a syntactic environment *a \_ b* in a context *C* is the strongest proposition or property *x* which guarantees that for any expression *d'* of the same type as *d*, for all strings *b'* for which *a d' b'* is a well-formed sentence,

$$C \models^{c' \rightarrow x} a (c' \text{ and } d') b' \leftrightarrow a d' b'$$

---

<sup>1</sup>  $C \models^{c' \rightarrow x} F$  indicates that each world *w* in *C* satisfies the formula *F* when *c'* denotes *x*, and  $\leftrightarrow$  stands for mutual entailment. The entire formula can be informally read as follows: it can be derived from the context *C* that *a (x and d') b'* and *a d' b'* mutually entail each other for any well-formed completion *b'*.

Schlenker shows that his incremental algorithm reproduces many of the predictions of the dynamic approaches. However, there are a number of challenges to the proposed algorithm. Ingason (2016) presents data from Japanese which hint that local contexts cannot be calculated in a strictly incremental fashion. He proposes that the order of context update mirrors structural hierarchy, that is, expressions that are higher in the structure updates the context prior to those that are lower. Romoli & Mandelkern (2017) formalize the hierarchy-based algorithm. More recently, Schlenker (2020) has presented yet another data that raise a problem for both his own incremental algorithm and the c-command-based alternative. He shows that certain English, Mandarin, and French nominals require their head noun to update the context before the modifiers. However, in such constructions the modifiers precede and also asymmetrically c-command the head noun. Thus both theories incorrectly predict that modifiers update the context before the head noun. Anvari & Blumberg (2021) make a similar point concerning the *Maximize Presuppositions* constraint in English partitive constructions, arguing that the observed competition demonstrates yet another case in which preceding expressions contribute to the local context of preceding expressions.

In this paper, I develop a hybrid theory which takes into account both linear order and syntax. While maintaining Schlenker’s view that the left-to-right bias inherent to parsing plays a crucial role in the calculation of local contexts, I submit that the algorithm should be run *domain by domain* rather than in a strictly incremental fashion, possibly *postponing* local context computation. In addition, I propose that local context is exclusively computed for each *maximal projection*, as opposed to each lexical item. I will demonstrate that these two innovations jointly resolve the issues raised in the literature, and additionally account for the projection behavior in belief reports in head-final languages.

## 2 Issues in extant parsing-based approaches

In this section, I will introduce four phenomena that are problematic to extant approaches: relative clauses (Ingason 2016), nominal modifiers (Schlenker 2020), partitives (Anvari & Blumberg 2021), and belief reports/coordination in head-final languages. Depending on the theory, some of these are accounted for, but no single theory explains all of the phenomena. **Table 1** summarizes the empirical coverage of the extant parsing-based accounts.

Key concept	Rel. clause	Nom. modifier	Partitives	Belief reports	Coord.
Linear order	✗	✗	✗	Δ	✓
C-command	✓	✗	✗	✓	✗

**Table 1:** Summary of extant parsing-based accounts.

The triangle in the belief reports column indicates that the strictly incremental theory can handle the phenomenon under an additional assumption, namely that the interpreter makes use of symmetric local contexts. See Section 5 for a related discussion.

## 2.1 Ingason (2016) on Japanese relative clause

Ingason (2016) claims that local context calculation cannot be sensitive to strict linear order, contra Schlenker’s original proposal. In particular, Ingason calls attention to the fact that head-final languages such as Japanese have a relative clause construction where the head noun follows the relative clause. However, under the widely-accepted assumption that redundancy effects are triggered when new information was already present in the common ground, it should be the head noun which updates the context before the relative clause. This would explain why (3a) does not feel redundant: the head noun *zyosei* ‘woman’ does not entail its relative clause *yamome-dearu* ‘who is a widow’. By contrast, (3b) sounds redundant because the head noun *yamome* ‘widow’ entails its relative clause ‘who is a woman’. Based on the view that the head noun c-commands its relative clause, Ingason proposes that syntactically higher elements update the context first.

- (3) a. Taro-ga [[**yamome**-dearu] **zyosei**-ni] atta.  
 Taro-NOM [[**widow**-COP] **woman**-DAT] met  
 ‘Taro met a woman who is a widow.’ (widow > woman)
- b. #Taro-ga [[**zyosei**-dearu] **yamome**-ni] atta.  
 Taro-NOM [[**woman**-COP] **widow**-DAT] met  
 ‘Taro met a widow who is a woman.’ (woman > widow)

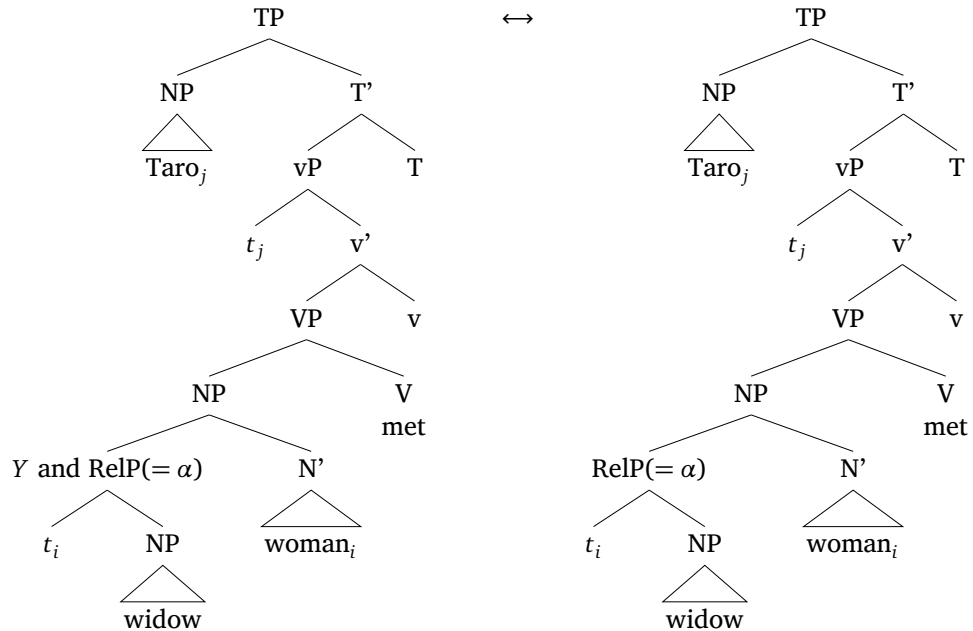
A formal implementation of the hierarchy-based algorithm can be found in Romoli & Mandelkern (2017), where the authors reform Schlenker’s algorithm in a way that local context is calculated at LF: when calculating the local context of *E* within a full clause *S*, the interpreter considers *only the expressions that asymmetrically c-command S at LF*, instead of considering the expressions that linearly precede it. Formally, the hierarchy-based version of local context is defined as follows:

- (4) Good-completion (Romoli & Mandelkern 2017)  
 A *good-completion* of *L* at  $\alpha$  is any well-formed LF which is identical to *L* except that any clause dominated or asymmetrically c-commanded by  $\alpha$  may be replaced by new material. For any sub-tree *Y*, a *Y-good-completion* of *L* at  $\alpha$  is any good completion of *L* at  $\alpha$  such that  $\alpha$  is replaced by a subtree beginning with [ *Y* [ and \_\_.
- (5) Hierarchical transparent local contexts (Romoli & Mandelkern 2017)  
 The local context of expression *E* in LF *L* and global context *C* is the strongest  $\llbracket Y \rrbracket$  s.t., where  $\alpha$  is the lowest node which dominates a full clause containing *E*, for all good-completions *D* of *L* at  $\alpha$ , and for all *Y-good-completions*  $D^Y$  of *L* at  $\alpha$ ,  $\llbracket D \rrbracket \cap C = \llbracket D^Y \rrbracket \cap C$ .

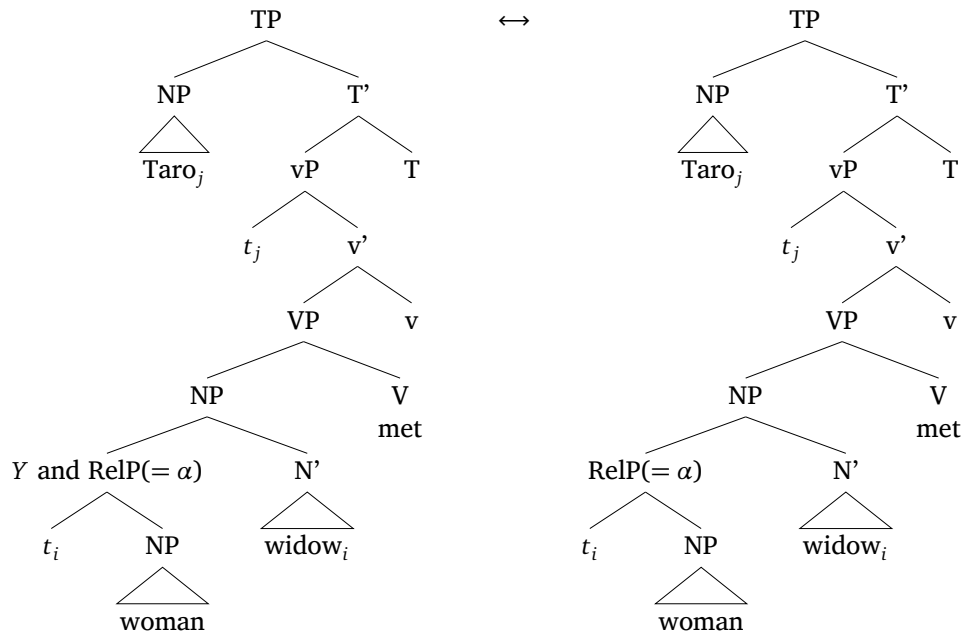
For illustration, consider Ingason’s (2016) examples (3a) and (3b), the LFs of which are respectively provided in (6a) and (6b). To calculate the local context of *widow* in (3a), we first have to find the  $\alpha$  which, by definition, is the smallest clause containing the expression. This will be the relative clause containing ‘ $t_i$  (is a) widow’. This  $\alpha$  dominates ‘ $t_i$  (is a) widow’ but does not dominate or asymmetrically c-command the remaining expressions in the LF (*Taro*, *woman*, *met*, ...), so the good-completions of the given LF at  $\alpha$  are well-formed LFs obtained by substituting ‘ $t_i$  (is a) widow’ with different expressions. For each good-completion  $D$ , we can create  $Y$ -good-completions  $D^Y$  by prefixing ‘[ $Y$  [ and’ to the relative clause. The local context is by definition the strongest  $\llbracket Y \rrbracket$  that makes each good-completion equivalent to the corresponding  $Y$ -good-completion.

The upshot is that structurally higher expressions update the context before the lower ones. In calculating the local context of the relative clause in (3a), only the expressions that asymmetrically c-command the relative clause are taken into account, hence *Taro*, *met*, and *woman*. So its local context can be restricted to the set of women that Taro met. Further updating the context with *widow* is informative, so the redundancy effect does not arise. By contrast, the local context of the relative clause in (3b) is the set of widows that Taro met. Thus, it would be redundant to further update the context with *woman*.

