Parameters of cross-linguistic variation in expectation-based Minimalist Grammars (e-MGs)

Cristiano Chesi*
NeTS-IUSS Pavia

The fact that Parsing and Generation share the same grammatical knowledge is often considered the null hypothesis (Momma and Phillips 2018) but very few algorithms can take advantage of a cognitively plausible incremental procedure that operates roughly in the way words are produced and understood in real time. This is especially difficult if we consider cross-linguistic variation that has a clear impact on word order. In this paper, I present one such formalism, dubbed Expectation-based Minimalist Grammar (e-MG), that qualifies as a simplified version of the (Conflated) Minimalist Grammars, (C)MGs (Stabler 1997, 2011, 2013), and Phase-based Minimalist Grammars, PMGs (Chesi 2005, 2007; Stabler 2011). The crucial simplification consists of driving structure building only using lexically encoded categorial top-down expectations. The commitment on the top-down procedure (in e-MGs and PMGs, as opposed to (C)MGs, Chomsky 1995; Stabler 2011) will be crucial to capture a relevant set of empirical asymmetries in a parameterized cross-linguistic perspective which represents the least common denominator of structure building in both Parsing and Generation.

1. Introduction

From the psycholinguistic perspective, it is often considered the null hypothesis to assume that Comprehension and Production share the same grammatical knowledge and a relevant part of the dynamic procedure needed to produce and comprehend sentences "on-line", that is, in real time: both speakers and listener are pro-active in postulating "actions" (speech acts) based on "forward" predictive models (Pickering and Garrod 2013). In computational terms, this amounts to saying that Parsing and Generation share not only the same grammatical declarative knowledge (e.g. rewriting rules in Chomsky's, 1957 sense) but also a significant part of the derivational procedure that allows an ideal hearer to parse a sentence and an ideal speaker to generate it. In concrete terms, from this perspective, one must assume that, given, for instance, a Context-Free Grammar, some subroutines like Prediction, Scanning, and Completion, as expressed by the Earley parsing algorithm (Earley 1970), should be used both to parse and to generate a sentence. Changing grammatical formalism does not undermine this core assumption: there should be a relevant procedural least common denominator that is shared in any processing task and that fits on-line processing. Defining the dimension(s) of this least common denominator is both a formal and an empirical problem. In this paper, I will concentrate on this core part both from the compu-

* My special thanks go to Greg Kobele and the ESSLLI 2022 students who discussed with me most of the ideas here presented.

© 2015 Associazione Italiana di Linguistica Computazionale

tational and the algorithmic perspectives (in Marr's sense, Marr 1982), by assuming the simplest possible lexicalist approach and the most minimal procedure to ensure a core incremental derivation that is the same in parsing and generation, being the only relevant difference associated with the information asymmetry which is available in the two tasks. Here, I will adopt the Minimalism perspective (Chomsky 1995, 2001), which is an elegant transformational grammatical framework that defines structural dependencies in phrasal (i.e. hierarchical) terms simply relying on one core structure building operation, Merge, that combines lexical items and the result of other Merge operations. In mainstream minimalism, Merge is a binary, bottom-up operation that takes two items, either lexical (e.g. $reads$ and $comics$) or result to other Merge operations (e.g. $[the\ books]$), and creates the set formed by the two, linearly unordered, items (i.e. $[reads\ comics]$ or $[reads\ [the\ books]]$, which are equivalent to $[comics\ reads]$ or $[[the\ books]\ reads]$). Under this perspective, (1a), below, is the representative result of two Merge operations (i.e. Merge($John$, Merge($reads$, $comics$)) both taking the items $John$, $reads$, and $comics$ directly from the lexicon (let us ignore for the moment morphological decomposition, cf. Kobele to appear), while (1b) relies on the so-called Internal Merge (Move): the re-Merge of an item ($comics$) that was already merged in the structure (as in (1a)), inducing a focalization construction.

(1) a. $[\ John\ [\ reads\ comics\ ]\ ]$ Merge
b. $[\ COMICS\ [\ John\ [\ reads\ \_{comics}\ ]\ ]\ (not\ books!)$ Merge + Move

As result, Move connects the item at the edge of the structure ($COMICS$, focalized in this case, as indicated by capital letters) with its "trace" ($\_{comics}$), a phonetically empty copy of the item that in a previous Merge operation combined with a hierarchically lower item ($reads$ in (1b)). In both (Conflated) Minimalist and Phase-based Minimalist Grammars ((C)MGs and PMGs respectively, Stabler 2011, 2013 Merge and Move are feature-driven operations, that is, a successful operation must be triggered by the relevant (categorial) features matching, and, once these features are used, they get deleted. Consequently, a feature pair is always responsible for each operation (unless specific features are left unerased after a successful operation, as in raising predicates and successive cyclic movement, Stabler 2011). One crucial difference between PMGs (Chesi 2007, 2012) and MGs is that while MGs operate from-bottom-to-top, (2), PMGs structure building operations apply top-down, (3)[1]:

(2) MERGE ($\alpha_{=X}$, ${}_{X}\beta$) = [α [α $\cancel{=X}$ $\cancel{X}$ β]] MGs
$Move$ (α ${}_{+Y}$ [... ${}_{-Y}$ β ...]) = [α [$\cancel{-Y}$ β [α $\cancel{+Y}$ [... (${}_{-Y}$ β) ...]]]]

(3) MERGE ($\alpha_{=X}$, ${}_{X}\beta$) = [α $\cancel{=X}$ [$\cancel{X}$ β]] PMGs
$Move$ (α ${}_{+Y}$ ${}_{=S}$ [${}_{Y}$ β]) = [α $\cancel{+Y}$ [${}_{Y}$ β] ${}_{=S}$ ${}_{S}$[... $\cancel{=Y}$ [$\cancel{Y}$ (β)] ...]]

The major differences can be illustrated with two examples: as long as the Merge operation is considered, the same featural decoration of the lexical items $[reads_{=D}]$ and

1 α and β are lexical items, $= X$ indicates the selection of X, where X is a categorial feature (e.g. $N(oun)$ or $V(erb)$ or Aux). Lexical items are tuples consisting of selections/expectations ($= X$) and categories (X, i.e. selected/expected features); for convenience, select features are expressed by rightward subscripts ($\alpha_{=X}$), and categories as leftward subscripts (${}_{X}\alpha$). Similarly, Move in MGs is driven by licensing (${}_{-Y}\alpha$, leftward subscripts) and licensors (α_{+Y}, rightward subscripts) features (Stabler 2011).

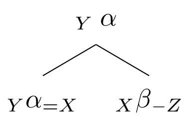
$[_D\ comics]$ would produce $[reads\ [reads_{\cancel{=D}}\ _{\cancel{D}}comics]]$ according to (2), and $[reads_{\cancel{=D}}$ $[_{\cancel{D}}comics]]$ under (3). The difference is more radical for Move: (1b) can be derived by adding a feature $+Foc$ (MGs) or $+D$ (PMGs) to the lexical item associated with movement (a, phonetically empty complementizer C, or a $Focus$ head, in the split-CP area Rizzi 1997, selecting the finite predicate $reads$) and $-Foc$ (necessary in MGs only) to $comics$; according to (2), $+Foc$ and $-Foc$ will be the last features evaluated in the derivation (i.e. $[[_{\cancel{-Foc}}comics]\ _{\cancel{+Foc}}$ ... $reads$ ...]). Following (3), the prediction is the opposite: the $+D$ / D feature pair is evaluated first, triggering a Merge operation forming $[[_D\ comics]\ _{\cancel{+D}}$ ... $reads$ ... $_{=D}]$ first, then "bookkeeping" the fact that the D feature of $[_D\ comics]$ remains unselected and must be licensed later (possibly, but not necessarily, by the $[reads..._{=D}]$ item, when merged).

All in all, under the definitions in (3), Merge generates hierarchically asymmetric structures ((3) [α [β]] vs. (2) [α β]), and Move (i) "stores" an unselected item ($_{\cancel{+Y}}[_Y\ \beta]$) and (ii) re-Merges it later in the structure as soon as the relevant categorial selection is introduced (i.e. =Y). Notice that MGs use the "+/-" feature distinction and the same deletion procedure after matching, while PMGs do not use "-" features and simply assume that both "+" and "=" select categorial features, but only those selected by "=" are deleted after Merge. In PMGs, both "+" features and partial selection (e.g. $_{\cancel{=X}}[_{\cancel{X}}\ _Y$ $\beta]$) force "memory storage", that is, the item partially selected or selected through "+" is flagged and remains prominent ("pending in memory") for next Merge operations. This implements the "movement" of the flagged item, which is available to re-merge before any other input token, until the relevant prominent category identifying the moved item (Y in (3)) is selected. If no appropriate select feature is found later in the derivation for any "pending" item, the sentence is ungrammatical. CMG as well dispenses the grammar with the +/- feature distinction and only relies on select features (=X), but it must assume that feature deletion can be procrastinated (again, for instance, in raising predicates). From a generative point of view, all these formalisms are equivalent and they all fall under the so-called mildly-context sensitive domain (Stabler 2011). It is however worth appreciating the dynamics of structure building "online", namely how the derivation unrolls (algorithmically speaking) word by word: Taking the MGs lexicon (4), the expected constituents in (1) are built adding items to the left-edge of the structure at each Merge/Move application, as described in (5).

