Supplementary appendix to the article "Predicate clefting and long head movement in Finnish" published in *Linguistic Inquiry*

Pauli Brattico

IUSS University School for Advanced Studies

2021

Abstract. This document is a supplementary appendix to the original article "Predicate clefting and long head movement in Finnish" published in *Linguistic Inquiry*. After providing further contextualization to the analysis, it addresses details of the computational implementation that were omitted or only briefly mentioned in the main article. Results of the simulation are examined in detail.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Framework

This document constitutes a supplementary appendix to the original article "Predicate clefting and long head movement in Finnish." The two documents approach the subject matter from complementary perspectives. Whereas the main article was concerned with details of the empirical data, its interpretation in the light of the current linguistic theory, and used the computational framework mainly for the justification of the analysis, this document elucidates the implementation in some more detail and furthermore discusses in depth the raw output produced by the algorithm. The proposed approach is compared with what I consider to be a fairly standard cyclic grammatical model at present writing. Results of the simulation experiments are presented in detail.

The main empirical problem addressed in this work can be introduced in the following way. We elicit linguistic data from human informants by having them judge properties of linguistic expressions, such as their grammaticality, acceptability, meaning and pragmatic felicity. Additional facts can be gathered by controlled experiments. Our goal is to provide an equation (formal algorithm, model) that captures these facts.[1] Here we are specifically interested in explaining such facts inasmuch as they

---

[1] First attempts at developing this approach go back to Chomsky (1957, 1964, 1965). These works were in turn preceded by linguistic structuralism which shared the explicit goal of constructing formal and mechanical models but were restricted by a narrow positivistic interpretation of science that limited their utility. My view is that the use of computational methdologies should be applied to justification, not to discovery. See my paper "Computational linguistics as natural science," currently available in LingBuzz.

concern the syntactic and semantic processing of morphologically complex phonological words, phenomena that fall mostly under the umbrella of "head movement" in the standard generative theory (Baker 1988; Koopman 1984; Matushansky 2006; Roberts 2010; Travis 1984). By saying that the model captures such facts we mean that it is sufficient to calculate the data; if the model is also necessary, then it satisfies an additional criterion of being also the simplest possible hypothesis.[2] Ultimately, though, we aim for a model that is both necessary and sufficient to capture the widest possible range of facts.

The number of possible grammatical frameworks and models one could adopt as a starting point in cognitive science is large, ranging from complex generative models to simple Pavlovian connectionist approaches. The standard method used in the more advances sciences for solving disputes of this type is to formalize the competing hypotheses, show that they calculate the facts, and then compare their accuracy, simplicity and generality. The choice of the starting point therefore reflects any author's subjective guess of what the best approach might be, but justification relies on calculation. We proceed by using this method and expect all disagreement to be solved in the same way.

Due to the combinatorial nature of linguistic data, the logical distance between theory and data tends to be vast. To overcome this problem, the hypothesis was written in a machine-readable formalism (Python programming language) so that we can examine its logical consequences efficiently and reliably by using a computer. To make this possible, the data was collected into a test corpus file and processed by a computer program embodying the hypothesis. We are interested in whether the output of the model agrees or disagrees with human behavior, both in relation to offline judgments and real-time reactions. The study design is summarized in Figure 1.

---

[2] Perhaps this requirement constitutes the core of "minimalism" in the sense of Chomsky (1995), but if so then minimalism would be standard scientific method applied to generative theorizing. My reading of the thesis proposed by Chomsky in the source cited and others is that it goes in some respects beyond this characterization, but the matter lies beyond the scope of the present inquiry.

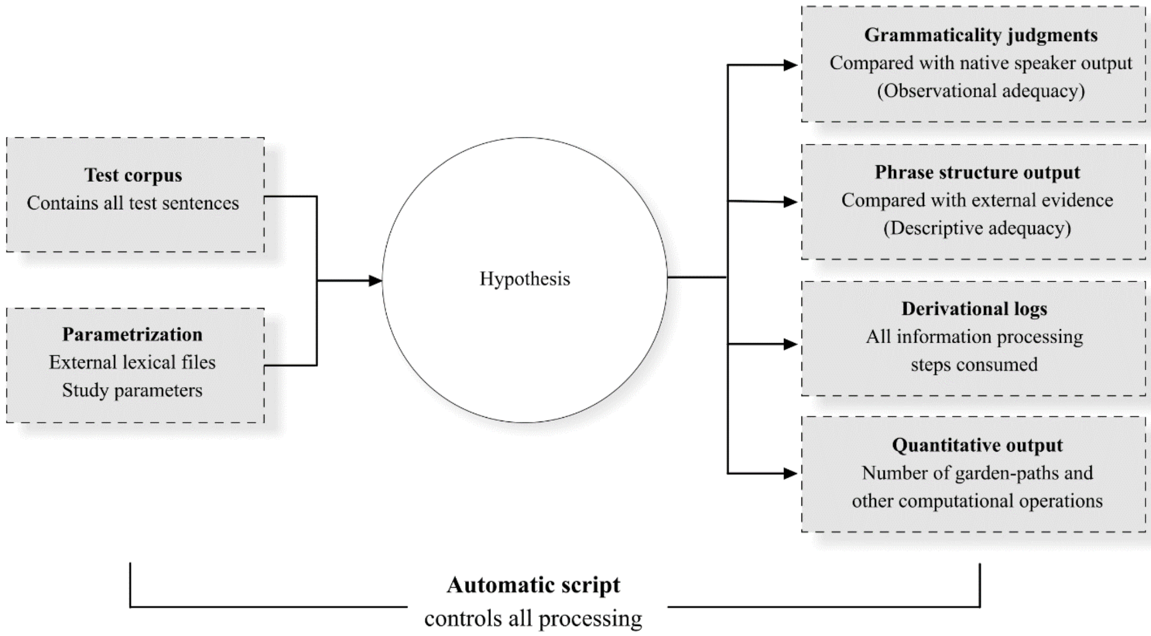


Figure 1. Design of the study. A test corpus containing sentences whose properties we are interested in together with the input parameters are used as the input to an algorithm embodying the hypothesis.

There are two in principle ways of bridging a formal linguistic hypothesis with a dataset: generation and recognition. In a generative theory, the axioms of the theory are employed to generate a set of expressions (phonological strings, syntactic analyses, semantic interpretations) from a set of lexical elements and other input parameters. The output set is compared with data from native speakers. The other method is recognition, in which the theory judges input expressions by linking them with grammaticality judgements, syntactic analyses and semantic interpretations. The two methods are mathematically equivalent under weak assumptions: it is easy to manufacture a recognition model from a generative algorithm, and vice versa. The recognition approach was adopted in this study. Even though a recognition algorithm can be and is sometimes seen as a performance hypothesis modelling real-time language comprehension, it incorporates competence inasmuch as it provides correct grammaticality judgments and replicates or otherwise correctly describes human semantic intuitions. In fact, once we develop an algorithm of this type it is often easy to separate its performance properties from those that are involved in competence calculations. Performance issues are discussed in Section 3.1; they are irrelevant to the main hypothesis.