(6) a. (= (3a))



b. (= (3b))



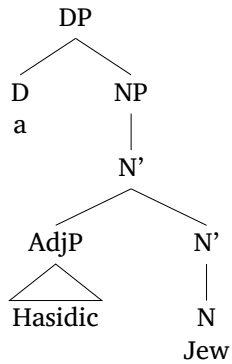
## 2.2 Schlenker (2020) on nominal modifiers

Despite its appeals, Schlenker (2020) points out that the c-command-based alternative, as well as his own strictly incremental algorithm, cannot explain redundancy effects arising from certain nominal modifications. He argues that the contrast in (7) is puzzling if the local context of the nominals and modifiers in (7a) and (7b) are calculated based on c-command. For instance, given the syntax of (7a) in (8a), the c-command-based algorithm incorrectly predicts that the adjective *Hasidic* first updates the context because it asymmetrically c-commands *Jew*. Had it been the case, it would have triggered a redundancy effect as *Jew* is entailed by *Hasidic*. The c-command-based algorithm also predicts that (7b), which is the mirror image of (7a), should not sound redundant, contrary to the intuition.

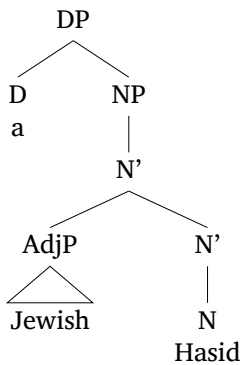
Schlenker also notes that this pair of examples is equally problematic for his strictly incremental algorithm, for the modifier precedes the head noun. He proposes a construction-specific analysis which basically stipulates that in these cases the head noun is visible to the interpreter when calculating the local context of the modifier.

- (7) *Context*: It is known that all Hasids are Jews, although there are Jews that are not Hasids. Under Nazi occupation, this heroic family hid...
- a. a Hasidic Jew.
  - b. #a Jewish Hasid.

(8) a. (= (7a))



b. (= (7b))



### 2.3 Anvari & Blumberg (2021) on partitives

In the literature on *Maximize Presuppositions* (henceforth MP) which has been claimed to be relativized to local contexts (Singh 2011), Anvari & Blumberg (2021) present the following contrast that is problematic for extant parsing-based approaches:<sup>2</sup>

- (9) a. #All of the two mainstream presidential candidates are crooked.  
 b. Both of the two mainstream presidential candidates are crooked.

Anvari & Blumberg first introduce the principle *Generalized Local MP*, which states that MP should be checked for “each constituent all the way down to particular lexical items”. They claim that (9a) violates MP because *both* is a stronger presuppositional variant of *all*, and this can only be captured if MP is localized, to the point that it is checked on the lexical item *all*. This is because the definite description “*the two mainstream presidential candidates*” already presupposes that there are exactly two candidates and both (9a) and (9b) have the same

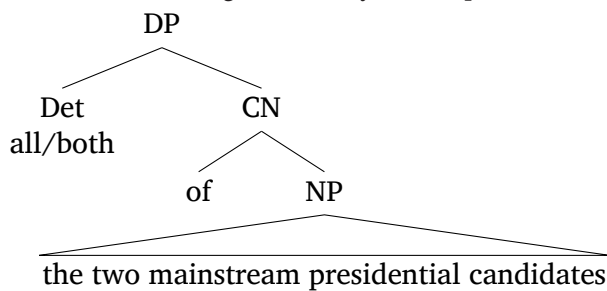
---

<sup>2</sup> This problem was initially noted in Anvari (2018), though without a solution.

presupposition. Had MP only been checked on a bigger constituent than *all*, (9a) should have not triggered an MP violation.

More importantly, Anvari & Blumberg point out that even if one adopts Generalized Local MP, extant parsing-based theories fail to correctly calculate the local context of *all/both*. This is primarily due to the fact that in doing so, the definite description “*the two mainstream presidential candidates*” should be taken into account but it neither precedes nor asymmetrically c-commands *all/both* given the structure Anvari & Blumberg posit:

(10) Anvari & Blumberg’s (2021) syntax of partitives



Since *all/both* linearly precedes the preposition and the definite description, the strictly incremental algorithm fails to consider the information that there are exactly two mainstream presidential candidates in calculating the local context of *all/both*. The c-command-based account fares no better, as *all/both* asymmetrically c-commands the definite description and the interpreter would ignore the information conveyed by the latter. Since neither linear order nor asymmetric c-command explains the aforementioned contrast, Anvari & Blumberg stipulate a construction-specific mechanism.

## 2.4 Belief reports in head-final languages

Unlike in head-initial languages, attitude verbs corresponding to English *believe* in head-final languages follow its complement, thereby posing a non-trivial problem for a strictly incremental theory. In head-initial languages like English, Schlenker’s strictly incremental algorithm works because *believe* precedes its complement. Consider (11), where the presupposition trigger *continue* is embedded within the attitude verb.

(11) John believes that Mary continues to smoke.

In calculating the local context of the embedded clause, Schlenker’s strictly incremental algorithm dictates that the interpreter takes into consideration all and only those expressions that precede the embedded clause. The bullet point in (12a) marks the boundary between the target expression (i.e., *d* in (2)) and a well-formed completion (i.e., *b'* in (2)).



- (12) a. John **believes that Mary continues to smoke** •  
 b. Corresponding equivalence:  
 For any expression  $d'$  of a propositional type,<sup>3</sup>  
 $C \models^{c' \rightarrow x}$  John believes ( $c'$  and  $d'$ )  $\leftrightarrow$  John believes  $d'$ ,  
 i.e.,  $\forall w' \in \text{Dox}_{\text{John}}(w): d'(w') = 1$  iff  $\forall w' \in \text{Dox}_{\text{John}}(w): (c' \wedge d')(w') = 1$

As the corresponding equivalence in (12b) shows, the attitude verb *believe* is visible to the interpreter, and Schlenker shows that the local context will be restricted to John's belief worlds.<sup>4</sup> Restricting the local context to such worlds does not affect the equivalence since the universal quantification is over those worlds. But further restricting the context could make the right hand side of the equivalence false. For instance, suppose that a stronger restriction leaves out a world  $w''$  and  $d'$  is a tautology. Then  $d'$  is true in all doxastically accessible worlds, but  $c' \wedge d'$  is false in the doxastically accessible  $w''$  and is false therein.

So far so good, but the situation gets complicated in head-final languages such as Korean where *mit* 'believe' follows its complement. Consider example (13) which includes a Korean translation of (11). To check the presuppositional status of the inference 'John believes that Mary used to smoke', the translation was embedded in a question.

- (13) John-un [Mary-ka (cikum-to) keysokhayse tambay-lul pi-n-tako]  
 John-TOP [Mary-NOM now-also continuously cigarette-ACC smoke-PRES-COMP]  
**mit-nun-ta-nun kes-i sasil-i-ni?**  
**believe-PRES-DECL-ADN thing-NOM truth-COP-Q**  
 'Is it true that John believes that Mary continues to smoke?'  
 (embedded clause > believe)

The presupposition of (13) remains identical to the English counterpart, which suggests that the local context of the embedded clause is the set of worlds compatible with John's beliefs at the world of evaluation. However, the strictly incremental algorithm makes a different prediction due to the fact that *mit* 'believe' follows the embedded clause. Omitting the question operator for simplicity, (14) demonstrates how the local context of the embedded clause is calculated.

<sup>3</sup> The equivalence does not make reference to the  $b'$  variable in (2), but this by no means is a departure from the Schlenker's view. In his analysis of belief reports, Schlenker assumes that the interpreter is informed that the sentence ends with the embedded clause, and  $b'$  invariably maps to the right parenthesis ) which marks the end of the embedded clause.

<sup>4</sup> This is an overly simplified view, as Schlenker (2009) notes that the embedded clause should not only be interpreted with respect to the world of evaluation (i.e., doxastically accessible worlds) but also the context of utterance. He thus assumes a Kaplanian double-indexing semantics (Kaplan 1989) where *believe* takes two world arguments rather than one, each representing the context of utterance and the world of evaluation. Accordingly, the embedded clause (hence the local context in consideration) is of type  $\langle s, \langle s, t \rangle \rangle$  rather than  $\langle s, t \rangle$ , where the first argument is the context of utterance. My argument holds regardless of whether one adopts a two-dimensional semantics, and I adopted a one-dimensional semantics merely for simplicity.

- (14) a. John-TOP [~~Mary-nom continuously smoke-pres-comp~~] • **believe**  
 b. Corresponding equivalence:  
 For any expression  $d'$  of a propositional type, and for all strings  $b'$  for which *John d'*  
 $b'$  is a well-formed sentence,  
 $C \models^{c' \rightarrow x} \text{John } (c' \text{ and } d') b' \leftrightarrow \text{John } d' b'$

The interpreter does not have access to the attitude verb, and is only aware that the sentence has something to do with John. One possible well-formed completion is *malha(y)* ‘say’, which does *not* signal that the embedded clause delivers information regarding John’s beliefs:

- (15) John-un [Mary-ka (cikum-to) keysokhayse tambay-lul pi-n-tako]  
 John-TOP [Mary-NOM now-also continuously cigarette-ACC smoke-PRES-COMP]  
**malhay-ss-ta.**  
**say-PAST-DECL**  
 ‘John says that Mary continues to smoke.’

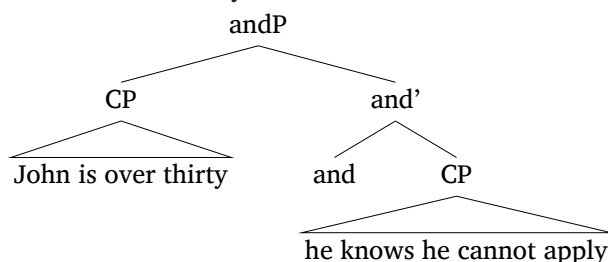
Since the strictly incremental algorithm quantifies over all possible well-formed completions and seeks the strongest yet innocuous restriction, the interpreter cannot narrow down the context set to John’s doxastic worlds, contrary to the intuition.

Note that the c-command-based approach does not suffer from this issue. The standard syntactic assumption is that embedded clauses do not asymmetrically c-command embedding predicates, so the theory predicts that attitude verbs invariably update the context before their complements. However, the c-command-based account in its current form faces a challenge in dealing with coordination in head-final languages.

## 2.5 Coordination in head-final languages

Coordination in head-final languages manifests a different structural relation between two conjuncts/disjuncts and favors accounts sensitive to linear order as opposed to c-command. In head-initial languages such as English, the left conjunct precedes *and* asymmetrically c-commands the right conjunct, making the predictions of the two theories at hand indistinguishable.