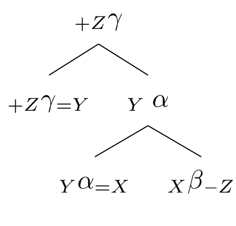
(4) Lex_{MG} = $[\ _Y\ \alpha\ _{=X}\ ]$, $[\ _{X-Z}\ \beta\ ]$, $[\ \gamma\ _{=Y+Z}\ ]$

(5)

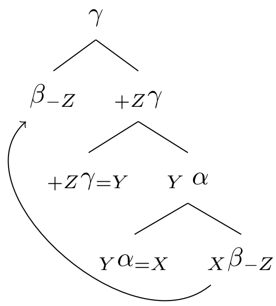
i. MERGE ($_Y\alpha_{=X}$, $_X\beta_{-Z}$) =



ii. MERGE ($_{+Z}\gamma_{=Y}$, $_Y\alpha$ [...]) =



iii. MOVE ($_{+Z}\gamma$ [α [$\alpha\ \beta_{-Z}$]]) =

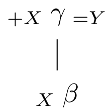


An equivalent structure is obtained in PMGs[2] as shown in (7). Notice a minimal difference in the lexicon (6), namely the absence of the "-" features.

(6) Lex_{PMG} = [$_Y\ \alpha\ _{=X}$], [$_X\ \beta$], [$\gamma\ _{+X=Y}$]

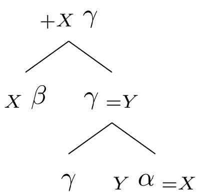
(7)

i. MERGE ($_X\ \beta$, $_{+X}\ \gamma\ _{=Y}$) =



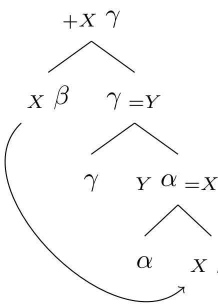
$_X\ \beta \rightarrow$ M

ii. MERGE ([[$_X\ \beta$] $\gamma\ _{=Y}$], $_Y\ \alpha\ _{=X}$) =



M = { $_X\ \beta$ }

iii. MOVE ([[$_X\beta$] γ [(γ) [$\alpha\ _{=X}$]]], M = { $_X\beta$ }) =



$_X\ \beta \leftarrow$ M

The result of the two derivations is (strongly) equivalent in hierarchical (and dependency) terms. The simplicity, in pretheoretical terms, of the two descriptions is

2 Move is implemented using a Last-In-First-Out addressable memory buffer M, where the item (β) with (at least) one unselected category (X) is stored ("$_X\beta \rightarrow M$") and retrieved ("$_X\beta \leftarrow M$") when selected (i.e. "$= X$").

comparable: while PMGs must postulate the M storage to implement Move (as result of the missing selection of a categorial feature), MGs must postulate an "independent workspace" (Nunes and Uriagereka 2000) to build nontrivial left-branching structures: for instance, instead of having a single-word subject like $John$, in (1a), a multi-word subject like $the\ boy$ must be created (by merging the with boy) before it can be merged with the relevant predicate phrase (i.e. $[reads\ comics]$); if this is not the case and $[reads\ comics]$ merges with the alone, any selecting feature of the would remain unsatisfied, since the only element accessible to further Merge operations would be the one selecting, namely $reads$, which remains at the top of the tree structure created so far. Furthermore, both formalisms must restrict the behavior either of the M buffer operativity or the accessibility to the $-f$ features to limit the Move operation (e.g., island constraints, Huang 1982).

1.1 The superiority of the Top-down perspective

Although from a purely formal perspective, the two approaches both try to define the very same computational domain (i.e. the identification of the exact set of well-formed sentences in a given language with their associated relevant structure(s) feeding compositional semantic evaluation), there are at least three good reasons to commit ourselves to the top-down algorithmic orientation instead of remaining agnostic or relying on the mainstream Minimalist brick-over-brick (from-bottom-to-top) approach (Chesi 2007): First, the order in which the structure is built is grossly transparent with respect to the order in which the words are processed in real-time tasks, both in Generation and in Parsing in PMGs, but not in MGs. Second, in PMGs, the simple processing order of multiple expectations is sufficient to distinguish between sequential (the last expectation of a given lexical item) and nested expectations (any other expectation): The first qualifies as the transparent branch of the tree (i.e. it is able to license pending items from the superordinate selecting item), while constituents licensed by nested expectations qualify as configurational islands (Bianchi and Chesi 2006; Chesi 2015). Moreover, successive cyclic movement is easily described in PMGs without relying on feature checking at any step or non-deterministic assumptions on feature deletion (Chesi 2015) contrary to (C)MGs. A third logical reason to prefer the top-down orientation over the bottom-up alternative is related to the unicity of the root node in tree graphs and it deserves a specific section (§1.2).

1.2 Single Root Condition

A logical reason to prefer the top-down orientation over the bottom-up alternative is related to the unicity of the root node. As anticipated, the creation of complex (binary) branching structures poses a puzzle for (C)MGs: independent workspaces must be postulated, namely [*the boy*] and the [*reads* ...] phrases must be created before the first can merge with the second (phrasal labels such as VP and DP are indicated for convenience):

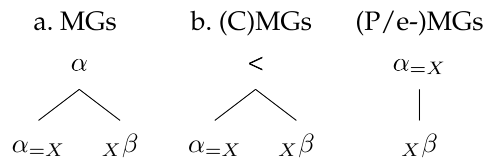
(8) $[_{VP}\ [_{DP}$ the boy$]\ [_{V}$ reads $[_{DP}$ a book$]]$

This is the case for any “complex” subject or adjunct (i.e., non-projecting constituents which are simply composed by multiple words) that must be the result of (at least) one independent Merge operation before this can merge with the relevant predicate (e.g. [$_V$ reads ...][3] in (8). The processing of these constituents represents a major difference between MGs and PMGs derivations. While MGs must decide where to start from (and both solutions are possible and forcefully logically independent from Parsing or Generation, which undeniably proceeds incrementally "from left to right"), PMGs take advantage of the "single root condition" (Partee, Meulen, and Wall 1993, p.439) and avoid this problem:

(9) In every well-formed constituent structure tree, there is exactly one node that dominates every node.

As indicated in (3), the binary operation Merge simply produces a hierarchical dependency in which the dominating (asymmetrically C-commanding, in the sense of Kayne 1994) item, is above the dominated (C-commanded) one. This is compatible with Stabler notation (3a-b) and plainly solves the ambiguity of the nature of the "label" of the constituent (Rizzi 2016). In this sense, PMGs (and e-MGs) can adopt directly a more concise description, that is (3c), totally transparent (see 6.1) with respect to the (Universal) Dependency approach (Nivre et al. 2017).

(10)



The higher node (possibly the root) is always a selecting item (a probe, in minimalist terms), and it is the first item to be processed. This does not necessarily imply that this item is linearized before the selected category (the goal, in minimalist terms): if the selecting node has multiple selections, it must remain at the right edge of the structure to license, locally, the other(s) selection expectation(s). E.g., with [$\alpha_{=X=Y}$], [$_X\beta$], [$_Y\gamma$]:

(11) $[\alpha \cancel{=X} =Y [\cancel{X}\ \beta] [(\alpha \cancel{=Y}) [\cancel{Y}\ \gamma]]]$

In this case, $<\alpha, \beta, \gamma>$ would be the default linearization, but it is easy to derive $<\beta, \alpha, \gamma>$ instead, assuming a simple parameterization on "spell-out" in case of multiple select features (§4).

1.3 The logics behind expectations

I want to conclude this introduction with a discussion of the notion of expectation which goes behind the computational and algorithmic perspective. Much work on psycholinguistics and cognitive literature adopts some notion of expectation (mainly in information-theoretic terms) to explain priming effects and various processing facilitations. In a broad sense, parsing/generating a determiner, for example in a language like Italian *il* or English *the*), casts an expectation on the next word category which is clearly biased in favor of a nominal rather than an auxiliary classification (*the book* vs. **the*

3 Considering the inflection "$-s$" as part of the lexical element or by (head) moving the root "read-" to T is uninfluential here. This sort of head movement will be trivially implemented within the lexical item in e-MGs (e.g. [$_T$ eat-s $\cancel{=V}$ [$\cancel{V}\_\_{eat}$ ...]]).

has) that can be calculated in precise probabilistic terms (e.g. $count(DN)/count(D)$ > $count(DAux)/count(D)$, that is, the probability of finding an Aux after a D is lower than the probability of finding N after D). Things are not always so simple and so "local". For instance, in head-final languages (like Japanese or Turkish), a clear facilitation, measured as a significant reduction in reading times, has been observed in the verb region when specific arguments are processed before the final predicate. This is also known as an "anti-locality" effect (Levy and Keller 2013) and it seems better modeled using probabilistic cues (e.g. training Simple Recurrent Network, Konieczny and Döring 2003) than in terms of structural integration cost (Gibson 1998). Notice, however, that "structural" considerations overtly model the evidence that a gap is never postulated into a (strong/configurational) island domain (Sprouse, Wagers, and Phillips 2012). The descriptive transparency of the structural assumptions, as well as its explanatory power, must then be evaluated against the robustness of purely probabilistic modeling. A perspective partially bridging the gap between a pure probabilistic and a structural perspective is both Roger Levy's relative-entropy-based approach (Levy 2008) and John Hale's surprisal-based approach (Hale 2011), for instance. The difference with respect to the current approach is that both adopt some explicit notion of corpus-based statistical prediction for robust parsing/comprehension, while here, the intent is much more restricted and can be summarized in the following research question: how far we can go if we assume that structure building is only driven by categorial, lexically encoded, expectations? The proposal should then be precise enough to allow one scholar to compare specific assumptions ("parameters" in §4) that are currently debated in generative linguistics and, possibly, to adapt statistical assumptions into the current categorial approach, for instance, to extract, automatically, categorial selection and build a richly decorated lexicon from an annotated (and, maybe, also not annotated) corpus (see discussion in §6) or to compare the e-MG core derivation procedure against Earley parsing algorithm under the surprisal-based approach (Hale 2001).