There is an additional benefit when the hypothesis is formalized as a computer program. Despite the fact that an enormous amount of calculations must be performed, they do not constitute an obstacle for any modern commercial-grade computer. The non-optimized and serial program utilized in this research consumes approximately 100ms on a standard laptop computer when it processes a single canonical sentence. This means that it is possible to verify the correctness of any hypothesis reliably over a potentially huge number of relevant data (sentences) in few minutes. This increases rigor, and the standards for justification, in the field.

1.2 Data

The dataset was discussed extensively in the main article and will not be repeated here. I will provide few examples for reference and clarify few issues discussed in the main article. An example of the empirical phenomena we are interested in in this study is provided in (1), where we pay special attention to the first word. The language under study is Finnish.[3]

(1)	Myy-dä-kö$_1$	Pekka	halus-i	__$_1$	koko	omaisuute-nsa?
	sell-A/INF-Q	Pekka.NOM	wanted-PST.3SG		all	possessions-PX/3SG

‘Was it selling that Pekka wanted to do with all his possessions?’

‘Did Pekka wanted <u>to sell</u> all his possessions?’[4]

---

[3] Symbol PX/3SG refers to the Finnish infinitival agreement suffix (Brattico and Huhmarniemi 2016; Huhmarniemi and Brattico 2015). Finnish is a canonical SVO language with a relatively free word order associated with a degree of discourse configurationality, agglutinative morphology and the nominative-accusative case system, with nominative grammatical subjects agreeing with finite elements in the clause. Grammaticalized articles are absent. Most of the example sentences cited in this document follow the canonical SVO template. For Finnish syntax in general, see Huhmarniemi (2012) and Manninen (2003).

[4] Translating these expressions into English is difficult. The most natural way to translate these sentences into English seems to be prosody. In some cases, the gerund construction captures the meaning quite well, but the target predicate is not gerund and Finnish has a separate gerund construction (*oliko se talon myyminen mitä Pekka suunnitteli?* ‘was it house selling what Pekka planned?’). The predicate is always what would appear in the canonical position marked by __ minus the left peripheral feature, and there is no extra clefting structure. The essential point is that the fronted verbal element

The first word *myy-dä-kö* 'sell-A/INF-Q' contains a transitive verbal stem *myy-* 'sell', an infinitival suffix -*tA* ('to') and a yes/no interrogative particle -*kO* (glossed as Q). The word therefore carries considerable amount of grammatical information, all intuitively and instantly understood by native speakers. Consequently, human language seems to make a distinction between three ways of articulating grammatical information. One way is by chunking the information inside one phonological word, here *myy-dä-kö* 'sell-A/INF-Q'. Another is by means of separate words, so that the relevant grammatical information is dispersed into the sentence as independent phonological units that are in the domain of phrasal syntactic rules. In English, for example, the complex meaning expressed by *myy-dä-kö* 'sell-A/INF-Q' requires at least three separate units, *whether* (yes/no interpretation), *to* (infinitival) and *sell* 'transitive verb'. Finally, a third way of articulating grammatical information is by inserting words into canonical and noncanonical positions. The canonical position of the complex infinitival verb is designated by $\_\__1$ in the example (1), but in this sentence it appears at the beginning of the clause, which we consider a noncanonical position for an element of this type. The verb should be where the empty position is. The canonical position of the word is shown in (2).

(2)	Pekka	halus-i	**myy-dä**	koko	omaisuutensa.
	Pekka	want-PST.3SG	sell-A/INF	all	possessions
	'Pekka wanted to sell all his possessions.'				

The dislocation operation is not redundant or irrelevant either, because it conveys grammatical information, namely the scope of the question.

To examine these phenomena further, we gather observations by presenting sentences of this type plus many variations to native speakers and ask them to judge whether the resulting sentences are possible or impossible and, if the former, what their meaning is. This investigation, when carried systematically, provides observations that we add to our research agenda (and to the test corpus). For example, the investigation shows that the fact that the infinitival word has been dislocated to the

---

(e.g., 'to sell') is targeted for an operator interpretation triggered by the left peripheral morphemes and their content (here -*kO*, 'did') and scope is fixed by movement.

noncanonical first position is related in some way to the presence of the yes/no morpheme Q. The presence of this morpheme forces the word to dislocate (3). If the morpheme is not present, the word can remain in its canonical position.

(3)	*Pekka	halus-i	myy-dä-**kö**	koko	omaisuutensa?
	Pekka	want-PST.3SG	sell-A/INF-Q	all	possessions

The dislocation operation can furthermore target only one syntactic position in the clause: the word cannot occur in any other place expect in the first position. In addition, if the word gets dislocated without the yes/no morpheme or any other overt left peripheral feature, it will receive a special contrastive interpretation and normally requires also prosodic stress (4).

(4)	MYY-DÄ$_1$	Pekka	halus-i	__$_1$	koko	omaisuutensa.
	sell-A/INF	Pekka	want-PST.3SG		all	possessions

‘It was selling (not loaning) that Pekka wanted to do with his possessions.’

Therefore, what we are seeing cannot be “idle rearrangement” that speakers could do or left undone without being aware of the changes such operations induce to the meaning (and hence also to the use) of the sentence. We must capture these semantic changes in the model.

These special interpretational effects (focusing, interrogativization) and the features that trigger them (stress, wh-features, Q) belong to a larger left peripheral C-system that involves also phrasal operator movement (Huhmarniemi 2012; Vilkuna 1989, 1995). The operation in (1) and phrasal movement use the same position, as shown in (5a). The two operations are complementary (5b), reinforcing the conclusion that we are looking at two variations of the same mechanism.

(5)	a.	[Koko	omaisuute-nsa-**ko**]$_1$	Pekka	halus-i	myy-dä	__$_1$?
		all	possessions-PX/3SG-Q	Pekka.NOM	want-PST.3SG	sell-A/INF	

‘Was it all his possessions that Pekka wanted to sell?’

b.	*Koko	omaisuute-nsa-**ko**	myydä-**kö**	Pekka	halus-i?
	all	possessions-PX/3SG-Q	sell-A/INF-Q	Pekka	want-PST.3SG

Since the same features and same syntactic position is involved in head movement and phrasal movement, our model must capture properties of both.

The dislocated element must correspond to a gap in the clause, and its grammatical properties must match with the properties of that gap. If the gap is filled in by an independent element (6a), or if the dislocated word is an infinitival that does not match with the grammatical position of the gap (here the control VA-infinitival)(b), the outcome is unacceptable.

(6)	a.	*Myy-dä-kö	Pekka	halus-i	**osta-a**	koko	omaisuutensa?
		sell-A/INF-Q	Pekka	want-PST.3SG	buy-A/INF	all	possessions
	b.	*Myy-**vän**-kö$_1$	Pekka	halus-i	__$_1$	koko	omaisuutensa?
		see-VA/INF-Q	Pekka	want-PST.3SG		all	possessions

We can therefore be reasonably certain that the dislocated word in (1) has been "moved" in some sense from its canonical position to the first position of the clause in order to create the special interpretation associated with its noncanonical position and the C-feature. This dependency is illustrated in (7).

(7)	Myy-dä-**kö**$_1$	Pekka	halus-i	__$_1$	koko	omaisuute-nsa?
	sell-A/INF-Q	Pekka.NOM	wanted-PAST.3SG		all	possessions-PX/3SG

This observation motivates in part the term "head movement" used frequently of phenomena of this type in the generative literature. This dependency is regulated by syntactic constraints. For example, it is not possible to extract and move a phonologically complex word out of a subject (8).