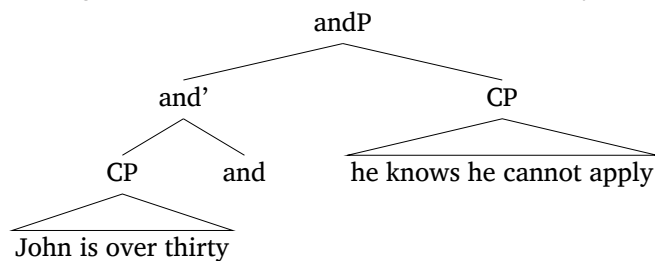
- (16) John is over thirty and he knows he cannot apply.  
 (17) Structural hierarchy mirrors linear order in head-initial languages



This nice alignment breaks in head-final languages such as Korean. The conjunction operator is suffixed to the left conjunct as shown in (18), indicating that the two form a constituent. As a consequence, the right conjunct sits higher in structure than the left conjunct, which is illustrated in (19).

- (18) John-i selun-i nem-ess-ko ku-ka caki-ka ciwenha-l swu eps-∅-ta-nun  
 John-NOM thirty-NOM over-PERF-and he-NOM self-NOM apply-cannot-PRES-DECL-ADN  
 kes-ul a(l)-n-ta-nun kes-i sasil-i-ni?  
 thing-ACC know-PRES-DECL-ADN thing-NOM truth-COP-Q  
 ‘Is it true that John is over thirty and he knows he cannot apply?’

- (19) Misalignment between linear order and hierarchy in head-final languages



The c-command-based account then incorrectly predicts that the right conjunct updates the context before the left conjunct. However, (18) has the same conditional presupposition as its English counterpart in (16), hinting that it is the left conjunct that first updates the context. Note also that the same issue arises with disjunction: despite the fact that the right disjunct asymmetrically c-commands the left disjunct in (20), the sentence does not carry the conditional presupposition that if John isn't depressed then he is incompetent but rather presupposes that John is incompetent. This suggests that the order of context update is invariably left to right.

- (20) John-un casin-i mwununghata-nun kes-ul al-kena/tunci, (ku-ka)  
 John-TOP self-NOM incompetent-ADN thing-ACC know-or he-NOM  
 wuwulcung-ey kelliess-∅-ta-nun kes-i sasil-i-ni?  
 depression-DAT is.suffering-PRES-DECL-ADN thing-NOM truth-COP-Q  
 ‘Is it true that John knows that he is incompetent or he is depressed?’

- (21)
- 
- ```

graph TD
  orP[orP] --- or_prime[or']
  orP --- CP1[CP]
  or_prime --- CP2[CP]
  or_prime --- or[or]
  CP2 --- John[John knows he is incompetent]
  CP1 --- he[he is depressed]
  
```

Let me point out that the coordination data do not offer a conclusive argument. As Ingason (2016) notes, there is no consensus on the syntax of coordination and the topic remains rather controversial. Nevertheless, suffixation (i.e., coordinators are suffixed to the left conjunct/disjunct in Korean) is a strong case in favor of the syntax postulated in this section. Unless an equally convincing piece of evidence for a revisionary syntax is provided, it seems reasonable to assume that what you see is what you get.

Romoli & Mandelkern (2017) point out that an alternative line of thought can be developed along the lines of Chierchia (2009). Chierchia proposes that the order of context update mirrors the order of semantic composition. Informally put, an argument that first composes with the functor updates the context before other arguments. This would nicely explain the Korean coordination case above, as it is the left conjunct/disjunct which first composes with the functor. But a problem immediately occurs in English, where *and/or* first composes with the right conjunct/disjunct. Chierchia thus suggests a revisionary syntax for English coordination, where a silent *both/either* carries the meaning of conjunction/disjunction and *and/or* is semantically vacuous.

(22) Silent *both/either* carries conjunction/disjunction meaning (Chierchia 2009)



Romoli & Mandelkern note that while this gets the job done, it requires a substantial revisionary assumption of coordination and calls for independent syntactic motivations. Moreover, they doubt that the empirical coverage of the theory can extend to relative clauses and belief reports in head-final languages.

## 2.6 Generalization of the observation

From the strictly incremental algorithm's point of view, Japanese relative clauses, English nominal modifiers and partitives, and Korean belief reports all share the same problem: the interpreter needs to parse certain expressions that follow the target before she calculates its local context. For Japanese relative clauses and English nominal modifiers, she needs to take the head noun into consideration, and for the quantifiers *all* and *both* in English partitives, she needs to consider all expressions in the nominal domain. As for clauses embedded under *mit* 'believe' in Korean, she has to first process the preceding matrix predicate before calculating the local context of the embedded clause. For the first group of phenomena, at least the entire nominal domain has to be parsed before local context calculation. For an analysis of embedded clauses in Korean, the entire matrix vP which embeds the target clause has to be parsed in advance. It appears that the interpreter is not entertaining a strictly incremental strategy but rather, she *waits* until the aforementioned expressions have been parsed.

Also, not every expression contained within the nominal domain is taken into consideration. For instance, when the target expression is a head noun, e.g., *Jew* in *a Hasidic Jew*, the nominal modifier has to be ignored. In the c-command-based account, all expressions that are asymmetrically

c-commanded by the target expression are ignored.<sup>5</sup> But since English nominal modifiers and partitives pose a problem for both the strictly incremental algorithm and the c-command-based alternative, we need a different way to sort out what should be considered and what should not be. In what follows, I submit that there are existing linguistic notions that are well-motivated and well-suited for the task.

### 3 Proposal

I maintain Schlenker’s view that the left-to-right bias in parsing influences local context computation. In light of the two observations just made above, I make two corresponding adjustments to Schlenker’s strictly incremental algorithm.

The first adjustment, which I call *domain-by-domain evaluation*, captures the observation that the interpreter seems to consider certain expressions that follow the target expression when calculating its local context. Procedurally, the interpreter first parses a certain chunk of expressions, and then she calculates the local contexts of the expressions within that chunk. The equivalence is calculated only after she parses a certain chunk of expressions. All parsed expressions, and *only those expressions*, are potentially considered in the equivalence. Expressions that are yet to be parsed will be ignored in local context computation.

Why not immediately calculate the local context of an expression as soon as the interpreter parses it? Granted that Schlenker’s equivalence is semantic in nature, I suggest that the interpreter calculates local contexts only when she can access the semantic values of target expressions.<sup>6</sup> And it is commonly assumed that access to the semantic values of expressions is limited to certain points in the structure building process. Chomsky’s phase theory (2000; 2001; 2008) is more or less the standard view, where the semantic information of lexical items is shipped to the CI interface for LF interpretation upon construction of either vP or CP (i.e., phases).<sup>7,8</sup> Depending

---

<sup>5</sup> Precisely speaking, Romoli & Mandelkern’s (2017) theory predicts that if two subclausal expressions are contained within the same clause then their local contexts will be identical. In calculating the local context of an expression  $E$ , their theory first seeks  $\alpha$  which is the smallest clause containing  $E$ , and keeps this  $\alpha$  constant for calculating good-completions. It follows that any two subclausal expressions contained in the same clause entertain the same good completions, yielding indistinguishable local contexts. In the main text, I assume that there would be a natural way to extend Romoli & Mandelkern’s theory to address this issue.

<sup>6</sup> Barker (2022) takes a different route to tackle this issue. Instead of relying on syntactic parsing, he suggests that local context computation is a purely semantic process. He presents a novel way of composing continuized expressions where the first argument of a continuized expression corresponds to its incremental local context. Local contexts are defined purely in semantic terms, eliminating much of the syntactic complexity present in parsing-based theories. Another notable advantage of this view is that depending on the lexical specification of a presupposition trigger, it becomes possible to impose a presuppositional requirement which is sensitive to expressions that follow the trigger such as Anvari & Blumberg’s (2021) partitive examples.

<sup>7</sup> Chomsky (2000) assumes that the entire phase is Spelled-Out, but he later changes his view and claim that it is the complement of a phase head that is Spelled-Out (Chomsky 2001; 2008). While the latter is currently the standard view, Bošković (2016) presents several empirical arguments as well as theory-internal ones that favor the former, and also points out that Chomsky’s more recent work on labeling algorithm (Chomsky 2013; 2015) favors the Spell-Out of the entire phase.

<sup>8</sup> An anonymous reviewer notes that entailment is a “generalizable notion which can be applied to any expression that has a type that end in  $T$ ”, and thus entailment per se does not motivate domain-by-domain evaluation. However, my

on one’s perspective, DPs (Matushansky 2005) or the highest phrase in the extended projection of any lexical category (Bošković 2014) are also taken to constitute a phase. In this work, I will adopt the phase theory and assume that phases are *designated domains* for semantic interpretation.

(23) **Adjustment 1: Domain-by-domain evaluation**

The interpreter parses a sentence from left to right, but the local context of an expression can be calculated only at points where the interpreter has access to the semantic values of the parsed expressions.

An immediate consequence of the domain-by-domain evaluation is that although the interpreter attempts to calculate the local context of a target expression as soon as possible, she will have to postpone the calculation if the parsed expressions altogether do not constitute a designated domain (i.e., phase), as their semantic values cannot yet be retrieved.<sup>9</sup> But as soon as the parsed expressions form a designated domain and phase Spell-Out can take place, the interpreter will immediately retrieve the semantic value of the target expression along with the semantic values of other parsed expressions and calculate the strongest yet innocuous restriction based on the semantic values she has access to.

The second adjustment aims to correctly determine which parsed expressions should or should not be considered in the calculation of equivalences. I propose that local context calculation does not target a lexical item (i.e., terminal nodes in a syntactic tree), but rather its *maximal projection*. The maximal projection of a head *X* is the phrase *XP* which is the highest level to which *X* projects (Chomsky 1995).<sup>10</sup> So for instance, if the interpreter targets a noun (i.e., an *N* head), she

---

claim is orthogonal to whether entailment is a generalizable notion or not, as it concerns how the syntactic component of the language faculty constrains access to the semantic values.

<sup>9</sup> An anonymous reviewer questions how the interpreter can properly parse a double nominative construction in Korean. As exemplified below, at the point when the interpreter just processed the embedded clause there are two possible parses, one treating *mwusep-* ‘be afraid of’ as the embedded predicate (correct parse) and the other treating it as the matrix predicate (wrong parse).

(i) John-i Mary-ka mwusep-ta(-ko) sayngkakha-n-ta.  
 John-TOP Mary-NOM scary/be-afraid-of-DECL-COMP think-PRES-DECL  
 ‘John thinks that Mary is scary.’

(ii) Correct parse: [<sub>CP</sub> John-i [<sub>CP</sub> Mary-ka mwusep-ta ] ... ]

(iii) Wrong parse: [<sub>CP</sub> John-i Mary-ka mwusep-ta ... ]

Schlenker (2009) stipulates that the interpreter has access to full syntactic structure even though she does not know what lexical items would follow the target expression. Technically speaking, the parsed string contains parentheses which contain syntactic information. Thus structural ambiguity during processing is not problematic, but it remains questionable whether this assumption was adequately justified. We could alternatively adopt a theory which builds structure in real time—during parsing (cf. Phillips 1996)—and assume, as the reviewer suggests, that local contexts calculated from wrong parses are discarded after the interpreter figures out the correct parse.