2. The grammar

As (C/P)MGs, e-MGs include a specification of a lexicon (Lex) and a set of functions (F), the structure building operations. The lexicon, in turn, is a finite set composed of words (or morphemes, in the sense of Kobele to appear) each consisting of phonetic/orthographic information ($Phon$) and a combination of categorical features (Cat), expressing $expect(ations)$, $expected$ and $agreement$ categories[4]. In the end, an optional set of Parameters (P) (see Chesi 2021b), inducing minimal modifications to the structure building operations F and, possibly, to the Cat set, under the fair assumption that F and Cat are universal. More precisely, any e-MG is a 5-tuple such that:

(12) $G = (Phon, Cat, Lex, F, P)$, where
$Phon$, a finite set of phonetic/orthographic features (i.e., orthographic forms representing words pronunciation, e.g., comics = /k oh ′ m i k s/)
Cat, a finite set (morphosyntactic categories, that indicate expectations or express agreement features e.g., "D", "V"... "gen(der)", "num(ber)", "pl(ural)", etc.)
Lex, a set of expressions built from $Phon$ and Cat (the lexicon)

4 As in MGs, lexical items could be specified both for phonetic ($Phon$) and semantic features (Sem). In e-MGs, expectations ($=/+X$) and expectees (X) correspond to MGs $selectors/licensors$ and $selectees/licensees$ respectively. $Agreement$ features indicate categorial values to be unified (Chesi 2021a).

F, a set of partial functions from tuples of expressions to expressions (the structure building operations)
P, a finite set of minimal transformations of F and/or Cat (the parameters), producing F' and Cat', respectively.

F corresponds to functions like the ones expressed in (3) and it is better re-defined in §2.2. P will be introduced in §4, while the structure of the lexical items (Lex) will be exemplified in the next section.

2.1 Lexical items and categories

Each lexical item l in Lex, namely each word (or morpheme), is a 4-tuple defined as follows[5]:

(13) $l = (Ph, Exp(ect), Exp(ect)ed, Agr(ee))$,
Ph, from $Phon$ in G (e.g., "the"; for simplicity, phonetic features are not used)
Exp, a finite list of ordered features from Cat in G (the category/ies that the item expects will follow, e.g., $= N$)
$Exped$ is a finite list of ordered features from Cat in G (the category/ies that should be licensed/expected, e.g., N)
Agr is a structured list of features from Cat in G (e.g., $gen.fem$, $num.pl$)

All $Exp(ect)$, $Exp(ect)ed$ and $Agr(ee)$ features are then subsets of Cat in G. In Agr, for instance, a feminine gender specification ($gen.fem$) expresses a subset relation (i.e., "feminine" $\subseteq$ "gender"). For sake of simplicity, each l will be represented as $[_{Exped(;Agree)} Phon_{=/+Exp}]$ as in (14):

(14) $[_D$ the $_{=N}]$, $[_{N;num.pl}$ dogs$]$, $[_{T;per.3\ num.sg}$ barks $_{=D}]$

This is equivalent to the more standard, but probably less readable format that uses double columns for separating the phonetic items from its ordered feature set (e.g. "the :: D =N", Stabler 2013). We refer to the most prominent (i.e., the first) $Exp(ect)ed$ feature as the Label (L) of the item. E.g., the label L of "the" will be D, while the label of "barks" will be T. Similarly, let us call S (for $select$) the first $Exp(ect)$ feature and R the $remaining\ Exp(ectactions)$ (if any).

2.2 Structure Building operations

Given l_x an arbitrary item such that $l_x = (P_x, L_x/Exped_x, S_x/R_x/Exp_x, Agr_x)$ we can define MERGE as follows:

$$\text{MERGE}(l_{1(S1)}, l_{2(L2)}) = \begin{cases} 1, [l_1(\cancel{S1})\ [l_2(\cancel{L2})]] & \text{if } S_1 = L_2. \\ 0, & \text{otherwise.} \end{cases} \quad (2.2.1)$$

5 This is the simplest possible implementation. Attribute-Value Matrices, as in HPSH (Pollard and Sag 1994) or TRIE/compact trees exploiting the sequence of expectations (Chesi 2018; Stabler 2013) are possible implementations.

MERGE is implemented as the usual binary function that is successful (it returns "1") and creates the dependency (asymmetric C-command or inclusion, in set theoretic terms) (10c), namely $[l_1[l_2]]$, if and only if the label of the subsequent item (l_2) is exactly the one expected by the preceding item (l_1), namely $S_1 = L_2$. This is probably both too strict in one sense (adjuncts are not properly selected) and too permissive in another (certain elements must agree to be merged). In the first case, I assume that $[l_1[l_2]]$ can be formed even if S_1 is not $= X$ but $+X$: while $= X$ corresponds to functional selection (in compositional semantics terms, Heim and Kratzer 1998), $+X$ corresponds to an intersective compositional interpretation (e.g. adjuncts and restrictive relative clauses). As for the agreement constraint, I postulate an extra (possibly parametrized) condition on MERGE, namely the sharing (inclusion) of the relevant Agr features associated with some specific categories. The auxiliary functions necessary to implement Agreement are AGREE and UNIFY and can be minimally defined as follows:

$$\text{AGREE}(l_{1(L1)}, l_{2(L2)}) = \begin{cases} 1, & \text{if } L_1 \wedge L_2 \in P\{Agr\} \implies UNIFY(l_1, l_2). \\ 0, & \text{otherwise.} \end{cases} \tag{2.2.2}$$

$$\text{UNIFY}(l_{1(L1)}, l_{2(L2)}) = \begin{cases} 1, a, \forall a : Agr_1 & \text{if } a \subseteq b. \\ 1, b, \forall b : Agr_2 & \text{if } b \subseteq a. \\ 1, ab, \forall a : Agr_1 \forall b : Agr_2 & \text{if } a \cap a = \varnothing\ . \\ 0, & \text{otherwise.} \end{cases} \tag{2.2.3}$$

Unification is simply expressed as an inclusion relation returning true and the most specific feature for any possible featural intersection between l_1 and $l_2 Agr$ features[6]. Notice that Agreement is a conditional, parametrized option, that is, it only involves specific categories specified in the parameter set P: if the L category belongs to the Agreement set (Agr) in P for the grammar G, unification will be attempted, otherwise agreement will be trivially successful. The fact that $AGREE$ should apply in conjunction with MERGE is straightforward in the $D - N$ domain: in most Romance languages, in which gender and number are shared between the determiner and the noun, we assume that D selects N (this happens also for intermediate functional specifications, according to the cartographic intuition, Cinque 2002). This is less evident in the $Subject$ - $Predicate$ case, in Subject Verb (SV) languages, where the predicate should select (then precede) D. Since the subject is clearly processed (i.e. merged) before T, in canonical SV sentences, and it does not select T, a re-merge operation should be considered (e.g. case checking). This re-merge (inducing local Agree, pace Chomsky 2001) is logically and empirically sound (movement and agreement can be related and parametrized, Alexiadou and Anagnostopoulou 1998). In this case, re-merge must be preceded by MOVE, an operation that stores in memory an item that is "not fully" expected (i.e. there are $exped_2$ features remaining) by the previous MERGE:

6 $UNIFY(num, num.pl)$ = $num.pl$; UNIFY($\varnothing, num.pl$) = $num.pl$; UNIFY($gen.f, num.pl$) = $gen.f, num.pl$, since gen and num are distinct agree subsets. On the other hand, UNIFY($[gen.f, num.sg], num.pl$) would fail.

$$\textsc{Move}(l_{1(M1)}, l_{2(L2)}) = \begin{cases} 1, Push(M_1, l_{2(Phon_2=0)}) & \text{if } L_2 \neq \varnothing. \\ 0, & \text{otherwise.} \end{cases} \tag{2.2.4}$$