(8)	*Myy-dä-kö$_1$	[Peka-n	suunnitelma	__$_1$	auto]	epäonnistui.
	sell-A/INF-Q	Pekka-GEN	plan		car	failed

Intended: 'Was it Pekka's plan to sell (and not to borrow) the car that failed?'

We must capture all restrictions of this type in our model. Notice that the noncanonical position of the dislocated complex word is not the literal first position of the sentence that we could arrive at by counting words. In the example (9)a, the dislocated word is in the fourth position, counting from the beginning.

(9)	a.	Pekka	mietti	että	**myy-dä-kö**	Merja	halua-a	_	omaisuutensa.
		Pekka	wondered	that	sell-A/INF-Q	Merja	want-PRS.3SG		possessions
		1.	2.	3.	4.				

‘Pekka wondered if Merja wanted to sell her possessions.’

Instead, the notion of “first position” refers to a left peripheral position in the CP-cartography that has well-defined syntactic and semantic properties, as discussed in the primary literature (Huhmarniemi 2012; Huhmarniemi and Brattico 2013; Rizzi 1997; Ross 1967; Vilkuna 1989). We can characterize it intuitively as the most prominent syntactic projection of its own finite clause. The pattern seen in Finnish is not crosslinguistically surprising. The first/most prominent position of the finite clause quite typically, if not always, represents discourse and clause typing features of this kind.

The phonologically complex word constitutes a literal complex word in the sense that its parts can occur inside other words as well. We can attach both the A-infinitival morpheme and the Q-morpheme to virtually any verbal stem. A model in which the complex word constitutes an independent atom that cannot be broken down into discrete grammatical constituents does not match with the grammatical system employed by native speakers and would be unacceptable in the present context.

The yes/no morpheme -kO used in the examples above constitutes one left peripheral suffix that is involved in the Finnish C-system and its CP-cartography. There are several others, and the suffixes can be stacked agglutinatively. For example, a word such as *KUKA-ko-han* ‘who.WH.FOC-Q-FAM’ combines an interrogative pronoun (and its *wh*-feature), contrastive focus by prosodic stress, a yes/no particle and a familiarity topic particle, all left peripheral C-features that contribute their own unique semantic effects to the sentence. I have counted more than twenty possible C-features and their combinations, but a detailed study of the complete repertoire is beyond the scope of the present work; twelve features and their combinations were addressed explicitly in this study and appear in the test corpus.[5] Some

[5] The Finnish left peripheral C*-features differ from each other both in syntactic behavior and semantic interpretation, justifying their inclusion into the study. The main points of behavioral divergence are (1) whether the particle can occur in situ (yes, no); (2) whether they affect clause typical and selection at the left peripheral position (yes, no); (3) whether they induce pragmatic root effects

properties of this system are described in Section 3.6 on the basis of (Brattico 2021a; Brattico et al. 2013).

Before we can fit a mathematical formula with the data, the data itself must be presented in some form that is amenable for such treatment. One way to accomplish this is to write all the data of interest – in essence, sentences like the ones just discussed – into an explicit test corpus file and then let the model to process that file, as is done here, or to generate a set which is compared with the test corpus. The test corpus defines what the key linguistic phenomena included into the study are. The test corpus and its contents were described in the main article and will not be repeated here; it had little more than 600 constructions.[6] What was not described in the main article, however, is that there exists a secondary test corpus that contains several basic construction types representing a wide variety of linguistic phenomena and which is involved implicitly in the testing procedure. Its purpose is to make sure that the proposed new mechanisms do not create systematic problems elsewhere or undo earlier results. Currently this secondary test corpus exists in the directory /language data working directory and is called default_corpus.txt. It contains 2000+ construction types (with full glosses for non-native speakers of Finnish). It also contains basic head reconstruction constructions from the present study, so that any future modification to the theory that fails these cases can be subjected to further scrutiny. This procedure ensures that changes to the theory are cumulative, and that a new solution to some problem does not undo previous results in a uncontrolled way.

Based on my (still relative short) experience in preparing and submitting computational studies of this type for publication, the following concern merits a brief comment. Our goal is to provide a mathematical formula that is both sufficient and necessary to capture (calculate, deduce) an explicit linguistic dataset. Because the subject matter of the study is linguistics – that is, the study of human languages – the proposed formula must capture properties of possible human languages. For example, here we are interested in what is possible in Finnish. Conversely, we are not interested in any model

(yes, no). These differences were ignored in this study for reasons of scope, but are examined in detail in a forthcoming paper (Brattico 2021a). See Section 3.6.

[6] Due to recursive rules the real number of constructions the model can handle is of course unbounded. On the other hand, most of these would be trivial expansions of the existing constructions.

that captures the attested expressed in Finnish plus expressions that are neither attested nor possible in Finnish or in any other language. Most of the work done under the umbrella of computational linguistics today is not concerned with the notion of human language in this sense. For example, a model that can "learn" a classification present in a linguistic dataset is virtually always able to "learn" classifications that are never attested in any human language.[7] A model of this type would be irrelevant to linguistic theorizing. In the same way, an engineering parser that parses both human and nonhuman languages, or which does not distinguish grammatical inputs from ungrammatical ones, will be judged as irrelevant. That being said, it is of course possible or desirable that the two approaches converge in the future in a mutually beneficial way.

I conclude this section by describing the above observations in the context of a linguistic theory. The basic grammatical template that we use as a starting point when modelling the Finnish facts is one in which the fronted verb (e.g., *myy-dä-kö* 'sell-A/INF-Q') moves to C^0 from XP in [C^0 [$_{XP}$...]], with XP being some finite projection such as FinP, NegP or TP depending on the theory, mostly irrelevant here (NegP, TP in the present case). There is no robust evidence that other head positions besides C^0 exist between the high complementizer *että* 'that' and the finite projection. Both head and phrasal movement to this position involve the same left peripheral C-features (including prosodic stress) and the same Ā-dependencies, minus few differences (e.g., there are no 'phrasal imperatives' or 'pure wh-heads'). Both head movement and phrasal movement are possible but never both; and never two heads or two phrases. We therefore have, as a null hypothesis, a single [$_{CP}$ Spec [C^0 XP]] configuration that checks left peripheral C-features. Verb movement to C requires that an overt C-feature, a morpheme or prosodic stress, is present; unstressed or unmarked verb-initial clauses are canonically ungrammatical in Finnish.[8] These properties were originally discovered by Vilkuna (1989, 1995), discussed more recently

---

[7] I put the word "learn" inside quotation marks here because what these models often do is that they select one particular model from an extremely narrow set of predefined set of possible models, hence they should be characterized as exhibiting *innate* constraints and limitations rather than any type of "learning." These innate constraints are often so extreme that it trivial to find unbounded number of counterexamples, patterns that are possible in some or all human language that can never be modelled by the proposed "learning" models.