<sup>10</sup> Chomsky (1995) conjectures that “only maximal projections seem to be relevant to LF-interpretation” and other bar-level projections are “invisible at the interface and for computation”, which, if on the right track, potentially supports Adjustment 2. Mutual entailment, which plays a key role in local context calculation, is a product of semantic interpretation based on LF.

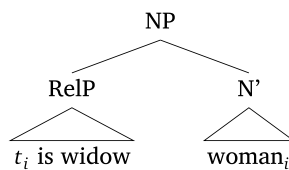
has to calculate the local context of its maximal projection, which is an *NP*. Consequently, she not only ignores the meaning of the noun but also the information contributed by all expressions dominated by the *NP* when she postulates the equivalence for local context calculation.

(24) **Adjustment 2**

Local context is calculated for each maximal projection

Note that this idea particularly resembles Anvari & Blumberg’s (2021) ‘Generalized Local MP’, in the sense that it checks local contexts of sub-clausal expressions. Though similar in spirit, it makes an additional prediction about which expressions are considered and which are excluded in the postulation of the equivalences. Consider again Japanese relative clauses. Recall that the c-command-based analysis predicts that the local context of *woman* in (25) does not contain the information that the denoted set is a set of widows because the head noun *woman* asymmetrically c-commands the relative clause “ $t_i$  is widow”.<sup>11</sup> In the proposed account, the interpreter will ignore the relative clause because if she were to target the head noun *woman*, she would have to target its maximal projection NP. And since the relative clause is dominated by the NP, it will be replaced by the  $d'$  variable in the equivalence together with the head noun.

(25) NP dominates RelP



A formal implementation of the algorithm that encodes both adjustments is provided below:

(26) **Revised incremental algorithm**

The local context of a maximal projection  $d$  which occurs in a syntactic environment  $a \_ b$   $b'$  (where  $a \_ d \_ b$  constitute the smallest designated domain that contains  $a \_ d$ ) in a context  $C$  is the strongest proposition or property  $x$  which guarantees that for any expression  $d'$  of the same type as  $d$ ,

$$C \models^{c' \rightarrow x} a(c' \text{ and } d') \_ b \leftrightarrow a \_ d' \_ b$$

where  $\leftrightarrow$  is a generalized mutual entailment

The revised incremental algorithm differs from Schlenker’s original version in several respects: First, it does not calculate the local context of a lexical item but rather, it calculates the local context of each maximal projection. Second, it does not quantify over all well-formed completions in postulating the equivalence. Instead, the variable that represents well-formed completions

---

<sup>11</sup> Precisely speaking, this is not a guaranteed truth. Depending on one’s theory of syntax, *woman* could be an N head which projects to an N’ head, which is the sister of RelP. Under this assumption, the N head does not c-command the relative clause.

(i.e.,  $b'$  in (2)) is replaced with a fixed expression  $b$  such that (i)  $b$  immediately follows the target expression  $d$  and (ii)  $a d b$  is the smallest designated domain that contains the string “ $a d$ ”. Despite the fact that certain expressions that follow the target  $d$  is considered, the algorithm reflects the left-to-right bias in parsing: given a sentence “ $a d b b'$ ”, every expression that precedes the target  $d$  (represented with  $a$ ) will be considered in postulating the equivalence, but whatever follows the string “ $a d b$ ” (represented with  $b'$ ) will be ignored. From a processing point of view, the string “ $a d b$ ” is the shortest string of expressions that the interpreter needs to parse before she gets to retrieve the semantic value of the target  $d$ . Once she has access to the semantic value of the target, she will calculate its local context solely based on the expressions she has parsed.

In what follows, I will present detailed analyses of the problematic examples. For each analysis, I will first depict the procedures the interpreter goes through, and then give a formal account based on the the equivalence in (26).

## 4 Analysis

I will assume that vPs and CPs are phases and therefore are designated domains. In addition, I follow Bošković (2014) in that the highest phrase in the extended projections of a nominal is a phase. This implies that if a language lacks a determiner (e.g., Japanese, Korean, ...) then NPs will function as a phase.<sup>12</sup> In the illustrations to follow, I will mark designated domains with rounded boxes. The round boxes are useful in checking whether a string of parsed expressions is a designated domain or not: a string of parsed expressions corresponds to a designated domain if and only if there exists a round box that just surrounds the string. I will use boldfaced font for the expressions that the interpreter has parsed.

### 4.1 Relative clause

The revised incremental algorithm accounts for the contrast between (3a) and (3b), repeated below as (27a) and (27b), respectively. Their syntax is depicted in (28). The generalization was that the relative clause must denote a more specific set of individuals than the head noun, independently of the relative word order between the two.

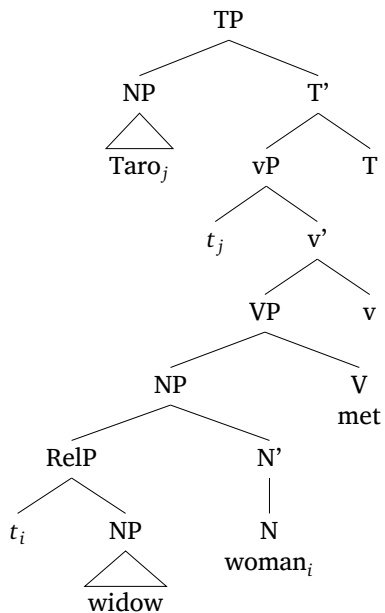
- (27) a. Taro-ga [[**yamome**-dearu] **zyosei**-ni] atta.  
 Taro-NOM [[**widow**-COP] **woman**-DAT] met  
 ‘Taro met a woman who is a widow.’
- b. #Taro-ga [[**zyosei**-dearu] **yamome**-ni] atta.  
 Taro-NOM [[**woman**-COP] **widow**-DAT] met  
 ‘Taro met a widow who is a woman.’

---

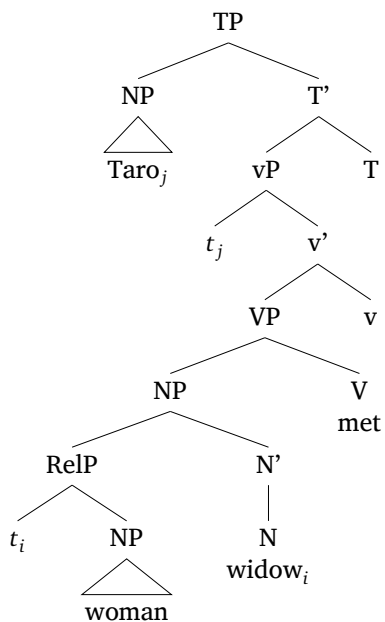
<sup>12</sup> Bošković’s (2014) claim not only concerns the extended projections of nominals but also that of verbs and other lexical categories, but this shouldn’t affect the predictions of the present proposal.



(28) a. (= (27a))



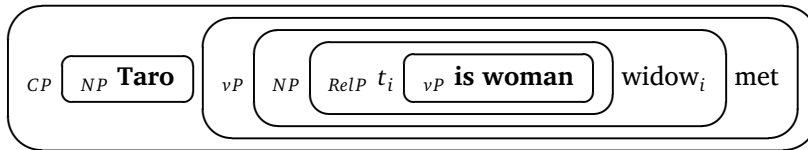
b. (= (27b))



I will just give an analysis of the redundant (27b), as it alone offers a good explanation of the asymmetry. Procedurally, the interpreter parses the sentence from left to right, first processing *Taro*. Since *Taro* on its own projects to an NP—the highest projection in the extended nominal domain in Japanese—it forms a designated domain and the interpreter calculates its local context based on what has been parsed. I will skip the calculation since it is trivial and is no different from Schlenker’s strictly incremental algorithm.

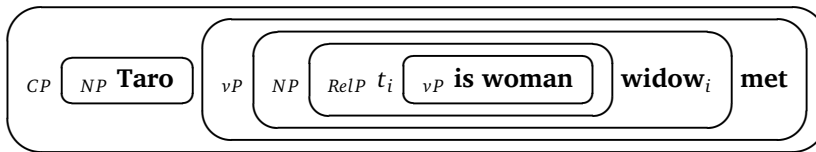
The interpreter further parses *is woman*. She cannot immediately calculate the local context of the relative clause at this point because the parsed string “*Taro t<sub>i</sub> is woman*” is not a designated domain—note that no round box contains just this string. Adjustment 1 (domain-by-domain evaluation) prevents her from fetching the semantic content of the parsed string and local context calculation has to be delayed. For the same reason, the interpreter still does not have semantic access to the parsed strings even after she has parsed *widow*.

(29) The interpreter sequentially parses *Taro*, and *is woman*



Upon processing *met*, the interpreter gets to retrieve the semantic content of the parsed expressions and calculate local contexts. In doing so, Adjustment 2 takes effect and dictates that the interpreter only targets maximal projections of newly parsed expressions. I will focus on the RelP “*t<sub>i</sub> is woman*” and the NP “*t<sub>i</sub> is woman widow<sub>i</sub>*”.

(30) The interpreter sequentially parses *widow<sub>i</sub>* and *met*



To calculate the local context of the RelP, the interpreter takes into consideration *Taro*, *widow*, and *met*. The table in (31a) maps each alphabetical variable in (26) to the corresponding expression. The variable *a* in the table represents whatever precedes the target maximal projection *d*, which is *Taro*. The target maximal projection *d* is the RelP “*t<sub>i</sub> is woman*” and in the corresponding equivalence, it will be replaced with the variable *d'* bound by a universal quantifier. The variable *b* stands for what *follows* the target maximal projection but is part of the smallest designated domain that contains the string “*a d*”. Since the string “*a d*” maps to “*Taro t<sub>i</sub> is woman*”, the variable *b* is the string “*widow<sub>i</sub> met*” because “*Taro t<sub>i</sub> is woman widow<sub>i</sub> met*” forms a CP phase. Lastly, *b'* is what follows the string “*a d b*”, which in this case is an empty string  $\emptyset$ . The corresponding equivalence is given in (31b) and the strongest yet innocuous restriction is the set of widows that *Taro* met.

(31) The local context of the RelP

a. Mapping parsed expressions to variables

| <i>a</i> | <i>d</i>                      | <i>b</i>  | <i>b'</i>   |
|----------|-------------------------------|-----------|-------------|
| Taro     | <i>t<sub>i</sub> is woman</i> | widow met | $\emptyset$ |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{Taro } (c' \text{ and } d') \text{ widow met} \leftrightarrow \text{Taro } d' \text{ widow met}$$

The equivalence in (26) reflects the procedure just described: to calculate the local context of the target maximal projection  $d$ , the interpreter takes into consideration the string “ $a d b$ ”, which is by definition the smallest designated domain that contains the string “ $a d$ ”. The presence of the  $b$  variable in the equivalence guarantees that she has to parse a full designated domain prior to calculating the local context of  $d$ .