The definition of MOVE tells us that an item (l_2) must be moved (pushed [7]) into the memory buffer (M_1) of the superordinate item (l_1) if it still has expected features to be selected ($L_2 \neq \varnothing$). This happens either in case L_2 is not selected by l_1 or it is selected by a $+L_2$ feature. Notice that the l_2 item that is moved in M_1 is not an exact copy of l_2: the used features (including $Phon$) will not be stored in memory. This definition produces the expected derivation if it applies right after $Merge$. In this case, if l_2 still has $exp(ect)ed$ features to be licensed, it must hold in the memory buffer of the selecting item, waiting for a proper selection of what has become the new l_2 label (i.e. Y). (Re-)Merge is then when agreement will be attempted. In the end, the top-down derivation in SV languages would unroll as follows: the subject (a DP) is first selected by a superordinate item (presuppositional subject position, situation topic, focus, etc.)[8] then it gets (partially) stored in the M buffer of the selecting item in virtue of the unselected D features, then re-merged as soon as a proper predicate, expressing the relevant T category requiring agreement (T should be included in the parameterized Agreement), is merged and properly selects a D argument (or it selects a V that later selects D). The content of the memory buffer is transmitted (inherited) through the last selected expectation, namely when the expecting and the expected categories successfully merge and the expecting item has no more expectations to be fulfilled ($R_1 \neq \varnothing$). If the expecting item has expectations, then the expected item constitutes a nested expansion, and the inheritance mechanism is blocked:

$$INHERIT(l_{1(M1)}, l_{2(M2)}) = \begin{cases} 1, M_1 \mapsto M_2 & \text{if } \textsc{Merge}(l_1, l_2) \wedge R_1 \neq \varnothing. \\ 0, & \text{otherwise.} \end{cases} \tag{2.2.5}$$

The M buffer of the last selected item that does not have other expectations (namely a right phrasal edge, i.e., $S_x = \varnothing$) must be empty (i.e., $M = \varnothing$). If not, the derivation fails (i.e., it stops) since a pending item remains unlicensed:

$$SUCCESS(l_{x(S_x, M_x)}) = \begin{cases} 1, & \text{if } S_x = \varnothing \implies M_x = \varnothing. \\ STOP, & \text{otherwise.} \end{cases} \tag{2.2.6}$$

Notice that the sequential item must be properly selected ($= S_X$). If this is not the case, the inheritance would transmit the content of the memory buffer of the superordinate phase into the memory buffer of an adjunct or a restrictive relative clause, which

7 PUSH and POP are trivial functions operating on arrays: $insert$ (PUSH) / $remove$ (POP) an item to/from the first available slot of a stack or a priority queue.

8 We have various options to implement this selection: a specific feature ($focus$, $topic$, $presupposed$, etc.) can be added to the relevant item (but this would lead to a proliferation of lexical ambiguity, e.g. [$_D$ the ...] vs [$_{FOC\ D}$ the ...]) or we assume that certain superordinate items can select specific categories, without deleting them (e.g. [$_{+D\ \epsilon\ FOC}$]). In this implementation, I will pursue this second, more economical, alternative.

clearly qualify as (right-branching) islands. Therefore, the "restrictive" (since feature driven) MERGE definition (2.2.1) seems correct and empirically more accurate than "free Merge" (Chomsky, Gallego, and Ott 2019, p.238).

3. The Derivation Algorithm

We can now define the full-fledged top-down derivation algorithm which is common both to generation and to parsing tasks (§3.2). Consider cn to be the current node, exp the list of pending expectations and mem the ordered list of items in memory. We initialize our procedure by picking up an arbitrary node from $G.Lex$ as cn. Being cn the root node of our derivation(al tree) and w the array of words we want to produce/recognize, we can define the function $DERIVE(cn, w)$ as follows:

Algorithm 1 Common (to Parsing and Generation) derivation algorithm

```
procedure DERIVE(cn, w)                      ▷ cn is the current node, w the input tokens
    while cn.exp ∧ w do                      ▷ while cn has expectations and w is not null
        while cn.mem do                      ▷ while items are pending in memory of cn
            for each cn.mem[i] do
                if MERGE(cn.exp[0], cn.mem[i]) then
                    POP(cn.exp)              ▷ consume any matching expectation in mem first
                    POP(cn.mem)
                else
                    BREAK
                end if
            end for
        end while
        if MERGE(cn.exp[0], w[0]) then
            POP(cn.exp)                      ▷ consume the incoming token matching the expectation
            if w[0].exped then
                MOVE(cn, w[0])               ▷ Move the token if it still has unlicensed Cat(s)
            end if
            if w[0].exp then                 ▷ Depth-first strategy: set the current item w as cn
                cn = w[0]
                INHERIT(exp[0], w[0])        ▷ Check if the item is nested or sequential
                SUCCESS(w[0])
            end if
            POP(w)                           ▷ Consume the successfully merged input token
            if not cn.exp then               ▷ In case cn has no more expectations
                cn = w[0]
                while not cn.exp ∧ cn is not root do
                    cn = cn.mother           ▷ Select higher nodes as cn
                end while
            end if
        else
            FAIL                             ▷ If the incoming items cannot be integrated, stops
        end if
    end while
end procedure
```

Informally speaking, as long as we have lexical items to consume (w), we loop into the set of expectations of cn ($cn.exp$), first attempting to Merge items from $(cn.)mem$ (if any), as in the active filler strategy (Frazier and Clifton 1989), then consuming words in the input (being $w[0]$ the first available token). Remember that each word has $exp(ect)ed$ features (the first being the label L), $exp(ectations)$ and $agr(eement)$ features. Cns have their own mem that can be inherited only by the last expected item, and, apart from the root node, a mother. The derivation is then a depth-first, left-right (i.e., real-time) strategy to derive a structure given a grammar, a root node, and a sequence of lexical items to be integrated.

3.1 The role of lexical ambiguity

Ignoring Parameters, the derivation procedure defined in §3 should face lexical ambiguity: the same $Phon$ in $w[n]$ might be associated with multiple items l in Lex with different features; the default option is to initialize a new derivational tree for any ambiguous item in Lex. Given an ambiguity rate m in Lex, the derivation procedure would have an exponential order of complexity $O(m^n)$. We can mitigate this, either by selecting the element(s) bringing only coherent (i.e. expected) categories (a categorial priming strategy, Ziegler et al. 2019), possibly relying on monotonic selection based on the height of the functional category[9], or to use a statistical oracle, following Stabler's beam search strategy (Stabler 2013), to limit (or rank) the number of possible alternatives. It is however important to stress that lexical ambiguity is the major source of complexity in this derivation: syntactic ambiguity is greatly subsumed by the lexicon, being the source of structural differences related to the set of categorial expectations processed and to the order in which lexical items are introduced in the derivation. With the strict version of MERGE defined in (2.2.1), no attachment ambiguity is allowed, since a matching selection must be readily satisfied as soon as the relevant configuration is created (but see Chesi and Brattico 2018). This is not the case if we would admit "free Merge" instead of select/licensors-driven Merge: in the first case, admitting that MERGE$(l_{1(S1)}, l_{2(L2)})$ is possible also if $S_1 \neq L_2$ would produce a syntactic ambiguity which is (exponentially) proportional to the number of items merged in the structure. This is a crucial argument to prefer feature-driven Merge. Notice, moreover, that admitting that re-merge is also possible without proper licensors/selectors, would quickly lead to unbounded unstoppable recursion. This must be prevented if we want to avoid the halting problem. Therefore, the licensors/selectors option seems to be a more logical, constrained, solution.

4. Parameters

A set of parameters can alter the domain of e-MGs in a relevant way both excluding unwelcome structures (e.g. non-agreeing constituents) or including various kinds of "discontinuous" phrasal dependencies that cannot be implemented in an explanatorily

9 According to the cartographic approach previously mentioned (Rizzi 1997; Cinque 1999, 2002; Belletti 2004; Rizzi 2004; Rizzi and Cinque 2016), we can assume that the $Exp(ect)$ features are strictly and universally associated with each functional category: being F_1, F_2, F_3 three functional categories hierarchically ordered (e.g. <Dem, Num, Adj> preceding N, as in Greenberg's Universal 20 - GU 20, Cinque 2005) by default: $[F_1\ _{=F2}]$, $[F_2\ _{=F3}]$, $[F_3\ _{=N}]$; if however $[F_3]$ (or N) immediately follows F_1, then we consider the $_{=F2}$ specification able to license immediately any lower functional category, that is, in this case, F_3. This just excludes the word order $<F_1$, F_3, F_2>, coherently with GU 20 prediction.

satisfactory way, but that are attested in different languages (e.g. certain kinds of cross-serial dependencies). Parameters minimally operate on F and Cat to implement various linguistic assumptions. Without altering the general architecture of G and the dynamics of the derivation, certain parameters leave the generative power of the grammar unaltered (i.e. mildly context-sensitive), while others extend the power beyond the mildly context-sensitive domain. Here, I will introduce three such parameters: one dealing with restrictions on Agreement categories, another with Reconstruction (Chomsky 1977), and the last one dealing with the so-called head-complement directionality (Baker 2001). The first one (agreement parameterization) will not alter the generative power of the grammar, while the second (requiring "delayed expansion") will. In this second case, the derivation problem could become quickly intractable and must be constrained precisely. On the other hand, this risk seems necessary and worth to be explored, since a new analysis of classic cross-serial dependencies will be available, which is possibly, empirically speaking, a promising alternative (Chesi 2007; Chesi and Brattico 2018). For the last parameter (the "head-directionality" one), I will sketch two solutions that do not have an impact on the generative power of the grammar but induce quite different derivations associated with completely different online predictions.