[8] It was claimed in the main article that "Finnish verb-initial clauses express C-features; without C-features, they are ungrammatical," which referred to this generalization. As pointed out in the

by Huhmarniemi (2012), and summarized in (Brattico et al. 2013). As emphasized in the main article, these properties by and large hold true independent of whether we are looking at local or nonlocal head movement. Although there are some differences, the basic template is the same: an element marked for C-feature(s) moves into CP. Thus, independent of whether we perform local or nonlocal V fronting we have the same features, same syntactic position, and the same restrictions.[9] One difference, discussed later in the main article, is that fronting of an auxiliary-like element without lexical content is always HMC compliant, and generates a verum focus interpretation (see, e.g., example 6a in the main article). When a lexical verb is fronted, this verum focus interpretation becomes optional, with the operator interpretation (i.e. the cleft interpretation targeting the predicate, not the whole clause) emerging as an additional and often the favored or possible the only option. This was interpreted as suggesting that the operator (predicate cleft) interpretation, licensing LHM, requires lexical content. There is independent evidence from other languages that lexical content plays a crucial role licensing V fronting (e.g., Roberts 1993).

In sum, then, the general [$_{CP}$ Spec [C^0 XP]] template should be understood as a minimal null hypothesis able to capture the basic facts, including LHM, with departures or complications requiring compelling empirical justification. Whether a more complex cartography is assumed is largely irrelevant to present concerns, however, since the C*-features can only appear on one head or phrase and the argument for Ā head movement does not depend on the details of this particular analysis.

1.3 Syntactic background model

The syntactic background model of the present study captures human linguistic behavior by modelling the information processing steps that occur when informants are presented with linguistic expressions such as (1-9) and are asked to elicit responses, such as grammaticality/acceptability judgments and

footnote, there are exceptional verb-initial clauses, such as pro-drop or existential sentences, but they do not involve V-movement to C and are on such grounds irrelevant.

[9] In the main article it was claimed that both local and nonlocal head movement are always possible for all C-features. Imperatives do not exhibit nonlocal head movement, however. In addition, due to the existence of HMC compliant head movement constructions, a theoretical distinction was later made between operator C*-features and C-features; the claim applies to the former.

semantic interpretations. The approach differs on a superficial level from the more traditional grammatical approaches, which model linguistic competence as a disembodied repository of linguistic knowledge (Chomsky 1965), but the difference is in one sense inconsequential, as the recognition model produces grammaticality judgments and semantic intuitions like any competence model does. In addition, the model generates similar and in many cases identical syntactic representations than those generated by standard cyclic theories. The idea that at least some universal properties of human language(s) could emerge from the comprehension process is not new, however (Cann, Kempson, and Marten 2005; Chesi 2012; Hale 2014; Kempson, Meyer-Viol, and Gabbay 2001; Merchant 2019; Phillips 1996). These ideas, Phillips (1996) in particular, have inspired the present approach.

We begin from the assumption that the human brain utilizes at least three components when interpreting and judging sentences presented via sensory systems: a lexico-morphological component that hosts the lexicon and is responsible for lexical retrieval and morphological decomposition; a syntactic component determining how words are organized in relation to each other; and a semantic component that creates semantic interpretations. It seems impossible to assume anything less, given the type of facts reviewed in Section 1.2. The components are illustrated in Figure 2, with further comments and refinements below.[10]

[10] The terminology used in this document and in the main article for the most part matches with the terminology used in the Python source code. See Section 3.7.

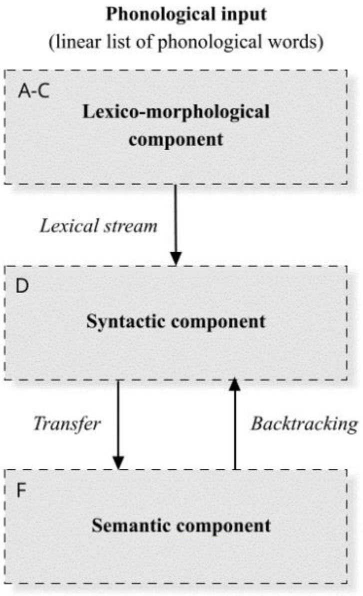


Figure 2. Components of the language faculty as assumed in the syntactic background theory.

The lexico-morphological component retrieves lexical information from the lexicon on the basis of the phonological words it recognizes in the input string. Each word is processed through the lexico-morphological and syntactic modules immediately after it arrives from the sensory systems, thus the model is incremental (Altmann and Kamide 1999; Altmann and Steedman 1988; Demberg, Keller, and Koller 2013; Keller 2010; Konieczny 2000; Marslen-Wilson 1973; Phillips 2003; Tanenhaus et al. 1995). The lexico-morphological module and the syntactic module are connected via *lexical stream,* which sends lexical information ('words and their parts') from the former to the latter in what could be characterized as a linear event sequence. Taking sentence (1) as an example, this component is responsible for recognizing the internal parts of the complex word *myy-dä-kö* 'sell-A/INF-Q' and for forwarding this information to the syntactic component.

How the lexico-syntax interface and the lexicon works is a controversial empirical question that cannot be answered in my view satisfactorily without hypothesizing something and then testing the consequences of that hypothesis. In the present work, lexical entries for morphologically complex words constitute linear sequences of derivational and inflectional elements that for the most part correspond directly to their surface decompositions (*myy-dä-kö* ~ sell#A/INF#Q). Lexical entries for the primitive lexical items (again derivational or inflectional elements) are sets of features without further structure. It is difficult to imagine using anything less while engaging meaningfully with the data presented in Section 1.2. Linguistic features and the primitive lexical items assembled from them constitute the ultimate cognitive primitives of the theory. Two languages can differ in the way they assemble their lexicons, capturing an important source of language-specificity.

The syntactic module attempts to organize the incoming lexical items into a hierarchical configuration. For example, the sentence *John saw the girl with a telescope* has two readings corresponding to two hierarchical arrangements of its lexical items, [*John saw* [$_{\text{DP}}$ *the girl with a telescope*] and [*John* [$_{\text{VP}}$ *saw* [*the girl*] *with a telescope*]]. Lexical items arriving to the syntactic component from the lexico-morphological module (Figure 2) are therefore attached to a partial phrase structure representation in the current syntactic working memory, which represents the syntactic interpretation developed by the algorithm on the basis of words it has seen up to that point.[11] This process is illustrated in Figure 3.

[11] The underlying notion of phrase structure is an asymmetric binary branching bare phrase structure closely related to the one proposed in Chomsky 1995:224ff, 2008. Incoming lexical items are merged from left to right, hence to the right edge of the partial phrase structure. Call the operation Merge^{-1}. A model of this type was first proposed by Phillips (1996, 2003). Suppose the incoming lexical item is β; then Merge^{-1} targets some node α at the right edge of the existing phrase structure and substitutes it with $\gamma = [\alpha\ \beta]$, the latter created by standard cyclic Merge (hence Merge^{-1} = target + Merge + substitute). The label of any phrase is the most prominent primitive lexical item in it. Suppose γ is a syntactic object created by Merge^{-1}, then the label of primitive γ is γ; otherwise suppose $\gamma = [\alpha\ \beta]$ and that if α is primitive, then it is the label; otherwise, if β is primitive it is the label; otherwise we apply the rule recursively to β.