Let us contrast the above result with the local context of the NP “ $t_i$  is woman widow $_i$ ”. In (28b), the maximal projection NP dominates RelP, and the consequence is that the  $d$  variable contains the relative clause. *Taro* and *met* maps to the variables  $a$  and  $b$ , respectively, and the interpreter posits the equivalence in (32b). It follows that the strongest yet innocuous restriction is the set of individuals that *Taro met*.

(32) The local context of the NP

a. Mapping parsed expressions to variables

|      |                      |     |             |
|------|----------------------|-----|-------------|
| $a$  | $d$                  | $b$ | $b'$        |
| Taro | $t_i$ is woman widow | met | $\emptyset$ |

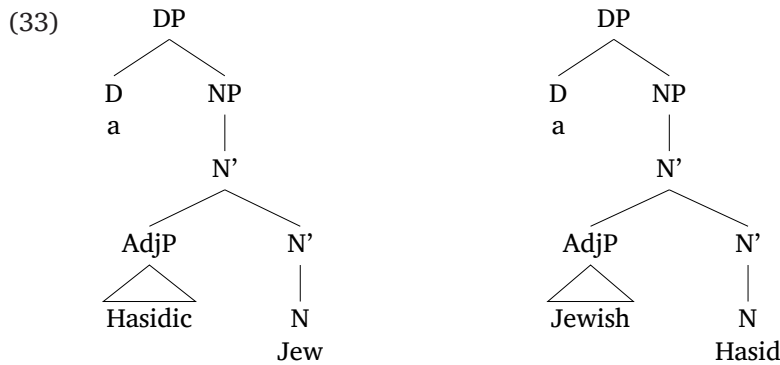
b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{Taro } (c' \text{ and } d') \text{ met} \leftrightarrow \text{Taro } d' \text{ met}$$

In short, calculating the local context of the RelP requires considering the information provided by the head noun, but the contribution of the RelP must be ignored in calculating the local context of the NP. This correctly predicts the asymmetry between RelP and NP concerning redundancy effects: the relative clause must be more specific than the head noun. Example (27b) sounds redundant because the head noun is more specific than the relative clause: the local context of the relative clause has already been restricted to the set of widows that *Taro met*, and the contribution of the relative clause such that the set consists of women is redundant. By contrast, in (27a), the relative clause denotes a more specific set of individuals (i.e., *widow*) than the head noun (i.e., *woman*), so we do not observe a redundancy effect. The result is consonant with the c-command-based account but does not require making reference to c-command, which, as discussed earlier, causes problems in analyzing nominal modifiers, partitives, and coordination in head-final languages.

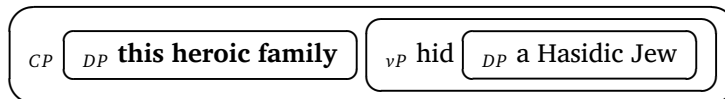
## 4.2 Nominal modifiers

The revised incremental algorithm does not entertain the notion of c-command and yet is capable of capturing the intuition that the head noun is evaluated before the nominal modifier. I will focus on the DP in (7a) since the analysis of (7b) will then become trivial. First recall that the syntax of the DPs in (7a) and (7b) are as follows:



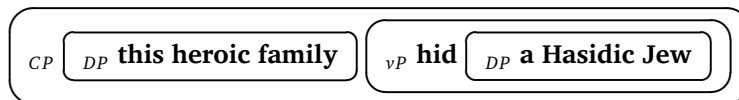
The interpreter traverses the sentence from left to right. First she sequentially parses *this, heroic, and family*, and at this point she has processed a full DP which is a designated domain. The local contexts of the maximal projections within the DP will then be calculated. I will omit the details since we are interested in the order of context update within the object.

(34) The interpreter sequentially parses *this, heroic, and family*



The interpreter goes on and parses the rest of the sentence. Once she reaches the end of the sentence, she will have another parsed designated domain, which is the CP. She calculates the local contexts of the maximal projections within the newly parsed expressions, that is, vP, DP, AdjP, and NP.

(35) The interpreter sequentially parses *hid, a, Hasidic, and Jew*



The local context of the NP is computed as follows: the table in (36a) maps each variable in the equivalence to the corresponding expressions. Variable *a* maps to the string of expressions preceding the target maximal projection, which in this case is “*this heroic family hid a*”. The target is the NP “*Hasidic Jew*”; note that the NP dominates *Hasidic* so the AdjP is contained within the target maximal projection. No expression follows the target maximal projection, so *b* and *b'* are both empty strings. This mapping gives rise to the equivalence in (36b), which does not contain the information that the individual that this heroic family hid is a Hasid.

(36) The local context of the NP “*Hasidic Jew*”

a. Mapping parsed expressions to variables

|                          |             |          |           |
|--------------------------|-------------|----------|-----------|
| <i>a</i>                 | <i>d</i>    | <i>b</i> | <i>b'</i> |
| this heroic family hid a | Hasidic Jew | ∅        | ∅         |

b. Corresponding equivalence

$C \models^{c' \rightarrow x}$  this heroic family hid a (*c'* and *d'*)  $\leftrightarrow$  this heroic family hid a *d'*

On the other hand, the prediction is that the local context of the AdjP *Hasidic* contains the information conveyed by the head noun. Variable *a* maps to the same string as above, but variable *b* is not an empty string but rather the head noun *Jew*, because the AdjP does not dominate the head noun. The corresponding equivalence is given in (37b), which manifests that the individual that this heroic family hid is a Jew.

(37) The local context of the AdjP “*Hasidic*”

a. Mapping parsed expressions to variables

|                          |          |          |           |
|--------------------------|----------|----------|-----------|
| <i>a</i>                 | <i>d</i> | <i>b</i> | <i>b'</i> |
| this heroic family hid a | Hasidic  | Jew      | ∅         |

b. Corresponding equivalence

$C \models^{c' \rightarrow x}$  this heroic family hid a (*c'* and *d'*) Jew  $\leftrightarrow$  this heroic family hid a *d'* Jew

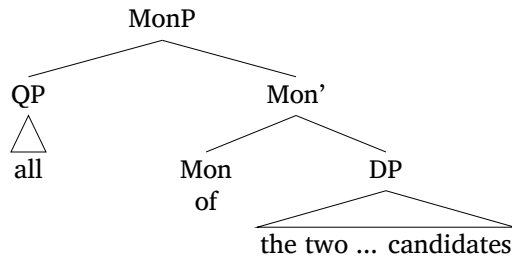
In short, the reason why the head noun appears to be evaluated before the AdjP is because the head noun’s maximal projection, NP, dominates the AdjP. It follows that the AdjP is contained within the target maximal projection *d*. By contrast, the AdjP does not dominate the head noun so the head noun will not be replaced with the *d* variable. While c-command does not seem fit for the task, the update behavior receives a principled account in terms of domination of maximal projections—yet another core syntactic notion.

### 4.3 Partitives

Let me also point out that Anvari & Blumberg’s (2021) contrast (repeated below as (38)) can be accounted for without requiring a lexical-level local context calculation. Recall that Anvari & Blumberg assume a specific syntax of partitive constructions where the quantifier takes the *of*-phrase as its complement. While the revised incremental algorithm in (26) would not calculate the local context of *all* had this been the underlying syntax, there are prominent alternatives such as Schwarzschild (2006) where the entire QP *all* sits at the specifier position of an independent projection MonP:

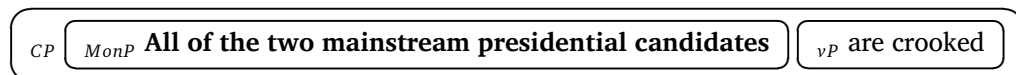
- (38) a. #All of the two mainstream presidential candidates are crooked.  
 b. Both of the two mainstream presidential candidates are crooked.

(39) Alternative syntax (Schwarzschild 2006)



Given the above syntax, the quantifier *all* does not take a complement and directly projects to a QP. The upshot is that my theory will calculate the local context of the QP without ignoring the preposition and the DP, capturing Anvari & Blumberg’s intuition. Procedurally, the interpreter parses the sentence from left to right, until she reaches the end of the subject. The MonP is the highest projection in the extended nominal domain, so given the current assumptions it functions as a phase (cf. Bošković (2014)). Having parsed a designated domain, the interpreter begins calculating the local contexts of the maximal projections within the MonP.

(40) The interpreter parses *All, of, the, two, mainstream, presidential, and candidates*



Let us focus on the QP “*all*”. As shown in the following table, *a* maps to an empty string because nothing precedes the target maximal projection. The *b* variable maps to “*of the two mainstream presidential candidates*”, as this is what follows the target and is minimally required for the parsed expressions to form a designated domain. The *b'* variable maps to “*are crooked*”, which are expressions yet to be processed. Given this mapping, the interpreter postulates the equivalence in (41b), which correctly takes into consideration the information that there are exactly two presidential candidates. As a consequence, *all* loses to *both* in MP competition.

(41) The local context of the QP “*all*”

a. Mapping parsed expressions to variables

| <i>a</i> | <i>d</i> | <i>b</i>                                      | <i>b'</i>   |
|----------|----------|-----------------------------------------------|-------------|
| ∅        | all      | of the two mainstream presidential candidates | are crooked |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} (c' \text{ and } d') \text{ of the two mainstream presidential candidates} \\ \leftrightarrow d' \text{ of the two mainstream presidential candidates}$$

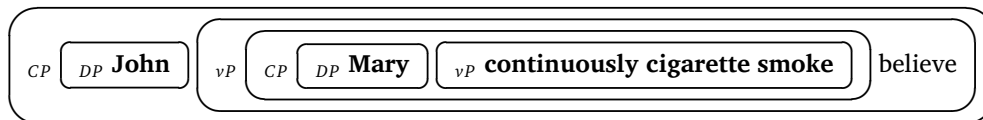
#### 4.4 Belief reports in head-final languages

The domain-by-domain evaluation guarantees that *mit* ‘believe’ which follows the embedded clause in (13) (repeated below as (42)) is processed before the local context of the embedded clause is calculated.

- (42) John-un [Mary-ka (cikum-to) keysokhayse tambay-lul pi-n-tako]  
 John-TOP [Mary-NOM (now-also) continuously cigarette-ACC smoke-PRES-COMP]  
**mit-nun-ta-nun kes-i sasil-i-ni?**  
**believe-PRES-DECL-ADN thing-NOM truth-COP-Q**  
 ‘Is it true that John believes that Mary continues to smoke?’