4.1 Agree categories

Agreement is a cross-linguistically parametrized option inducing specific featural unification between two distinct items. A list of categories requiring agreement is provided in the $P(arameters)$ set of an e-MG, as well as the specific conditions for which agreement holds. For instance, in Italian, as in many other Romance languages, DPs fully agree in gender and number. To express this, we include $gen(der)$ and $num(ber)$, in association with D, A[10], and N categories in Agr (henceforth, Agr features in l, e.g. $[_{num.sing,gen.fem}]$, are abbreviated, i.e. $[_{sg,f}]$):

$$P.Agr = \{D.\{num, gen\}, A.\{num, gen\}, N.\{num, gen\}\} \tag{4.1.1}$$

This is sufficient to accept (15.a) but not (15.b):

(15)	a.	La	prima	notizia
		$[_{D;sg,f}$ la]	$[_{A;sg,f}$ prima]	$[_{N;sg,f}$ notizia]
		the.SG.F	first.SG.F	news.SG.F
	b.	La	*prime	notizia
		$[_{D;sg,f}$ la]	$[_{A;pl,f}$ prime]	$[_{N;sg,f}$ notizia]
		the.SG.F	first.PL.F	news.SG.F

Similarly, subject-verb agreement and object-past participle ($V.pp$) agreement is expressed as follows:

10 The nature of adjectival modification cannot be fully addressed here. For simplicity, we assume that intersective adjectives (e.g. “beautiful”, as well as restrictive relative clauses and adverbial adjuncts) get licensed by the superordinate item without being properly selected ($= / + X$) or forced to move, while others (e.g. “ordinals”) are expected by D (see footnote 9, Cinque 2002). This induces a tolerable level of lexical ambiguity (either we assume both $[_D the_{=A}]$ and $[_D the_{=N}]$ or $[_A \epsilon_{=N}]$; the second option, $[_D the_{=A}] + [_A \epsilon_{=N}]$ in case of DP only composed by D and N, seems more coherent with the cartographic intuitions, it reduces lexical ambiguity and can be formalized by the monotonic selection discussed in the previous footnote).

$$P.Agr = \{T.\{per, num\}, V.pp\{num, gen\}\} \tag{4.1.2}$$

In SVO languages, S will first be licensed higher than T (unless aux-S inversion applies), then T-S agreement should be checked (case checking), then the subject S should reach the thematic role. These three operations are implemented simply including the relevant features in the lexicon as in (16):

(16)	$[_{C}\epsilon_{+D,=T}]$,	$[_{T;3,s} ha_{+D,=V.pp}]$,	$[_{V.pp}$ cantato $_{=D}]$,	$[_{D;3,s}$ Maria]
		has.PRS.3SG	sung.PTCP.PST.SG.M	Mary.SG.F

Exemplifying the derivation (following the procedure presented in §3), the root node C (phonetically empty) is selected first as the current node cn (initialization step), then [$_D$ Maria] is merged, satisfying the $+D$ expectation of C[11]. The expect feature $+D$ does not delete the expected D feature (see note 9), therefore [$_D$ ... Maria] is inserted in the memory buffer of C (since $Maria.exp(ect)ed$ = D). [$_T$ ... ha $_{+D,=V}$] is then merged, satisfying the last expectation of C (i.e., $= T$). Since T is the last expected item, it inherits the content of the superordinate memory buffer C (i.e. [$_D$ Maria]). In virtue of the $+D$ expectation of T, [$_D$ Maria] is remerged, and since both T and D categories are in $P.Agr$, agreement must be verified between [$_{T;3,s}$ ha] and [$_{D;3,s}$ Maria]. The check is successful, but still [$_D$ ... Maria] remains in memory (because, again, of the $+D$ expectation of T), and it is transmitted to V, which, in the end, is the last expected category of T. Now the $= D$ expectation of V finally re-moves [$_D$ Maria] from memory and licenses it as a V first ("external") argument. In a similar vein, we can implement object clitic – past participle agreement. Notice however that the simple specification of the relevant categories in $P.Agr$ would predict that the past participle always agrees with the object, also when it just appears in a post-verbal position. This is an incorrect prediction as shown in (17):

(17)	a.		Maria	l'ha	cantata
			M.	it.-CL.SG.F has	sung.PTCP.PST.SG.F
	b.	*	Maria	ha cantata	una canzone
			M.	has sung.PTCP.PST.SG.F	[a song].SG.F
	c.		Maria	ha cantato	una canzone
			M.	has sung.PTCP.PST.SG.M	[a song].SG.F

To capture this, we need to restrict (certain) agreement configurations to elements that are moved/remerged, namely *V.pp* will be an agreement category only when merged with an item taken from memory (i.e., the clitic in (17a)). We express this by adding a superscript in $P.Agr$ relevant categories: i.e. $V.pp^M$. It is important to consider sub-specifications of V since we don't want V to agree with the (external) argument of an unergative predicate, (18b)) vs (18c)):

11 This instantiates the topic of the predication in a general sense. The features on C can be parametrized: with the $+D$ feature associated with C (or below), we obtain the SV parameterization (which is different from the classic head-directionality parameter solution; see §4.3).

(18) a. Maria ha corso
Maria.SG.F has run.PTCP.PST.SG.M

b. * Maria ha corsa
Maria.SG.F has run.PTCP.PST.SG.F

c. Maria è caduta
Maria.SG.F is fallen.PTCP.PST.SG.F
Maria has fallen

This can be captured, not only by marking those inflections in which the relevant agreement features are overt (i.e. $V.pp$, namely past participle) but also by considering that the external argument and the internal one are licensed by two different categories, v and V respectively (Kratzer 1996), and only the second is relevant in terms of agreement (this is also necessary for selecting the correct auxiliary, $have$, (18a), vs be, (18c)).

4.2 Delayed expectation

Both remnant movement (Haegeman 2000), was-für Split (Chesi and Brattico 2018), and reconstruction (Bianchi and Chesi 2014) seem to require some sort of "late expansion" of some complement. When the "delayed expectation" parameter is set, this becomes an option, and an expectation (possibly nested) can be procrastinated. If the item bearing such expectation has only one expect feature, the only available possibility is to wait for its re-merge and then expanding such expectation at that time. Certain (non-presuppositional) subjects that do not behave as islands and seem transparent to sub-extraction, require this option to be active. A significant contrast is reported in (19):

(19) [$_P$ Of which sculpture] is [$_D$ one copy $\_{P}$] ...

a. ... *absolutely [perfect $\_{D}$]?

b. ... already [available $\_{D}$] ?

In (19a) the subject [one copy $_{=P}$] is expected outside the predicative nucleus [perfect] (presuppositional subject) and there it can't receive its argument [$_P$ of which masterpiece] (it is in a nested position). In (19b), reconstruction is possible under the stage level predicate [available], but the P expectation of [one copy $_{=P}$] must wait to be fulfilled after the subject is reconstructed as an argument of the predicate. Similarly, to capture the relevant dependency in inverse copular constructions we need this option:

(20) La causa della rivolta sono le foto del muro
The cause of the riot are the pictures of the wall

According to (Moro 1997), [$_D$ cause] is the predicate, while [$_D$ picture] is the subject of this predicate. To integrate [$_D$ picture] into the correct position we need to include the relevant expectation under the predicate, i.e., [$_D$ cause $_{=D,=P}$][12], then wait for the $= D$ projection (delayed expectation) when the predicate is remerged after the copula (that selects a D that is qualified to be "a predicate", that is, it brings another D expectation). Notice that while agreement parameterization decreases the derivational complexity

12 Being the subject the "external argument", it should come first than $= P$, which is the expectation triggering Merge of the "internal argument" [$_P$ of the riot].

(restricting the set of successful merges), delayed expectation introduces an extra level of syntactic ambiguity that is proportional to the number of expectations of each lexical item (Chesi 2007). This mechanism is computationally very onerous: for n selection features, we would have an exponential number of options to consider (2^n, since any time we must decide if the selection should be readily expanded or not) if we do not restrict delayed expansion to contexts in which the derivation would fail otherwise (last resort). On the other hand, this solution allows the grammar to capture cross-serial dependencies in a way that is impossible with (C)MGs. The empirical relevant data is something like D_A, D_B, $and\ D_C\ are\ respectively\ P_{=D_{(A)}}$, $P_{=D_{(B)}}$, $and\ P_{=D_{(C)}}$ (e.g. *Axel, Bert and Carl are respectively married*$_{Axel}$, *divorced*$_{Bert}$, *and single*$_{Carl}$). To capture the correct order (and cross-serial analysis) <$P_{=D_{(A)}}$, $P_{=D_{(B)}}$, $\mathrm{P}_{=D_{(C)}}$> we can either assume that the memory buffer is not a Last-In-First-Out memory buffer (and this might not be sufficient, since we should also ensure that Adam, Bert, and Carl can be respectively uniquely selected by a specific predicate, and, apart from gender distinction, no other relevant features are encoded neither in the proper name lexical items, which all qualifies as D, or in the predicate categorial selection), otherwise, once the lexical item "respectively" is processed, late expansion is activated, and $P_{=D_{(A)}}$, first, then $P_{=D_{(B)}}$ are stacked "on hold" exactly as pending unselected items. At this point, the last predicate introduced by $and\ P_{=D_{(C)}}$ will receive the first available item in memory (C). The most prominent item will then become B, which is compatible with the last delayed expectation, $\mathrm{P}_{=D_{(B)}}$. In the end, $\mathrm{P}_{=D_{(A)}}$ will be satisfied with the last pending item A. The sketched derivation is probably not psycholinguistically very plausible (it is unlikely that A should wait for B and C to be integrated before being selected by $P_{=D_{(A)}}$), but this is a sound way to derive XX unbounded reduplication. We conclude that e-MGs with "late expansion" exceed the power of (C)MGs which are not able to perform this computation and capture XX-like unbounded dependencies.