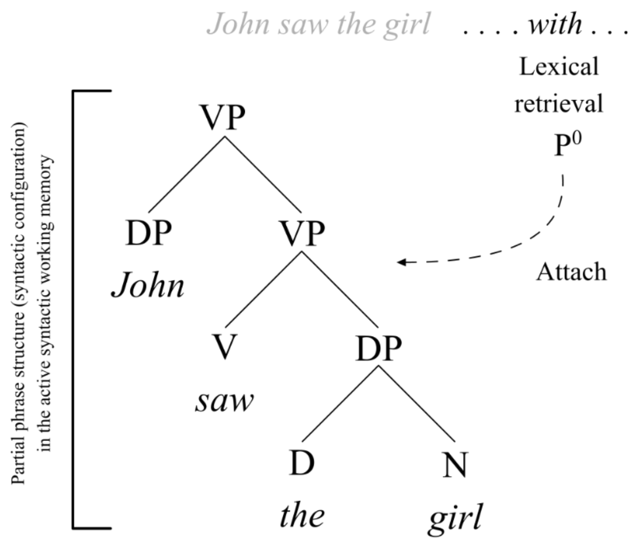


Figure 3. Incremental attachment of incoming lexical items to the current partial phrase structure representation in the active syntactic working memory. The partial phrase structure representation in current syntactic working memory is expanded on its right edge.

Let us assume that if the output of the syntactic component cannot be interpreted, the solution is rejected by the semantic component, the outcome is communicated back to the syntactic component, which then reacts by trying to find alternative solutions by backtracking. This creates garden path effects seen in the case of sentences such as *the horse raced past the barn fell*.[12] The same mechanism detects ambiguities: after finding an interpretation for *John saw* [*the girl with the telescope*] where the girl has the telescope, backtracking allows the syntactic component to find others (namely, one where John had it). Thus, there is a backward pointing arrow "backtracking" in the model in Figure 2 that goes from the semantic system to the syntactic component. When the model backtracks, the operation consumes more computational resources and time, predicting processing delays in the case of real human language comprehension.

[12] The first pass parse [*the horse* [*raced* [*past the barn*]]] + *fell* will be rejected because the last word *fell* cannot be integrated into the existing partial phrase structure. The model backtracks until a well-formed representation ('the horse, which raced past the barn, fell') is found.

We will assume that English-speakers have access to the same cognitive capacities and conceptual representations as Finnish-speakers, and vice versa. If this were not the case, then translation from one language into another would not be possible. The assumption seems to be an implicit or explicit part of virtually all cognitive theories of human reasoning and cognition, and in linguistics too it is commonly assumed that the conceptual systems and the interface between language and meaning "must be universal, in that any thought expressible in a human language is representable in it" (Chomsky 1995:18). Sensorimotoric properties of linguistic expressions do, however, vary depending on language. Some properties of (1-9) are not attested in English. The model must involve a component which sanitizes the input from language-specific sensorimotoric noise. That process is called *transfer*. It takes place when a syntactic object is sent form the syntactic component to the semantic system for interpretation; see the arrow "transfer" in Figure 2. Properties of transfer are crucial for our analysis of (1-9) and will be elucidated later.

If the input cannot be mapped into any legitimate syntactic object, the model will judge it ungrammatical. If the input is judged grammatical, then one or several syntactic analyses are generated which are linked with semantic interpretations, both which are checked for correctness by comparing them with native speaker intuitions. I will explain how this verification is performed later in this document. In addition, if we were interested in psycholinguistic realism, we would compare the performance properties of the model with those collected from native speakers in controlled experiments; this aspect will be elucidated briefly in Section 3.1.

The model must be able to decide if the words in the input match with their grammatical contexts created in the syntactic component. The following variations of (1), in which we use the VA-infinitival instead of the A-infinitival, are ungrammatical.

(10) a.	*Myy-**vän**-kö	Pekka	halusi	__	koko	omaisuutensa?	
	see-VA/INF-Q	Pekka	wanted		all	possessions	
b.	*Pekka	halusi	myy**-vän**	koko	omaisuutensa.		
	Pekka	wanted	sell-VA/INF	all	possessions		

Intuitively the words do not “match” with each other, either syntactically, semantically, or perhaps by a mixture of both criteria. We therefore assume that the primitive lexical items coming out of the lexico-morphological system contain lexical features defining what their admissible local grammatical contexts are. For example, the verb *halusi* ‘wanted’ contains a lexical selection feature which allows it to select for a phrase headed by an A-infinitival (the corresponding English selection rule would be ‘want + to sell’). We can then determine that some phrases *cannot* occur in this position by using negative selection features or by assuming that only positively selected phrases are possible. For example, the finite verb ‘want’ cannot select for a control VA-infinitival (**Pekka halusi* ***myy-vän*** *omaisuutensa* ‘Pekka wanted sell-VA/INF possessions’). The lexicon is provided as an external parameter to the simulation, currently in three separate files which contain the primary language-specific lexicon, universal lexical redundancy rules, and a third file containing universal lexical elements or grammatical-functional morphemes. The lexicon is discussed further in Section 3.5. These three sources are synthesized into one lexicon at runtime, and separate lexicons are created for the speakers of each language. The Finnish agglutinative profile is modelled by assuming that the entries in this list can have complex internal structure. The entry for the verb *myy-dä-kö* ‘sell-A/INF-Q’ in the example (1), for example, contains four morphemes: the verb stem, a transitivizer morpheme v, the A-infinitival morpheme, and the yes/no particle. Each then points to another entry in the same lexicon which, if primitive, contains only a lexical feature set and does not decompose further.[13]

The question of whether the lexical features that participate in the process of determining admissible grammatical contexts for words are semantic or syntactic (or most likely, both) has been controversial. In a fully computational model any feature that plays a role in the system must have a formal shape which ultimately determines what it does; there are no immaterial semantic features that have no formal properties. A semantic feature has both syntax and semantics, while a purely syntactic feature is invisible outside of the syntactic processing pathway. Both types of features exist in the model.

[13] The lexico-morphological component must be able to perform morphological parsing. Morphological parsing has not been implemented at present writing, but the user can simulate it by providing morphological decompositions directly in the input. Thus, the model can read both *admires* and *admire#3p#sg*.

We can imagine at least three ways of interpreting a computational model of this type, following an influential proposal by Marr (1982). One is to consider the computational algorithm as a research tool that is employed to verify whether some proposed hypothesis concerning grammatical competence is correct. This corresponds to Marr's notion of an "abstract computational theory," in which "the performance of the device," here the human brain, "is characterized as a mapping from one kind of information to another" (p. 24), in present case from sensory inputs into grammaticality judgments and semantic interpretations. Accordingly, the "grammar as a whole can thus be regarded as, ultimately, a device for pairing phonetically represented signals with semantic interpretations, this pairing being mediated through a system of abstract structures generated by the syntactic component" (Chomsky 1964:9–10). This would represent a pure competence theory. It describes what is possible in human language. Marr proposed however that we go further and also consider the representations and algorithmic transformations the system uses in performing the mapping, and further how the algorithm is implemented in the human brain. These constitute the two lower levels of description in Marr's theory, the algorithmic level and the implementation level. Using the former, we consider that the components present in Figure 2 describe real information processing pathways activated in the human brain during language comprehension. Pursuing an implementation theory would require all functions in the formal model to be written in such a way that they would also constitute a model of the underlying brain physiology. My own interpretation of the background theory identifies it with Marr's first and second levels but not with the third: it is written as a literal model of human language comprehension that respects (or is intended to respect) competence and performance data, but remains agnostic concerning neuronal implementation whose properties are, in my view, still beyond rigorous computational modeling when it comes to the calculation of linguistic data.[14]

[14] Intuitions concerning physiological interpretation can guide the development of the algorithm. My own view is that the implementation will rely on linear list-like structures, temporally evolving neuronal signals, that can be nested hierarchically. This hypothesis is visible in many of the fundamental assumptions and implementation choices in the model.