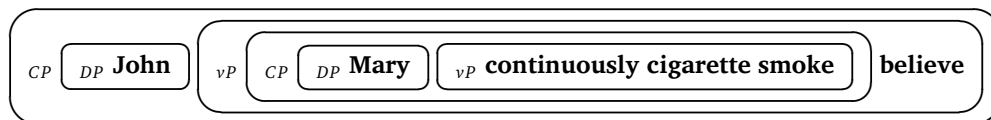
In the analysis, I will omit the question operator for simplicity. The interpreter parses the sentence from left to right, first processing *John*. Since *John* constitutes a designated domain on its own, the interpreter calculates its local context. She then parses *Mary*, *continuously*, *cigarette*, and *smoke*. At this point, she has parsed the matrix subject and the embedded clause, but she will not calculate the local context of the embedded clause because “*John Mary continuously cigarette smoke*” is not a designated domain. This is manifested by the fact that no round box contains exactly these expressions:

- (43) The interpreter sequentially parses *John*, *Mary*, *continuously*, *cigarette*, and *smoke*



The interpreter further parses the matrix verb *believe*, and at this point she has access to the semantic values of the matrix *vP* and the embedded *CP*. To calculate the local context of the embedded *CP*, the interpreter maps the variables *a*, *d*, *b*, and *b'* to “*John*”, “*Mary continuously cigarette smoke*”, “*believe*”, and an empty string, respectively. The corresponding equivalence is provided in (45b), which encodes the crucial information that the sentence conveys John’s beliefs. Despite the fact that the Korean example exhibits a different word order, the prediction is no different from Schlenker’s (2009) analysis of the English counterpart (cf. (12b)).

- (44) The interpreter additionally parses *believe*



- (45) The local context of the embedded *CP* “*Mary continuously cigarette smoke*”

- a. Mapping parsed expressions to variables

| <i>a</i> | <i>d</i>                          | <i>b</i> | <i>b'</i> |
|----------|-----------------------------------|----------|-----------|
| John     | Mary continuously cigarette smoke | believe  | ∅         |

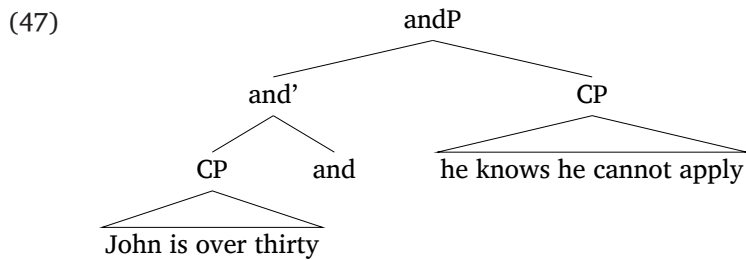
- b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{John } (c' \text{ and } d') \text{ believe} \leftrightarrow \text{John } d' \text{ believe}$$

## 4.5 Coordination

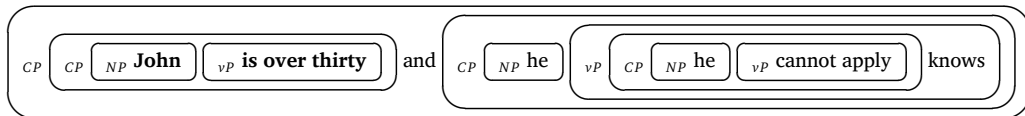
The revised incremental algorithm predicts that sentential coordination is evaluated in an incremental fashion, that is, the left conjunct/disjunct is evaluated before the right conjunct/disjunct. The coordination data were a problem for the c-command-based account because in head-final languages, the right conjunct/disjunct asymmetrically c-commands the left conjunct/disjunct. In what follows, I only present my analysis of conjunction, as the way in which the interpreter processes disjunction is more or less identical to the conjunction case. The Korean conjunction example in (18) and its syntax (19) are repeated below as (46) and (47), respectively.

- (46) John-i selun-i nem-ess-ko ku-ka caki-ka ciwenha-l swu eps-Ø-ta-nun  
 John-NOM thirty-NOM over-PERF-and he-NOM self-NOM apply-cannot-PRES-DECL-ADN  
 kes-ul a(l)-n-ta-nun kes-i sasil-i-ni?  
 thing-ACC know-PRES-DECL-ADN thing-NOM truth-COP-Q  
 ‘Is it true that John is over thirty and he knows he cannot apply?’



I will omit the question operator for simplicity. The interpreter parses the sentence from left to right, stops when she parses the NP subject and calculates the local context of *John*. She then parses the entire left conjunct, the point at which she has access to the semantic values of the entire CP and the vP. Regarding the local context of the CP, no expression precedes the parsed CP and no expression that has been parsed follows the CP, therefore the interpreter constructs the table in (49a). Since no parsed expression is taken into consideration in the equivalence in (49b), the strongest yet innocuous restriction is the global context.

- (48) The interpreter parses the left conjunct<sup>13,14</sup>



<sup>13</sup> The standard assumption is that the auxiliary *can* moves to the *T* head, but this does not affect my analysis.

<sup>14</sup> Although the highest node in (47) is labeled as an andP, the standard assumption is that the conjunction of two XPs has the status of an XP. Since the example at hand involves conjunction of two CPs, I will treat the andP as a CP in my analysis.



(49) The local context of the left conjunct “*John is over thirty*”

a. Mapping parsed expressions to variables

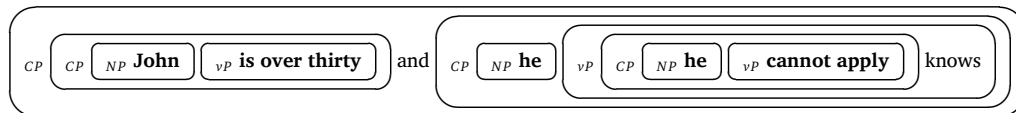
| $a$         | $d$                 | $b$         | $b'$                         |
|-------------|---------------------|-------------|------------------------------|
| $\emptyset$ | John is over thirty | $\emptyset$ | and he he cannot apply knows |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} (c' \text{ and } d') \leftrightarrow d'$$

After calculating the local context of the left conjunct, the interpreter resumes parsing. She processes the expressions in the right conjunct one by one. She does not have access to the semantic value of the embedded clause when she has just finished parsing the embedded predicate *apply*; no round box contains just the parsed expressions in (50).<sup>15</sup> Therefore, she waits until a full designated domain has been parsed.

(50) The interpreter sequentially parses *and*, *he*, *he*, *cannot*, and *apply*



She further parses the matrix predicate *knows*, and only after then she has a newly parsed designated domain in her workspace. There are two maximal projections that are worthy of closer inspection: the right CP conjunct and the embedded CP of the right conjunct. Regarding the former, the interpreter takes into account all of the parsed expressions, modulo the target CP. No expression follows the target CP, and the interpreter introduces the equivalence in (52b). It contains the information that John is over thirty and this proposition will be conjoined with the denotation of the target CP, so the strongest yet innocuous restriction is the set of worlds in which John is over thirty. Thus, the

<sup>15</sup> A question arises as to whether the local context of the matrix subject of the right conjunct (i.e., *he*) encodes the information about the knowledge that John cannot apply. The proposed analysis predicts that this is the case, because only after the interpreter reaches the end of the sentence does she get to calculate the local context of the matrix subject. I believe that this is the wrong prediction, and a quick fix would be to hypothesize that already-parsed CPs are taken as given. In other words, if the interpreter has already parsed one or more CPs and currently aims to calculate the local context of an expression *E* that follows those CPs, she tries to build a designated domain only with the expressions that follow those CPs. Technically, we can revise the proposed algorithm as follows:

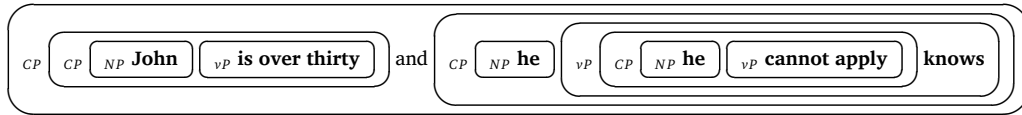
- (i) The local context of a maximal projection  $d$  which occurs in a syntactic environment  $a' a \_ b b'$  (where  $a'$  is a sequence of 0 or more CPs and  $a d b$  constitute the smallest designated domain that contains  $a d$ ) in a context  $C$  is the strongest proposition or property  $x$  which guarantees that for any expression  $d'$  of the same type as  $d$ ,

$$C \models^{c' \rightarrow x} a' a (c' \text{ and } d') b \leftrightarrow a' a d' b$$

The innovation is the introduction of the  $a'$  variable, which denotes the already-parsed CPs. Applying this idea to (46), assuming that the conjunction *and* does not affect the phasal status of a CP,  $a'$  maps to “*John is over thirty and*”,  $a$  to  $\emptyset$ ,  $d$  to *he*, and  $b$  to  $\emptyset$ , and  $b'$  to “*he cannot apply knows*”. The strongest yet innocuous restriction for the NP “*he*” will then only contain the information that John is over thirty.

local context of the right conjunct is the global context intersected with the denotation of the left conjunct. This exemplifies the way in which coordinated CPs are evaluated incrementally.

(51) The interpreter parses *knows*



(52) The local context of the right conjunct “*he knows he cannot apply*”

a. Mapping parsed expressions to variables

| <i>a</i>                | <i>d</i>           | <i>b</i> | <i>b'</i> |
|-------------------------|--------------------|----------|-----------|
| John is over thirty and | he he cannot apply | knows    | ∅ ∅       |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{John is over thirty and } (c' \text{ and } d') \leftrightarrow \text{John is over thirty and } d'$$

Concerning the local context of the embedded CP “*he cannot apply*”, the expressions that precede the target CP is “*John is over thirty and he*”. The predicate *knows* follows the target CP, and no expression is yet to be parsed. The mapping of the variables in (53a) yields the equivalence in (53b), which crucially contains the information that the sentence conveys some information about John’s knowledge and that John is over thirty.

(53) The local context of the embedded clause of the right conjunct “*he cannot apply*”

a. Mapping parsed expressions to variables

| <i>a</i>                   | <i>d</i>        | <i>b</i> | <i>b'</i> |
|----------------------------|-----------------|----------|-----------|
| John is over thirty and he | he cannot apply | knows    | ∅         |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{John is over thirty and he } (c' \text{ and } d') \text{ knows} \\ \leftrightarrow \text{John is over thirty and he } d' \text{ knows}$$

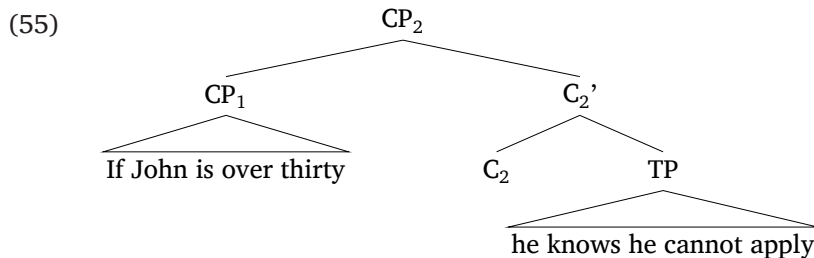
The general prediction on sentential coordination is that cross-linguistic structural variation is orthogonal to the order of context update: since the interpreter invariably parses a given sentence from left to right and calculates the local context of the left conjunct before parsing the right conjunct, it does not matter whether the left conjunct asymmetrically c-commands the right conjunct or vice versa.