4.3 Head directionality

One of the major sources of cross-linguistic variation has been historically associated with the so-called head-directionality parameter (Baker 2001): there are languages in which the head H precedes the complement C (as in English: "John reads$_H$ books$_C$") or the way around (as in Japanese, using English words for simplicity: "John books$_C$ reads$_H$"). This behavior is relatively systematic in both nominal and verbal domain, though significant variations must be accommodated (Biberauer, Holmberg, and Roberts 2014; Sheehan 2017). Under e-MGs parameterization approach, there are two available options to capture this cross-linguistic contrast. One is more general and probably better captures the intuition behind the head-complement directionality generalization. This is done by locally inverting the lexical selection: if in head-initial languages, A selects B, in head-final languages, B would select A, that is, if in English predicates select the arguments, in Japanese, arguments cast selections over the predicates. Extending this intuition to any functional and lexical category, pure mirror image languages would exist, but this is not a correct prediction (Kayne 2020). Either we restrict this inverse selection to certain categories (lexical, for instance, such as N, A, and V) or we adopt a purely antisymmetric perspective (Kayne 1994) and we assume that in SOV languages, simply S and O must move into the memory buffer and late discharged into V. Notice that there, again, the late expansion option would become a necessity: considering that S and O are both D, which is probably a too restrictive hypothesis (Case can surely operates as a filter for selection), the predicate should wait for integrating the prominent O argument (late expansion of $= D_O$) after the external

S argument is discharged (since $V_{=D_S=D_O}$). We do not have space for digging into the pros and cons of these proposals here, but it should be clear that both solutions are compatible with the e-MGs formalism and it remains an empirical matter to favor one implementation over the other.

5. Generation and Parsing Tasks

It is worth reminding the reader that so far, we just discussed a general derivation which should be the least common denominator in Parsing and Generation. That means that both a Generation and a Parsing procedure must be fully specified yet. The next two sections begin to fill this gap.

5.1 Generation

As far as Generation is concerned, the procedure described in §3 is a sufficient complete algorithm to produce a sentence with the associated, dependency-based, structural description. As long as the sequence of words w is concerned, once a root node is selected, it is easy to imagine a dynamic function, instead of the static ordered sequence w, that incrementally proposes items to be integrated, given the history of the derivation or, at least, the last expectation (a sort of structural priming, possibly enriched with semantic features if we add to the lexicon *Sem(antic)* specifications in addition to Cat and $Phon$ ones). Notice that the lexicon can include phonetically empty categories (e.g. the empty root complementizer licensing the pre-verbal subject position by means of a $+D$ feature); this is not a problem for the generation procedure, which consumes input tokens one by one, and then considers a phonetically empty category on a par with phonetically realized ones, namely each item should be postulated as an incoming token to be processed.

5.2 Parsing

As long as phonetically empty items are concerned, the Parsing procedure is minimally different since it must guess the presence of these items (e.g. the presence of an empty subject in pro-drop languages), by deducting that the w sequence received in input is incomplete/incompatible with certain structural hypotheses. Focusing exactly on the pro-drop case, one proposal (Brattico and Chesi 2020) relies on inflectional morphology as an overt realization of unambiguous person and number features cliticized on the predicate, hence doubling the (null) subject. Otherwise, only after a relevant category is selected (with its agreement features) and unmatched by the current input, the empty item could be postulated. This non-determinism is exacerbated by the attachment/selection ambiguity: given [l$_{1=/+X}$ [l$_{2=/+X}$]], for instance, an incoming item with $Xexp(ect)ed$ feature that should be merged with l_2 first, according to the derivation algorithm provided in §3, could, in fact, be merged also with l_1, assuming that $l_2 = X$ expectation can be satisfied with an empty item bearing X as $exp(ect)ed$. Similarly, an adjunct marked with $Yexp(ect)ed$ category could be merged with both l_1 and l_2 in [l_1 [l_2]] in case of lexical ambiguity ([l_1], [l_{1+Y}], [l_2], [l_{2+Y}]). In this sense, the derivation procedure in §3 is insufficient as a full-fledged parsing strategy and must be integrated with disambiguation routines dealing with the possibilities just mentioned. It is important to stress that these disambiguation strategies do not alter the general derivation procedure introduced here, which remains the lowest common denominator of Generation and Parsing in e-MGs: the major change will apply to the

MERGE success conditions, which should include the empty category option when one MERGE operation fails. The relation between grammar and parser (and, more generally, competence and performance) is then predicted to be monotonic.

6. On model coverage

Before concluding this paper, it is worth speculating both on the coverage of the proposed model and on the fit of the algorithmic derivation concerning available real performance data. The two issues are partially related since only by demonstrating the scalability of the model from a toy-grammar to a large-scale grammar we would probably be able to approach naturalistic datasets that are becoming now more and more popular (Brennan et al. 2016; Siegelman et al. 2022). The first issue, namely how robust is the actual derivation, as compared for instance with the state-of-art (dependency) parsers (e.g. Stanza, Qi et al. 2020), needs two steps: first, a large-scale decorated lexicon must be created, second, a full-fledged parsing algorithm must be integrated into the derivation algorithm proposed in §3. I will try to explore some proposals in this sense in the next two sections.

6.1 Grammar extraction: from UD to eMG

An easy way to build a large scale lexicon, compatible with the definition given in §2.1, is to start from available UD treebanks (de Marneffe et al. 2021) and follow a deterministic grammatical extraction procedure. This amounts to accepting that (i) the Cat set is restricted to the specific tagset used (e.g. UPOS), (ii) we can populate the $Exp(ect)ed$ and $Exp(ect)$ sets only in accordance with the dependency types defined in the UD treebank, (iii) only the annotated "morphological features" can be used to refine the Cat set and isolate the relevant $Agree$ features. All these steps have been implemented, but require some critical discussion: as far as the Cat set is considered, we should notice that the set of functional categories available in UD is relatively poor with respect to the one assumed by the cartographic inquiry (e.g. Cinque 1999 for the verbal functional fields and Rizzi 1997 for the complementizer field): simply looking at the nominal domain, we notice that quantifiers like *all* and definite determiners like *the* are tagged in the same way (DET); this is a logical problem for the e-MG approach, since, for certain categories, in a given language, precise constraints restrict both linear order (*all the books* vs. **the all books*) and complementary distribution (**some the books*). If a relevant sub-categorization is missing, our structural description will be inadequate as far as grammatical prediction is concerned. Certain categorial sub-specifications can be recovered by combining morphological features and the POS tag (e.g. possessive adjectives can be distinguished by qualitative ones and this is crucial to predict *her beautiful books* vs. **beautiful her books*), but this is not always possible (e.g. unaccusative vs. unergative/transitive predicates are not indicated, so it is not possible to predict passivization constraints: *è stato visto/(he) has been seen* vs. **è stato caduto/(he) has been fallen*). Second, the absence of phonologically empty items (apart from the $root$ node) limits the set of non-local dependencies available: for instance, topicalization or focalization constructions, which are clearly distinct in Romance languages, as indicated by the presence of clitics in the first case but not in the second [13].

Moreover, some standard dependency orientation in UD is exactly the opposite with respect to the one assumed in Generative Grammar: This is for instance the case of the dependency between Determiners and Nouns or between the Auxiliary and the Past Participle predicate. In both cases, in UD, Determiners and Auxiliaries are dependent on Nouns and Verbs respectively, while in constituency-based descriptions, functional categories dominate the lexical ones, that is, they "select" (in e-MGs terms) their lexical category. The difference is substantial here: first, we cannot exclude certain bare nouns in argumental positions (e.g. *boys read comics* vs. **boys read book*), second, if we want to apply the derivation procedure defined in §3, each leftward dependency in UD would trigger a MOVE operation and this might be a costly option.

To assess the impact of these factors, a deterministic extraction procedure has been used and applied to the training sets taken from four distinct UD treebanks: Two treebanks representing head-initial languages, the UD English GUM for English (Zeldes 2017) and UD Italian ISDT (Bosco, Dell'Orletta, and Montemagni 2014) has been selected, together with two head-final language treebanks, the UD Turkish PENN treebank (Oflazer et al. 2003), representing an agglutinative language, and the UD Japanese GSD treebank (Tanaka et al. 2016; Asahara et al. 2018). All treebanks adopted the UPOS tagset. Table 1 reports the results of this extraction, indicating the level of ambiguity (distinguishing between Lexical, i.e. POS related, Morphological, i.e. related to morphosyntactic featural specification, and dependency-based ambiguity, i.e. the number of dependents that project dependencies in a non-uniform way, for instance, sometimes selecting one argument, some other time two or three arguments). The amount of "backward dependencies", namely a dependency towards a preceding item (i.e. those triggering movement in e-MGs) are also indicated with a specification of the locality of such dependencies (in percentage, how many times these "backward dependencies" are resolved with the immediately preceding item).