2 HEAD RECONSTRUCTION ANALYSIS

2.1 Overview

Let us consider the head reconstruction analysis proposed in the main article. The background model elucidated in the previous section can process phonologically complex words and was used in one published study (Brattico and Chesi 2020). The model attaches incoming lexical components automatically into a phrase structure in the order they appear in the input and are transmitted to the syntactic component via lexical stream (11). Inflectional morphemes, such as the third person singular feature '3sg', are inserted as features inside adjacent morphemes during lexical streaming.

(11) a. John + admires + Mary (Input)

b. John admire-#v#T#3sg Mary (Lexical decompositions)

c. John, 3sg, T, v, V, Mary (Lexical stream)

c. [John [$T_{[3SG]}$ [v [V Mary]]] (Attachment, i.e. Merge^{-1})

This model corresponds to a hypothesis in which complex phonological words are created and processed inside the lexico-morphological module (or what is called "phonology" in the standard theory) and are invisible to the syntactic and semantic components. This idea was discussed briefly in Chomsky (2001) and then developed in further work (Platzack 2013; Schoorlemmer and Tanja Temmerman 2012). The model produces useful results in a range of cases. It was noted immediately during the abovementioned study however that this assumption runs into problems with standard head movement constructions such as English aux-inversion. The calculated output in such cases is (12).

(12) Does + John + admire + Mary?

[T [John [v [V Mary]]]

The fronted auxiliary will discharge its internal lexical constituents to the highest position in the clause, matching its position in the left-to-right order. One problem is that the EPP-feature of T remains unchecked. These cases were handled by ad hoc rules,[15] but the approach is insufficient and quite likely incorrect due to the Finnish long head movement constructions (1), repeated here as (13).

(13)	Myy-dä-kö$_1$	Pekka	halus-i	__$_1$	koko	omaisuute-nsa?
	sell-A/INF-Q	Pekka.NOM	wanted-PST.3SG		all	possessions-PX/3SG

‘Was it selling that Pekka wanted to do to all his possessions?’

Discharging the elements of the first word to the beginning of this clause produces gibberish, and it is also clear that some of these internal components should be in the position marked by the gap. This observation formed the background for the present study. Empirical investigation of (13) showed furthermore that the operation that dislocates the word to the first position of the clause follows syntactic constraints. The sentence is, furthermore, interpreted so that the speaker has specifically targeted the lexical content of the complex predicate ‘to sell’ for a special left peripherical yes/no interpretation (‘is it selling…’), further pointing towards the possibility that the language faculty has access to a syntactic-semantic notion of ‘complex predicate’ that can be targeted for a cleft interpretation independently of its possible role within some larger phrase. Standard phrase structure objects created by merge or attachment presuppose, however, that the individual lexical nodes constitute separate phonological words, and that merging of two such units generates phrases.

Looked from this vantage point, the crucial empirical question then becomes whether we could capture all these facts without assuming some novel notion of ‘complex head’ in the theory. In the phonological model described above no such thing was needed, so the question had considerable weight. One possibility, discussed in the main article, is that we separate head movement from morphological decomposition (Harizanov and Gribanova 2018) and try to do this in a way that does not require postulation of complex heads as special grammatical objects. This could be accomplished for example by manipulating the left peripheral C-head and its noncanonical position exceptionally as one

[15] One solution is to reconstruct the postverbal subject upwards into SpecTP to check the EPP. This requires an operation that moves phrases from their canonical positions into formal EPP positions, in traditional terms. A second possibility is to relativize the EPP requirement, perhaps by nullifying it under some circumstances. Yet, the EPP principle seems to be operational in these verb-initial clauses: they do not become ‘subjectless’.

unit while handling the verbal morphological decomposition in phonology (as was done in the earlier version, thus *myy-dä-* 'sell.v-A/INF' would generate [A/inf [v V]]). Due to the empirical reasons mentioned in the main article, and for other reasons that have to do with the reverse-engineered architecture itself, I was not able to develop this idea into an elegant algorithm.[16] Still, it remains within the realm of empirical possibility.

If the language faculty has access to a notion of complex predicate or complex head, then there must exist a syntactic configuration that does not behave like a syntactic phrase. It should be pointed out that this conclusion is theoretically costly, requiring an enriched phrase structure theory. In addition, several possible solutions that can achieve the same outcome exist in the literature or could be easily imagined, some of which are more costly than others. Yet, it is hard to decide between them unless we run simulations. I therefore decided to adopt an engineering approach in order to experiment with complex heads. This is the rationale for using the theory-neutral notation X(Y) for a complex head X that contains another head Y, leaving the implementation open for further experimentation and implementation.[17]

To add this option to the model, then, the notion of syntactic phrase was first defined so that a constituent K is considered a *syntactic phrase* if and only if it has exactly two daughter constituents and a *primitive nonphrasal constituent* otherwise. A syntactic phrase is, by definition, an object that is in the domain of phrasal syntactic operations. A constituent that has no daughter constituents is a primitive

[16] Discussion of this issue would merit its own article. One relevant concern is that once we examine the computational rule required for the reconstructing the complex A-infinitival head (e.g., 43A-B in the article), it is easy to see that the reconstruction operation generalizes automatically to the case of morphological decomposition and to the local T-to-C movement. Still, there are alternatives. One alternative is to enrich noncanonical heads with phonological or morphological information but keep them syntactically simple; this solution did not work. Another is to extract heads with noncanonical positions first into wrong positions and then reconstruct, perhaps independently. The computational operations required to do this looked extremely unnatural and complex.

[17] The standard assumption at present writing seems to be a system in which complex heads are ordinary phrase structure objects specifically marked for being heads. Thus, Chomsky 1995:225 posits a category of "terminal elements" that can be feature bundles or phrasal objects, corresponding to complex heads in the present model. Thus, on such assumptions [$_{\text{DP}}$D N] differs from [$_{\text{N}}$D N], [$_{\text{D}}$ D N]. I found this proposal impossible to implement, leading to complications and empirically incorrect outcomes. It remains within the realm of empirical possibility and can be tested in some future study. This approach taken in the current study was inspired by (Roberts 1993).

nonphrasal constituent, but it has a further property of being a *terminal constituent*. This leaves room for a constituent that has only one daughter, which makes it a primitive nonterminal. These entities were used to represent complex heads. They are designated by X(Y), Y being the sole daughter of X. The rule applies iteratively, allowing the algorithm to wrap a linear sequence of heads inside another head.[18] This is notated by writing X(Y, Z, ...). Word-internal lexical items are attached to the structure in this way when they arrive from the lexico-morphological component to the syntactic component. The result is (14). This should be compared with (11).