## 4.6 Conditionals

Despite the revisions, the proposed algorithm makes the same prediction as Schlenker’s strictly incremental algorithm regarding conditionals. I will make the standard assumption due to Bhatt & Pancheva (2006) that the *if*-clause is adjoined to the consequent clause. This implies that the conditional morpheme and the antecedent constitute a CP, which is a designated domain.

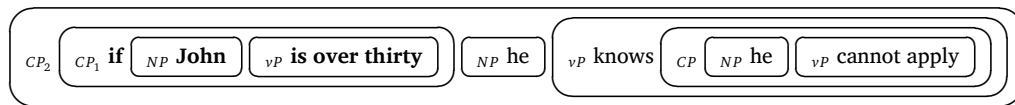
Let us consider the conditional example in (54) and the corresponding syntax in (55).

(54) If John is over thirty, he knows that he cannot apply.



The interpreter first parses the *if*-clause, which is a CP and corresponds to a designated domain.<sup>16</sup> The local context of the *if*-clause will be calculated based on what has been parsed. The corresponding mapping of variables and equivalence are provided in (57a) and (57b), respectively. Since no parsed expression precedes or follows the *if*-clause, the interpreter does not consider any parsed expression and the strongest yet innocuous restriction is the global context.

(56) The interpreter parses the *if*-clause



(57) The local context of the *if*-clause (CP<sub>1</sub>)

a. Mapping parsed expressions to variables

| $a$         | $d$                    | $b$         | $b'$        |
|-------------|------------------------|-------------|-------------|
| $\emptyset$ | if John is over thirty | $\emptyset$ | $\emptyset$ |

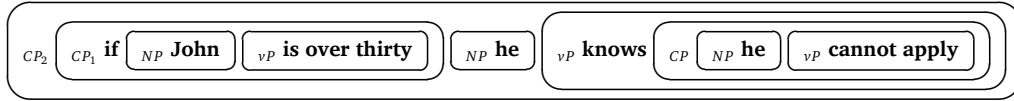
b. Corresponding equivalence

$$C \models^{c' \rightarrow x} (c' \text{ and } d') \leftrightarrow d'$$

The interpreter then parses the consequent clause, after which she has access to the semantic value of the entire conditional construction. I will consider two maximal projections that would be of interest: (i) the entire conditional (CP<sub>2</sub>) and the matrix TP of the consequent clause. Regarding the former, no parsed expression precedes or follows CP<sub>2</sub>; note that CP<sub>2</sub> dominates the *if*-clause (CP<sub>1</sub>). It follows that the interpreter does not consider any parsed expression in the calculation of the local context of CP<sub>2</sub>, and as a result the local context of CP<sub>2</sub> is the global context. This is expected, since the local context of the entire conditional construction is intuitively the global context.

<sup>16</sup> A question arises as to whether the local context of *John* should be calculated before the interpreter parses *is over thirty*. The current proposal predicts that the interpreter has to wait until she parses the entire *if*-clause, for “*if John*” does not constitute a designated domain. I am not aware of any work that discusses local contexts of sub-clausal expressions within an *if*-clause, so I leave it as future research whether the theory requires further refinement.

(58) The interpreter parses the consequent



The local context of “*he knows he cannot apply*” can be calculated by targeting the matrix TP of the consequent rather than the entire  $CP_2$ . This should not be taken as a construction-specific stipulation since Adjustment 2 submits that every maximal projection is a target of local context computation. Given the standard syntax of conditionals in (55), the TP does not dominate the *if*-clause. Consequently, the *if*-clause is not part of the variable  $d$  and the  $a$  variable maps to the *if*-clause. No parsed expression follows the TP so  $b$  and  $b'$  map to an empty string. The interpreter postulates the equivalence in (59b) based on this mapping, which correctly captures the intuition that the she should take into account the contribution of the antecedent clause.

(59) The local context of the matrix TP of the consequent

a. Mapping parsed expressions to variables

| $a$                    | $d$                      | $b$         | $b'$        |
|------------------------|--------------------------|-------------|-------------|
| if John is over thirty | he knows he cannot apply | $\emptyset$ | $\emptyset$ |

b. Corresponding equivalence

$$C \models^{c' \rightarrow x} \text{if John is over thirty } (c' \text{ and } d') \leftrightarrow \text{if John is over thirty } d'$$

I will not fully flesh out the analyses as the equivalences introduced in this section are identical to those provided in Schlenker (2009). It suffices to prove that the revised incremental algorithm makes the same prediction as the original version.

## 5 Symmetric local contexts and domain-by-domain evaluation

The left-to-right bias in parsing-based theories such as Schlenker (2009) is only a bias, and *symmetric local contexts* should be available to the interpreter when needed. That is, the interpreter has the option to consider expressions that follow the target expression.

(60) Local context (symmetric version)

The symmetric local context of a propositional or predicative expression  $d$  that occurs in a syntactic environment  $a \_ b$  in a context  $C$  is the strongest proposition or property  $x$  which guarantees that for any expression  $d'$  of the same type as  $d$ , if  $c'$  denotes  $x$ , then

$$C \models^{c' \rightarrow x} a (c' \text{ and } d') b \leftrightarrow a d' b$$

Barbara Partee offers a convincing case.<sup>17</sup> In (61), the sentence as a whole does not have a presupposition but the left disjunct contains a definite description which should presuppose the

<sup>17</sup> The example initially appeared in Roberts (1989) as Partee (p.c.).

existence of a bathroom. A reasonable explanation is that the presupposition of the left disjunct is filtered due to the information provided by the right disjunct. In technical terms, the interpreter considers the symmetric local context of the left disjunct.

(61) Either the bathroom is well hidden or there is no bathroom.

A possible objection to the present proposal provided by an anonymous reviewer is that the same strategy could apply to belief reports in head-final languages illustrated in (13). Specifically, there might be no need to endorse domain-by-domain evaluation because the interpreter has a good reason to delay local context computation until she parses the preceding attitude verb *mit* ‘believe’. For this could be just one case in which the interpreter resorts to the symmetric algorithm, as she would find it pathological to run the strictly incremental algorithm and repetitively compute the weakest restriction that the embedding predicate can impose.

However, there are some advantages for endorsing domain-by-domain evaluation. First, it receives independent motivation from the syntax-semantics interface and is not in need of a separate explanation of why the interpreter has to consider symmetric local contexts for belief reports in head-final languages. Moreover, it plays a more general role, informing the interpreter how far she needs to parse before calculating the local contexts of nominal modifiers, partitives, and relative clauses. For example, to compute the local context of *Hasidic* in the DP “*a Hasidic Jew*”, the interpreter needs to wait until she has parsed the head noun because it marks the end of a designated domain. The aforementioned alternative explanation offers no provisions for this.

## 6 Further issues

### 6.1 Scrambling

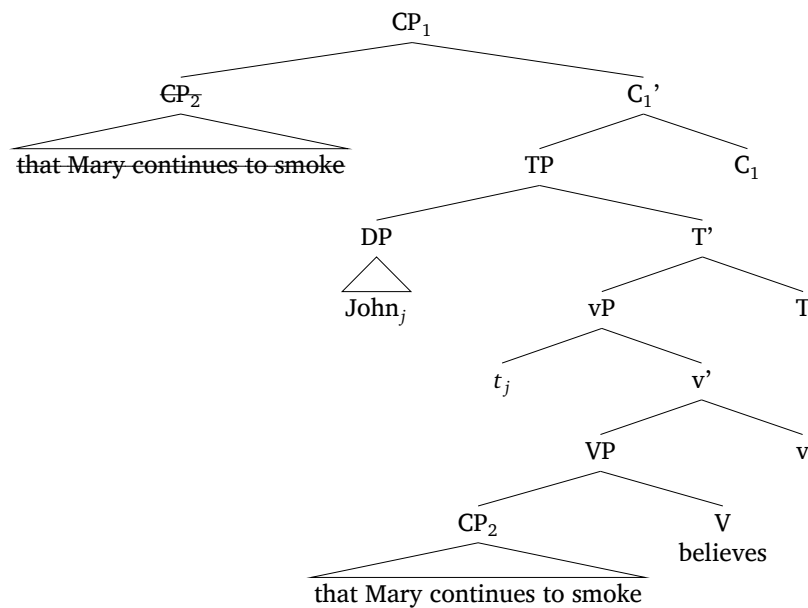
Scrambling is a phenomenon where a syntactic constituent optionally undergoes movement and yields a non-canonical word order. As an example, the embedded clause in (62) moved to the sentence-initial position.

(62) [Mary-ka keysokhayse tambay-lul pi-n-tako]<sub>i</sub> John-i t<sub>i</sub>  
 Mary-NOM continuously cigarette-ACC smoke-PRES-COMP John-NOM  
 mit-nun-ta-nun kes-i sasil-i-ni?  
 believe-PRES-DECL-ADN thing-NOM truth-COP-Q  
 ‘Is it true that Mary continues to smoke, John believes?’

This non-canonical word order poses a problem for Schlenker’s (2009) incremental algorithm as the interpreter would not have access to any expression at all when calculating the local context of the fronted embedded clause. The algorithm wrongly predicts that its local context is the global context. Not to mention that scrambling is an optional process, which makes it even more difficult to posit a construction-specific stipulation.

Do linear order-based theories have no way out? Let me first point out that LF-based theories such as Romoli & Mandelkern (2017) do not suffer from this problem due to their ability to make use of the correspondence of the LFs of scrambled sentences with their unscrambled counterparts via operations like reconstruction, as illustrated in (63).<sup>18</sup> I speculate that a possible solution could be developed along this line of thought, that is, run the revised incremental algorithm on the unscrambled equivalent with which the sentence is identified via reconstruction. However, this is by no means a trivial solution, as it potentially weakens the role of processing which justifies the left-to-right bias. I leave further development of the solution for future study.

(63) Scrambled embedded clause reconstructs



## 6.2 Other attitude reports

As a parsing-based theory that makes predictions about local contexts based on parsed materials, the present proposal cannot account for cases where projection behavior depends on lexical

<sup>18</sup> Let me also note that there is a natural explanation of this phenomenon in a continuation-based approach (Shan & Barker 2006; Barker 2009; Barker & Shan 2014) if the order of evaluation mirrors the order in which local contexts of expressions are calculated. Consider the following continuation-based version of domain-by-domain evaluation: the interpreter only gets to run Schlenker's (2009) incremental algorithm upon evaluation of a clause (as opposed to phase Spell-Out in the original version), and clauses yet to be evaluated are not semantically visible to the interpreter. Now, suppose Barker's (2009) analysis of reconstruction effects in terms of delayed evaluation is on the right track, that is, a displaced string of expressions is evaluated after its preceding expressions. Given these assumptions, the displaced materials (e.g., scrambled embedded clause) will be semantically invisible to the interpreter when she calculates the local context of the preceding expressions (e.g., matrix clause containing a gap of the scrambled embedded clause), because at this point the displaced materials are yet to be evaluated. As a result, scrambling will not have any effect on local context computation.

idiosyncrasy. Blumberg & Goldstein (2023) present such cases, particularly involving attitude verbs other than *believe*. Consider the following example (Blumberg & Goldstein’s (16)) which involves a fictive *imagine*. The current proposal predicts that the local context of the embedded clause is the set of Ann’s imagination worlds, and that the presupposition of the entire sentence is ‘Ann *imagined* that Bill used to smoke’. However, what we actually observe is belief projection, i.e., ‘Ann *believed* that Bill used to smoke’. Since we cannot rule out the possibility that Ann’s belief worlds and her imagination worlds are disjoint, the prediction is not borne out.