Table 1
Ambiguities ratio and dependency locality in the extracted lexica

	EN	*IT*	*TR*	*JP*
Tokens processed	126530	294403	166514	168333
Lexicon size (Types)	13757	27021	33036	20140
Ambiguity ratio	0.38	0.28	0.34	0.33
Lexical ambiguity	*0.33*	*0.22*	*0.20*	*0.25*
Morphological ambiguity	*0.17*	*0.03*	*0.08*	*< 0.01*
Dependencies ambiguity	*0.50*	*0.75*	*0.72*	*0.75*
Backward dependency ratio	0.69	0.61	0.83	0.48
Locality ratio	*0.54*	*0.60*	*0.69*	*0.32*

All in all, the level of ambiguity is relatively stable across corpora ($\approx$ one token over three is ambiguous). This level of ambiguity is determined, in 3/4 of the cases, by

13 This is the "clitic left dislocation", CLLD, construction (Cecchetto 1999):

Il libro,	l'ho	letto	vs.	IL LIBRO	(*l')ho	letto
the book,	it.CL (I) have	read (it)		the book	(it.CL) (I) have	read (it)

the ambiguity in establishing dependencies: This is a relatively precise measure of the syntactic ambiguity that, also in e-MGs, needs to be solved by relying on probabilistic cues since the sub-categorization information is not sufficient here. Less problematic is the lexical and morphological ambiguity (from 1/3 in English to about 1/4 of these cases in other languages) for which constraints on local categorial selection might be sufficient. Even though a precise accuracy measure is out of the scope of this paper, one comforting piece of information comes from the locality levels of the backward dependency: apart from Japanese (which however presents a significantly lower number of backward dependencies: 48% vs. 61-83% revealed in other datasets), more than half of them (54-69%) can be trivially (i.e. locally) resolved, in the end suggesting that (considering an appropriate parameterization from language to language), the usage of memory buffer is in these cases not very onerous (an item gets store and immediately integrated as a selected item of the next token).

6.2 Robustness, predictivity in performance data, and the utility of minimal contrasts

A last concern is related both to the robustness and to the applicability to real performance data of the proposed formalism. Even though this topic would require an independent paper, it is worth making some considerations in this direction: first of all, the availability of various performance datasets (Brennan et al. 2016; Siegelman et al. 2022) collected in various experiments, including ecological listening/reading experiences, can be partially exploited in a relevant way also without full implementation of the ambiguity resolution strategy advocated in the previous section. In this sense, a relevant application of this formalism is discussed in the literature (Chesi and Canal 2019) and it is readily available: Chesi and Canal show how simple complexity metrics associated with the derivation presented in §3 can be formulated: This includes an integration cost (on the line of Gibson 1998) and a featural intervention cost (refinement of Friedmann, Belletti, and Rizzi 2009). These two components are sufficient to predict both off-line grammaticality/acceptability judgments and on-line reading times (gaze duration and total reading time in eye-tracking) in certain non-local dependencies better than alternative models (notably Lewis and Vasishth 2005; Friedmann, Belletti, and Rizzi 2009; Gibson 1998; Gordon, Hendrick, and Johnson 2004).

Given the level of ambiguity revealed in the previous chapter, the proposed model, as it is, cannot be compared with more tolerant memory-based / expectation-based hybrid models, such as the one discussed in (Futrell, Gibson, and Levy 2020). Phenomena such as "structural forgetting" cannot be readily accounted for in the explicit derivational approach here proposed, unless the probability of application of structure building operations is considered and modeled.

I think it is however important to stress that this perspective provides an explanatory option, which is precluded by most of the alternative probabilistic approaches: comparing minimal pairs still has the advantage of understanding the impact of single featural modifications or minimal reordering in structure-building operations that are overtly specified. The grammar here defined is not "robust" at all in a standard sense, that is, it does not return a plausible structure for any possible ill-formed input and it is not able to process sentences in which OOV words are included. But unlike probabilistic approaches, it can precisely predict why a specific sequence of words is never attested (e.g. *these books* or *three books* or *these three books* vs. **three these books*). The cost of these predictions is a richer (fine-grained) set of categories to be considered (much richer than UPOS tagset) and we are aware of the implication that this richer set might have on efficiency and robustness of various models training (Anderson and Gómez-Rodríguez 2020).

7. Conclusions

The e-MGs formalization proposed here is a simple (parametrized) framework suitable for comparing syntactic (competence-based) predictions and human parsing/generation performance. This is made possible by the core derivation assumed, which is the same in both tasks (back to the token transparency hypothesis discussed by Miller and Chomsky 1963). While there is little to add to implement a full-fledged Generation procedure, as long as the Parsing perspective is concerned, the information asymmetry of this task with respect to Generation requires extra routines to be implemented, in addition to the basic derivation algorithm: lexical ambiguity must be resolved "on-line" and phonetically empty items must be postulated when needed. This creates an extra level of complexity which is however manageable under the same derivational perspective here presented: the core derivation is sufficiently specified to operate independently from Parsing-specific disambiguation assumptions that apply, monotonically, to MERGE, MOVE and AGREE. Measuring the actual source of ambiguities and non-determinism from available UD treebanks, without entering into a complex assessment of the annotated structure available, we concluded that, despite a poor categorial specification, less than about one-fourth of the non-local dependencies would require extra probabilistic considerations to be addressed properly. In the end, then, this parametrized grammatical model provides an interesting foothold for complexity metrics that aim at comparing the predicted difficulty not only globally (De Santo 2020; Graf, Monette, and Zhang 2017) but also "on-line" that is, on a word by word basis (Chesi and Canal 2019) as illustrated in the simple implementation of the derivation algorithm discussed in this paper[14]: the first step for creating a large scale lexicon has been implemented; now it remains to be checked the on-line predictions based on the proposed derivations/parameterization against real performance data (e.g. reading times, gaze duration/fixation or regressions). While the first results on minimal pair comparisons (for instance in relative clause and other non-local dependencies processing) seem promising, a full commitment with specific structural attachment disambiguation strategies (either formal or probabilistic) remains to be explored.

References

Alexiadou, Artemis and Elena Anagnostopoulou. 1998. Parametrizing AGR: Word order, v-movement and EPP-checking. *Natural Language & Linguistic Theory*, 16(3):491–539. Publisher: Springer.

Anderson, Mark and Carlos Gómez-Rodríguez. 2020. On the frailty of universal POS tags for neural UD parsers. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 69–96, Online, November. Association for Computational Linguistics.

Asahara, Masayuki, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies version 2 for Japanese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Baker, Mark C. 2001. *The atoms of language*. Basic Books, New York, 1st ed edition.

Belletti, Adriana. 2004. *Structures and Beyond: The Cartography of Syntactic Structures, Volume 3*. Oxford University Press.

14 https://github.com/cristianochesi/e-MGs

Bianchi, Valentina and Cristiano Chesi. 2006. Phases, left-branch islands, and computational nesting. *Proceedings of the 29th Annual Penn Linguistics Colloquium*, 12.1:15–28.

Bianchi, Valentina and Cristiano Chesi. 2014. Subject islands, reconstruction, and the flow of the computation. *Linquistic Inquiry*, pages 525–569.

Biberauer, Theresa, Anders Holmberg, and Ian Roberts. 2014. A syntactic universal and its consequences. *Linguistic Inquiry*, 45(2):169–225.

Bosco, Cristina, Felice Dell'Orletta, and Simonetta Montemagni. 2014. The Evalita 2014 Dependency Parsing Task. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 and of the Fourth International Workshop EVALITA 2014 9-11 December 2014, Pisa*. pisa university press.

Brattico, Pauli and Cristiano Chesi. 2020. A top-down, parser-friendly approach to pied-piping and operator movement. *Lingua*, 233(102760):1–28. Publisher: Elsevier.

Brennan, Jonathan R., Edward P. Stabler, Sarah E. Van Wagenen, Wen-Ming Luh, and John T. Hale. 2016. Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and Language*, 157-158:81–94.

Cecchetto, Carlo. 1999. A Comparative Analysis of Left and Right Dislocation in Romance. *Studia Linguistica*, 53(1):40–67, April.

Chesi, Cristiano. 2005. On the directionality of non-local dependencies. In *HOWL3: Hopkins Workshop on Language, Non-local dependencies in phonology and syntax*, pages 9–12. John Hopkins University.

Chesi, Cristiano. 2007. Five reasons for building phrase structures top-down from left to right. In *Nanzan Linguistics (Vol. 3 Special)*. Nanzan University, pages 71–105.

Chesi, Cristiano. 2012. *Competence and Computation Toward a Processing Friendly Minimalist Grammar*. Unipress.

Chesi, Cristiano. 2015. On directionality of phrase structure building. *Journal of Psycholinguistic Research*, pages 65–89.