(14)	John	+	admires	+	Mary	(Input)
	John,		T, v, V,		Mary	(Lexical stream)
	[John		[T(v, V)^0		Mary]]	(Attachment)

The result differ markedly from the earlier model, since now the first phrase structure object that arrives to syntax is much like the surface string itself. It is compressed and even anomalous, in that most syntactic theories do not recognize it as a well-formed syntactic structure. But, having admitted T(v, V)^0 into phrase structure, we can consider reconstruction. Head reconstruction is executed during transfer (see Figure 2). The algorithm travels downstream on the right edge of the phrase structure in the active working memory, following labelling and selection, targets complex heads X(Y) and reconstructs Y downstream into a position [H [Y…]] where it can be selected from above by H based on lexical features, adopting the first and hence most local solution available.[19] If no position is available, last resort solution [X [Y…]] is adopted. X remains in situ, thus in the position where it was

[18] This notation makes room for primitive constituents that contain phrasal constituents. This was originally considered a potentially useful way of modelling clitics and incorporation. Perhaps it could be used to redefine the notion of specifier, so that [XP [H YP]] could be represented as [H(XP)^0 YP]. There are, however, no lexical or phrasal syntactic operations in the current implementation that have access to word-internal phrasal constituents.

[19] If X(Y) is the bottom right constituent, it is reconstructed as [X Y]. An additional edge case is presented by a situation in which the head needed for selection resides *inside* the bottom right node. This was solved by reconstructing the bottom right node before the reconstructing higher head is fitted into the structure.

merged during left-to-right parsing or backtracking.[20] The algorithm then continues downwards. Thus, if Y is itself a complex head, it will be targeted later. All complex heads are reconstructed, in the manner shown in (15).[21] Q will remain at the left periphery, representing the fact that the sentence has interrogative force. It is also used for representing the propositional scope of the operation, as elucidated in Section 2.5.

(15)	Myy-dä-kö	Pekka	halus-i	_		koko	omaisuute-nsa?
	Q(__)	Pekka.NOM	wanted-PAST.3SG	A/INF(__)	V	all	possessions-px/3sg

Once these mechanisms are in place it is possible to perform calculations by feeding the model with sentences with phonologically complex words.

If we assume this model as a starting point and look at it from an inverse point of view, the system appears to be doing syntactic compression. The inverse of (15) removes both heads *and phrasal positions* from the structure. In (15), for example, the spellout structure will have ‘all possessions’ at the (wrong) complement position of ‘want’ due to grammatical compression, and a much simpler object is submitted to the sensorimotoric interface than would be the case if these operations were not performed. An alternative interpretation is that complex predicates are natural kinds also for the larger cognitive theory, but this requires that we give LF interface access to them or create a syntax-semantics interface between spellout and some semantic component.

I will present one additional comment on the latter idea. Some amount of skepticism, most of it justified in my view, exists in the literature against giving UG access to complex heads. My own first attempt at using this model for grammatical analysis was based on the hypothesis that complex heads

[20] This means that the model cannot process null or idle head movement, in which (stated from the point of view of a standard cyclic theory) a head moves without merging with a higher head that has independent syntactic and/or semantic content.

[21] There is no axiom which prevents complex heads from occurring at the syntax-semantic interface. If a complex head X(Y) survives reconstruction, Y will behave at the syntax-semantics interface as if it were not present at all and the representation crashes because some selection rules involved with Y cannot be satisfied and/or because other reconstruction rules, namely those which depend on Y, fail. I will later discuss one edge case where this occurred.

exist only inside the lexico-morphological system (“phonology”). The idea was to break complex words into their constituent parts inside the lexico-morphological component, perhaps as a consequence of some simple perceptual mechanism, and then feed these parts into syntax in some order that would have resulted plausible phrase structure objects. But perhaps this is a misjudgment, or an accidental feature of focusing on relatively isolative Indo-European languages. Finnish is a polysynthetic and agglutinative language, and thus if we examined this language group more closely we could potentially find that the complex head/complex predicate strategy is as natural and widespread as the isolative one. Thus, perhaps complex heads do represent grammatical natural kinds in the same way as phrases do. Still, for a substantial range of cases the output of head reconstruction is identical to the earlier model in which lexical decomposition was performed inside the lexico-morphological component. This makes these concerns still relevant.

2.2 Simple sentences

We will begin the more detailed examination by looking at the derivation of simple sentences such as English transitive clause *John admires Mary* (sentence number #84 in the test corpus and output). To get a sense of how the model calculates the data, we examine the detailed derivational log file that contains a record of the linguistically relevant computational steps executed during the processing of the sentence. Figure 5 shows the initial part of the derivation (the file is available online).

```
22395 #84. John admires Mary
22396 ['John', 'admires', 'Mary']
22397
22398      1. ['John', 'admires', 'Mary']
22399
22400        Next item: "John". Lexical retrieval...(45ms) Word "John-#D#sg#3p#def#hum#m" contains multiple morphemes ['John-', 'D$', 'sg
22401        Next item: "hum$". Lexical retrieval...(45ms) Inflectional feature hum$...(50ms) Added ['LANG:EN', 'PHI:HUM:HUM'] into memory
22402        Next item: "def$". Lexical retrieval...(45ms) Inflectional feature def$...(50ms) Added ['LANG:EN', 'PHI:DET:DEF'] into memory
22403        Next item: "3p$". Lexical retrieval...(35ms) Inflectional feature 3p$...(40ms) Added ['LANG:EN', 'PHI:PER:3'] into memory...
22404        Next item: "sg$". Lexical retrieval...(35ms) Inflectional feature sg$...(40ms) Added ['LANG:EN', 'PHI:NUM:SG'] into memory..
22405        Next item: "D$". Lexical retrieval...(25ms) Adding inflectional features ['LANG:EN', 'PHI:DET:DEF', 'PHI:GEN:M', 'PHI:HUM:HU
22406        Item enters active working memory. Wiring semantics [1] for D...
22407
22408        Next item: "John-". Lexical retrieval...(55ms) Done.(60ms)
22409        Item enters active working memory.
22410
22411     2. Consume "John": D + John
22412        One solution due to sinking.(65ms) Done.
22413        Results:
22414                (1) D
22415        Sinking John into D = D(N)(70ms)
22416        Next item: "admires". Lexical retrieval...(75ms) Word "admire-#v#T/fin#[-s]" contains multiple morphemes ['admire-', 'v$', '
22417        Next item: "T/fin$". Lexical retrieval...(65ms) Adding inflectional features ['LANG:EN', 'PHI', 'PHI:GEN:F', 'PHI:GEN:M', 'P
22418        Item enters active working memory.
22419
22420     3. Consume "T": D(N) + T
22421        Working memory operation...1 nodes currently in active memory.
22422        Filtering and ranking merge sites...Filtering...Done. Ranking...Bottom-up baseline ranking...+Spec selection for D(N)...(100
22423        Results:
22424                (1) D(N)
22425        Now exploring solution [D(N) + T]...Transferring left branch D(N)...(80ms) Result: [[D John] T]...Done.
22426
22427        Next item: "v$". Lexical retrieval...(25ms) Done.(30ms)
22428        Item enters active working memory.
```

Figure 5. Screenshot from the initial part of the derivational log file containing the computations for sentence *John admires Mary*. See the main text for explanation.