(64) Ann imagined that Bill stopped smoking.

The problem extends to cases involving other attitude verbs such as *want*, *wish*, *hope*, and *dream* as these verbs all exhibit belief projection. The source of the trouble is that in parsing-based theories, the local context of the embedded clause is a function of the semantics of the embedding attitude verb but the majority of attitude verbs do not directly encode the semantics of *believe*. As Blumberg & Goldstein point out, no extant parsing-based theory can account for belief projection in full generality. The present proposal is no exception.

## 7 Conclusion

This paper introduces a hybrid theory of local contexts which takes into account both linear order and syntax. In light of the advances in syntax-semantics interface, I proposed that local context computation, which concerns semantic entailment, should only be performed when syntax ships semantic information to the CI interface for LF interpretation. This implies that local context is calculated domain by domain, rather than in a strictly incremental fashion. In addition, I proposed that local context should be calculated for each maximal projection as opposed to each lexical item. The two innovations jointly explained away problems that plagued existing parsing-based theories exclusively relying on either linear order or c-command—most notably nominal modification and partitive construction. In addition, the proposed theory offered a principled account of data drawn from head-final languages such as Korean, where belief reports and coordination pose a problem for the strictly incremental theory and the c-command-based alternative, respectively.

---

## Abbreviations

ACC = accusative, ADN = adnominal, COMP = complementizer, COP = copula, DECL = declarative, DAT = dative, NOM = nominative, Q = question particle, PAST = past, PERF = perfect, PRES = present

## Acknowledgements

I gratefully acknowledge comments and suggestions from Amir Anvari, Chris Barker, Emmanuel Chemla, Paul Egre, Maria Esipova, Janek Guerrini, Semoon Hoe, Heejeong Ko, Jeremy Kuhn, Salvador Mascarenhas, Daniel Plesniak, Jacopo Romoli, Benjamin Spector, Philippe Schlenker, and three anonymous referees.

## Funding information

This work was supported by the New Faculty Startup Fund from Seoul National University.

## Competing interests

The author has no competing interests to declare.

---

## References

- Anvari, Amir. 2018. A problem for Maximize Presupposition! (locally). *Snippets* 33. 1–2. DOI: <https://doi.org/10.7358/snip-2018-033-anva>
- Anvari, Amir & Blumberg, Kyle. 2021. Subclausal local contexts. *Journal of Semantics* 38(3). 393–414. DOI: <https://doi.org/10.1093/jos/ffab004>
- Barker, Chris. 2009. Reconstruction as delayed evaluation. In Hinrichs, Erhard W. & Nerbonne, John A. (eds.), *Theory and evidence in semantics*, 1–28. CSLI Publications.
- Barker, Chris. 2022. Composing local contexts. *Journal of Semantics* 39(2). 385–407. DOI: <https://doi.org/10.1093/jos/ffac003>
- Barker, Chris & Shan, Chung-chieh. 2014. *Continuations and natural language*. Oxford University Press. DOI: <https://doi.org/10.1093/acprof:oso/9780199575015.001.0001>
- Bhatt, Rajesh & Pancheva, Roumyana. 2006. Conditionals. In Everaert, Martin & van Riemsdijk, Henk (eds.), *The Blackwell Companion to Syntax*, 638–687. Blackwell Publishing. DOI: <https://doi.org/10.1002/9781118358733.wbsyncom119>
- Blumberg, Kyle & Goldstein, Simon. 2023. Attitude verbs' local context. *Linguistics and Philosophy* 46. 483–507. DOI: <https://doi.org/10.1007/s10988-022-09373-y>
- Bošković, Željko. 2014. Now I'm a phase, now I'm not a phase: On the variability of phases with extraction and ellipsis. *Linguistic inquiry* 45(1). 27–89. DOI: [https://doi.org/10.1162/LING\\_a\\_00148](https://doi.org/10.1162/LING_a_00148)



- Bošković, Željko. 2016. What is sent to spell-out is phases, not phasal complements. *Linguistica* 56(1). 25–66. DOI: <https://doi.org/10.4312/linguistica.56.1.25-66>
- Chemla, Emmanuel. 2008. An epistemic step for anti-presuppositions. *Journal of Semantics* 25(2). 141–173. DOI: <https://doi.org/10.1093/jos/ffm017>
- Chierchia, Gennaro. 2009. On the explanatory power of dynamic semantics. Handout from talk at Sinn und Bedeutung 14.
- Chomsky, Noam. 1995. Bare phrase structure. In Webelhuth, Gert (ed.), *Government and Binding Theory and the Minimalist Program*, 383–439. Oxford: Blackwell Publishers.
- Chomsky, Noam. 2000. Minimalist inquiries. In Martin, Roger & Michaels, David & Uriagereka, Juan (eds.), *Step by step: Essays on minimalist Syntax in honor of Howard Lasnik*, 89–155. MIT Press.
- Chomsky, Noam. 2001. Derivation by phase. In Kenstowicz, Michael (ed.), *Ken Hale: A life in language*, 1–52. MIT Press. DOI: <https://doi.org/10.7551/mitpress/4056.003.0004>
- Chomsky, Noam. 2008. On phases. In Freidin, Robert & Otero, Carlos P. & Zubizarreta, Maria Luisa (eds.), *Foundational issues in linguistic theory: Essays in honor of Jean-Roger Vergnaud*, 132–166. MIT press. DOI: <https://doi.org/10.7551/mitpress/9780262062787.003.0007>
- Chomsky, Noam. 2013. Problems of projection. *Lingua* 130. 33–49. DOI: <https://doi.org/10.1016/j.lingua.2012.12.003>
- Chomsky, Noam. 2015. Problems of projection: Extensions. In Demonico, Elisa & Hamann, Cornelia & Matteini, Simona (eds.), *Structures, Strategies and Beyond: Studies in honour of Adriana Belletti*, 1–16. John Benjamins. DOI: <https://doi.org/10.1075/la.223.01cho>
- Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of the Second West Coast Conference on Formal Linguistics*, 114–125. Stanford: CSLI Publications.
- Heim, Irene. 1990. Presupposition projection. In van der Sandt, Rob (ed.), *Reader for the Nijmegen workshop on presupposition, lexical meaning, and discourse processes*. University of Nijmegen.
- Ingason, Anton Karl. 2016. Context updates are hierarchical. *Glossa: a journal of general linguistics* 1(1). 1–9. DOI: <https://doi.org/10.5334/gjgl.71>
- Kaplan, David. 1989. Demonstratives. In Almog, Joseph & Perry, John & Wettstein, Howard (eds.), *Themes from Kaplan*, Oxford University Press.
- Karttunen, Lauri. 1974. Presupposition and linguistic context. *Theoretical Linguistics* 1(1–3). 181–194. DOI: <https://doi.org/10.1515/thli.1974.1.1-3.181>
- Katzir, Roni & Singh, Raj. 2014. Hurford disjunctions: Embedded exhaustification and structural economy. In Etcheberria, Urtzi & Fălăuș, Anamaria & Irurtzun, Aritz & Leferman, Bryan (eds.), *Proceedings of Sinn und Bedeutung 18*, 201–216.
- Matushansky, Ora. 2005. Going through a phase. In *Perspectives on phases*, 157–181.
- Mayr, Clemens & Romoli, Jacopo. 2016. A puzzle for theories of redundancy: Exhaustification, incrementality, and the notion of local context. *Semantics and Pragmatics* 9(7). 1–48. DOI: <https://doi.org/10.3765/sp.9.7>

- Percus, Orin. 2006. Antipresuppositions. In Ayumi, Ueyama (ed.), *Theoretical and empirical studies of reference and anaphora: Toward the establishment of generative grammar as an empirical science*, 52–73.
- Phillips, Colin. 1996. *Order and structure*: Massachusetts Institute of Technology dissertation.
- Roberts, Craige. 1989. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy* 12(6). 683–721. DOI: <https://doi.org/10.1007/BF00632602>
- Romoli, Jacopo & Mandelkern, Matthew. 2017. Hierarchical Structure and Local Contexts. In *Proceedings of Sinn und Bedeutung 21*, 1017–1034.
- Sauerland, Uli. 2008. Implicated presuppositions. In Steube, Anita (ed.), *The discourse potential of underspecified structures*, 581–600. Mouton de Gruyter. DOI: <https://doi.org/10.1515/9783110209303.4.581>
- Schlenker, Philippe. 2009. Local contexts. *Semantics and Pragmatics* 2(3). 1–78. DOI: <https://doi.org/10.3765/sp.2.3>
- Schlenker, Philippe. 2010. Presuppositions and local contexts. *Mind* 119(474). 377–391. DOI: <https://doi.org/10.1093/mind/fzq032>
- Schlenker, Philippe. 2011a. Presupposition projection: Two theories of local contexts – Part I. *Language and Linguistics Compass* 5(12). 848–857. DOI: <https://doi.org/10.1111/j.1749-818X.2011.00299.x>
- Schlenker, Philippe. 2011b. Presupposition projection: Two theories of local contexts -Part II. *Language and Linguistics Compass* 5(12). 858–879. DOI: <https://doi.org/10.1111/j.1749-818X.2011.00300.x>
- Schlenker, Philippe. 2020. Inside out: A note on the hierarchical update of nominal modifiers. *Glossa: a journal of general linguistics* 5(1). DOI: <https://doi.org/10.5334/gjgl.1130>
- Schwarzschild, Roger. 2006. The role of dimensions in the syntax of noun phrases. *Syntax* 9(1). 67–110. DOI: <https://doi.org/10.1111/j.1467-9612.2006.00083.x>
- Shan, Chung-chieh & Barker, Chris. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29. 91–134. DOI: <https://doi.org/10.1007/s10988-005-6580-7>
- Singh, Raj. 2011. Maximize Presupposition! and local contexts. *Natural Language Semantics* 19(2). 149–168. DOI: <https://doi.org/10.1007/s11050-010-9066-2>
- Soames, Scott. 1982. How presuppositions are inherited: A solution to the projection problem. *Linguistic inquiry* 13(3). 483–545.
- Soames, Scott. 1989. Presupposition. In *Handbook of philosophical logic*, 553–616. Springer. DOI: [https://doi.org/10.1007/978-94-009-1171-0\\_9](https://doi.org/10.1007/978-94-009-1171-0_9)
- Stalnaker, Robert. 1974. Pragmatic presuppositions. In Stalnaker, Robert (ed.), *Context and content*, 47–62. Oxford University Press. DOI: <https://doi.org/10.1093/0198237073.003.0003>