Chesi, Cristiano. 2018. An efficient trie for binding (and movement). In *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253.

Chesi, Cristiano. 2021a. A common derivation for parsing and generation with expectation-based minimalist grammars (e-MGs). In *CLiC-it 2021 Italian Conference on Computational Linguistics 2021*, volume 3033. CEUR-WS. org. Issue: urn: nbn: de: 0074-3033-7.

Chesi, Cristiano. 2021b. Expectation-based minimalist grammars. *arXiv:2109.13871 [cs]*.

Chesi, Cristiano and Pauli Juhani Brattico. 2018. Larger than expected: constraints on pied-piping across languages. *RGG. Rivista di Grammatica Generativa*, 2008.4:1–38.

Chesi, Cristiano and Paolo Canal. 2019. Person features and lexical restrictions in italian clefts. *Frontiers in Psychology*.

Chomsky, Noam. 1977. On wh-movement. In Peter W Culicover, Thomas Wasow, and Adrian Akmajian, editors, *Formal syntax*, volume 65. Academic Press, San Francisco, CA.

Chomsky, Noam. 1995. *The minimalist program*. MIT press.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz, editor, *Ken Hale: A life in language*, 36. MIT Press, pages 1–52.

Chomsky, Noam, Ángel J Gallego, and Dennis Ott. 2019. Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*, pages 229–261.

Cinque, Guglielmo. 1999. *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press.

Cinque, Guglielmo, editor. 2002. *Functional structure in DP and IP*. Number v. 1 in Oxford studies in comparative syntax. Oxford University Press.

Cinque, Guglielmo. 2005. Deriving greenberg's universal 20 and its exceptions. *Linguistic Inquiry*, 36(3):315–332.

de Marneffe, Marie-Catherine, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, 07.

De Santo, Aniello. 2020. MG parsing as a model of gradient acceptability in syntactic islands. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 59–69. Association for Computational Linguistics.

Earley, Jay. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Frazier, Lyn and Charles Clifton. 1989. Successive cyclicity in the grammar and the parser. *Language and Cognitive Processes*, 4(2):93–126.

Friedmann, Naama, Adriana Belletti, and Luigi Rizzi. 2009. Relativized relatives: Types of intervention in the acquisition of a-bar dependencies. *Lingua*, 119(1):67–88. Publisher: Elsevier.

Futrell, Richard, Edward Gibson, and Roger P. Levy. 2020. Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing. *Cognitive Science*, 44(3).

Gibson, Edward. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76, August.

Gordon, Peter, Randall Hendrick, and Marcus Johnson. 2004. Effects of noun phrase type on sentence complexity. *Journal of memory and Language*, 51(1):97–114. Publisher: Elsevier.

Graf, Thomas, James Monette, and Chong Zhang. 2017. Relative clauses as a benchmark for minimalist parsing. *Journal of Language Modelling*, 5(1).

Haegeman, Liliane. 2000. Remnant movement and OV order. In Peter Svenonius, editor, *The derivation of VO and OV*. John Benjamins Publishing Company Amsterdam, pages 69–96.

Hale, John. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Hale, John. 2011. What a rational parser would do. *Cognitive Science*, 35(3):399–443. Publisher: Wiley Online Library.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in generative grammar*. Number 13 in Blackwell textbooks in linguistics. Blackwell.

Huang, C.-T. James. 1982. *Logical relations in Chinese and the theory of grammar*. phdthesis, MIT.

Kayne, Richard S. 1994. *The antisymmetry of syntax*. Number 25 in Linguistic inquiry monographs. MIT Press.

Kayne, Richard S. 2020. Antisymmetry and externalization. *Ms. New York University*.

Kobele, Gregory M. to appear. Minimalist Grammars and Decomposition. In Grohmann Kleanthes K. and Evelina Leivada, editors, *The Cambridge Handbook of Minimalism*. Cambridge University Press, Cambridge (UK).

Konieczny, Lars and Philipp Döring. 2003. Anticipation of clause-final heads: Evidence from eye-tracking and srns. In *Proceedings of iccs/ascs*, pages 13–17. Sydney, NSW.

Kratzer, Angelika. 1996. Severing the external argument from its verb. In Johan Rooryck and Laurie Zaring, editors, *Phrase Structure and the Lexicon*, volume 33. Springer Netherlands, pages 109–137. Series Title: Studies in Natural Language and Linguistic Theory.

Levy, Roger. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177. Publisher: Elsevier.

Levy, Roger and Frank Keller. 2013. Expectation and locality effects in german verb-final structures. *Journal of Memory and Language*, 68(2):199–222.

Lewis, Richard L and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419. Publisher: Wiley Online Library.

Marr, David. 1982. *Vision: a computational investigation into the human representation and processing of visual information*. Freeman, San Francisco (CA), w.h. freeman edition.

Miller, George A. and Noam Chomsky. 1963. Finitary models of language users. In D. Luce, editor, *Handbook of Mathematical Psychology*. John Wiley & Sons., pages 2–419.

Momma, Shota and Colin Phillips. 2018. The relationship between parsing and generation. *Annual Review of Linguistics*, 4(1):233–254. _eprint: https://doi.org/10.1146/annurev-linguistics-011817-045719.

Moro, Andrea. 1997. *The raising of predicates: Predicative noun phrases and the theory of clause structure*, volume 80. Cambridge University Press.

Nivre, Joakim, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, and others. 2017. Universal dependencies 2.1.

Nunes, Jairo and Juan Uriagereka. 2000. Cyclicity and extraction domains. *Syntax*, 3(1):20–43.

Oflazer, Kemal, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish Treebank. In Nancy Ide, Jean Véronis, and Anne Abeillé, editors, *Treebanks*, volume 20. Springer Netherlands, Dordrecht, pages 261–277. Series Title: Text, Speech and Language Technology.

Partee, Barbara H., Alice ter Meulen, and Robert E. Wall. 1993. *Mathematical methods in linguistics*. Number volume 30 in Studies in linguistics and philosophy. Kluwer Academic Publishers, corrected second printing of the first edition edition.

Pickering, Martin J. and Simon Garrod. 2013. An integrated theory of language production and comprehension. *Behavioral and Brain Sciences*, 36(4):329–347.

Pollard, Carl Jesse and Ivan A. Sag. 1994. *Head-driven phrase structure grammar*. Studies in contemporary linguistics. Center for the Study of Language and Information ; University of Chicago Press.
Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
Rizzi, Luigi. 1997. The Fine Structure of the Left Periphery. In Liliane Haegeman, editor, *Elements of Grammar*. Springer Netherlands, Dordrecht, pages 281–337. Series Title: Kluwer International Handbooks of Linguistics.
Rizzi, Luigi. 2004. *The Structure of CP and IP: The Cartography of Syntactic Structures Volume 2*, volume 2. Oxford University Press on Demand.
Rizzi, Luigi. 2016. Labeling, maximality and the head–phrase distinction. *The Linguistic Review*, 33(1):103–127. Publisher: De Gruyter Mouton.
Rizzi, Luigi and Guglielmo Cinque. 2016. Functional categories and syntactic theory. *Annual Review of Linguistics*, 2(1):139–163.
Sheehan, Michelle. 2017. The final-over-final condition and processing. In Michelle Sheehan, Theresa Biberauer, Ian Roberts, and Anders Holmberg, editors, *The Final-Over-Final Condition: A Syntactic Universal*, volume 76. MIT Press, pages 79–98.
Siegelman, Noam, Sascha Schroeder, Cengiz Acartürk, Hee-Don Ahn, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, Sara Maria Da Fonseca, Nicolas Dirix, Wouter Duyck, Argyro Fella, Ram Frost, Carolina A. Gattei, Areti Kalaitzi, Nayoung Kwon, Kaidi Lõo, Marco Marelli, Timothy C. Papadopoulos, Athanassios Protopapas, Satu Savo, Diego E. Shalom, Natalia Slioussar, Roni Stein, Longjiao Sui, Analí Taboh, Veronica Tønnesen, Kerem Alp Usal, and Victor Kuperman. 2022. Expanding horizons of cross-linguistic research on reading: The Multilingual Eye-movement Corpus (MECO). *Behavior Research Methods*, 54(6):2843–2863.
Sprouse, Jon, Matt Wagers, and Colin Phillips. 2012. A test of the relation between working-memory capacity and syntactic island effects. *Language*, 88(1):82–123.
Stabler, Edward. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, pages 68–95. Springer Berlin Heidelberg.
Stabler, Edward. 2011. Computational perspectives on minimalism. In Cedric Boeckx, editor, *The Oxford Handbook of Linguistic Minimalism*. Oxford University Press.
Stabler, Edward. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5(3):611–633.
Tanaka, Takaaki, Yusuke Miyao, Masayuki Asahara, Sumire Uematsu, Hiroshi Kanayama, Shinsuke Mori, and Yuji Matsumoto. 2016. Universal Dependencies for Japanese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1651–1658, Portorož, Slovenia, May. European Language Resources Association (ELRA).
Zeldes, Amir. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.
Ziegler, Jayden, Giulia Bencini, Adele Goldberg, and Jesse Snedeker. 2019. How abstract is syntax? evidence from structural priming. *Cognition*, 193:104045.