Each numbered step represents the processing of one lexical item. For example, *John* is decomposed into John#D (N#D) in the lexicon, followed by a list of inflectional features such as 'singular' or 'human' (lines 22400-22405) that we can ignore here but which are involved in the agreement system and in the creation of referential indexes and the semantic discourse inventory hosting the entities the sentence "talks about." The D-element arrives to syntax on line 22406, at which point it already contains all inflectional features inserted into it during lexical streaming (line 22405). It is followed by the noun stem (line 22411). At this point, the active working memory contains D, while N is waiting in the queue to be attached. Since D and N were inside the same phonological word, N will be attached inside D = D(N) (lines 22412-22415). The operation is called "sinking" at the level of the source code.[22] Sinking was added to the model during this research project, being required by the complex head system. The

[22] To recapitulate, phrasal Merge-1 targets α and β and yields [α β], whereas sinking yields [$_{\alpha}$ β]. The theoretical cost associated with the latter is obvious at the level of source code: we must write separate code for phrasal merge and sinking when both operations are possible. This is undesirable and should eventually by replaced by code that handles both as special cases of some more abstract operation.

rest follows the same formula: an item arrives, the syntactic module considers how to attach it, and the iteration continues until all words have been consumed and attached. After all words have been consumed, the syntactic component ends up with (16) (line 22467).

(16) John admires Mary

[D(N)^0 [T(v, V)^0 D(N)^0]]

This can be considered a "first pass parse," a solution the parser considers the most plausible, after reading all words once. It is a fairly transparent and simple syntactic representation of the input sentence. This transparency results from the mechanism that creates complex heads, in that now phonological words at the sensory level are mirrored by complex heads inside syntax. Solutions generated by the left-to-right attachment operation are called *spellout structures* because they are transparently related to sensory objects (i.e., the original linear input string). Spellout structure are also highly 'compressed' structures due to the limited amount of sensory information available during the first steps of the derivation.

There is another possible solution, namely γ = [$_{DP}$ [$_{VP}$ *John admires*] *Mary*], which the model rejects. All possible solutions are ranked in terms of their psycholinguistic plausibility, and the rankings are used during backtracking. The ranking is based on psycholinguistic heuristic principles that the user can activate or inactivate. All heuristic principles currently available were used in the simulations reported in this study. If we examine however what happened during the generation of γ, we find that the parser rejected this solution as impossible (line 22453), thus it was not inserted into the search tree. This is because [$_{DP}$[$_{VP}$ *John admires*] *Mary*] is so ill-formed that it can be rejected locally: there is no possible continuation that could render it useful. When a parsing path is closed off locally, we say that it is *filtered*. Filtering and ranking are the two mechanisms the parser uses to navigate through possible derivations. Filtered solutions cannot be targeted later by backtracking; they are closed off permanently. Once a solution has been generated, however, it will be sent off to the semantic component for evaluation via transfer. These operations are visible in the derivational log file and shown in Figure 6.

```
22467  Trying spellout structure  [[D John] [T(v,V) D(N)]]
22468  |   Checking surface conditions...Done.
22469  |   Transferring to LF...(75ms)
22470  |   |   |   1. Head movement reconstruction...Reconstruct v(V) from within T(v,V)...(80ms) =[[D John] [T [v(V) D(N)]]]...Reconst
22471  |   |   |   |   = [[D John] [T [v [admire [D Mary]]]]](90ms).
22472  |   |   |   2. Feature processing...Done.
22473  |   |   |   |   = [[D John] [T [v [admire [D Mary]]]]](90ms).
22474  |   |   |   3. Extraposition...Done.
22475  |   |   |   |   = [[D John] [T [v [admire [D Mary]]]]](90ms).
22476  |   |   |   4. Floater movement reconstruction...Done.
22477  |   |   |   |   = [[D John] [T [v [admire [D Mary]]]]](90ms).
22478  |   |   |   5. Phrasal movement reconstruction...(95ms) (100ms) Done.
22479  |   |   |   |   = [[D John]:2 [T [__:2 [v [admire [D Mary]]]]]](100ms).
22480  |   |   |   6. Agreement reconstruction...(105ms) (105ms) "T" acquired PHI:GEN:M by Agree-1 from __:2 inside its sister..."T" ac
22481  |   |   |   |   = [[D John]:2 [T [__:2 [v [admire [D Mary]]]]]](105ms).
22482  |   |   |   7. Last resort extraposition...(110ms) LF-interface test...Done.
22483  |   |   |   |   = [[D John]:2 [T [__:2 [v [admire [D Mary]]]]]](110ms).
22484  |   Done.
22485  |   LF-legibility check...Checking LF-interface conditions...(115ms) LF-interface test...
22486  |   Transferring [[D John]:2 [T [__:2 [v [admire [D Mary]]]]]] into the conceptual-intentional system...
22487  |   Resetting semantic interpretation..."v" with ['PHI:DET:_', 'PHI:NUM:_', 'PHI:PER:_'] was associated at LF with 1. [D John]
22488  |   Wire narrow semantics...Done.
22489  |   Computing attitude semantics and information structure...Done.
22490  |   Solution was accepted at 1125ms stimulus onset.
22491  |
22492  |   Semantic interpretation:
22493  |   Recovery: ['Agent of v(John)', 'Patient of admire(Mary)']
22494  |   Aspect: []
22495  |   D-features: []
22496  |   Operator bindings: []
22497  |   Speaker attitude: ['Declarative']
22498  |   Information structure: {'Marked topics': [], 'Neutral gradient': ['1', '2'], 'Marked focus': []}
22499  |
22500  |   ------------------------------------------------------------------------------------------------------------------------
22501  |   Solution:       [[D John] [T [[D John] [v [admire [D Mary]]]]]]
22502  |   Grammar:        [[D John]:1 [T [__:1 [v [admire [D Mary]]]]]]
22503  |   Spellout        TP = [DP:1 [T [__:1 [v [V [D N]]]]]]
22504  |   ------------------------------------------------------------------------------------------------------------------------
```

Figure 6. Screenshot from the derivational log file containing transfer. The candidate solution is transferred to the semantic component (syntax-semantic interface), where it is evaluated (and in this case, accepted). Transfer involves several operations, numbers from 1-7 on lines 22470-22483, including the head reconstruction algorithm proposed in this study (line 22470).

The result is that all complex heads reconstruct (line 22471) and the grammatical subject is reconstructed to its canonical thematic position SpecvP (lines 22478-9).[23] The analysis returned by the model is (17), also shown in Figure 7. Notice that phrasal labels are invisible in the output; they are determined dynamically by the labeling algorithm.

(17) [$_{TP}$ [$_{DP}$ D *John*]$_{1}$ [$_{TP}$ T [$_{vP}$ _$_{1}$ [$_{vP}$ v [$_{VP}$ *admire* [$_{DP}$ D *Mary*]]]]]]

[23] By A-reconstruction, which represents inverted A-movement. An algorithmic for A-movement has proven elusive. It is easy to formulate principles that work in connection with limited datasets but fail when tested with more representative samples. This difficulty stems in part from the fact that I have Finnish sentences in the larger datasets while Finnish A-movement remains largely unexplored (no published studies exist, to my knowledge).

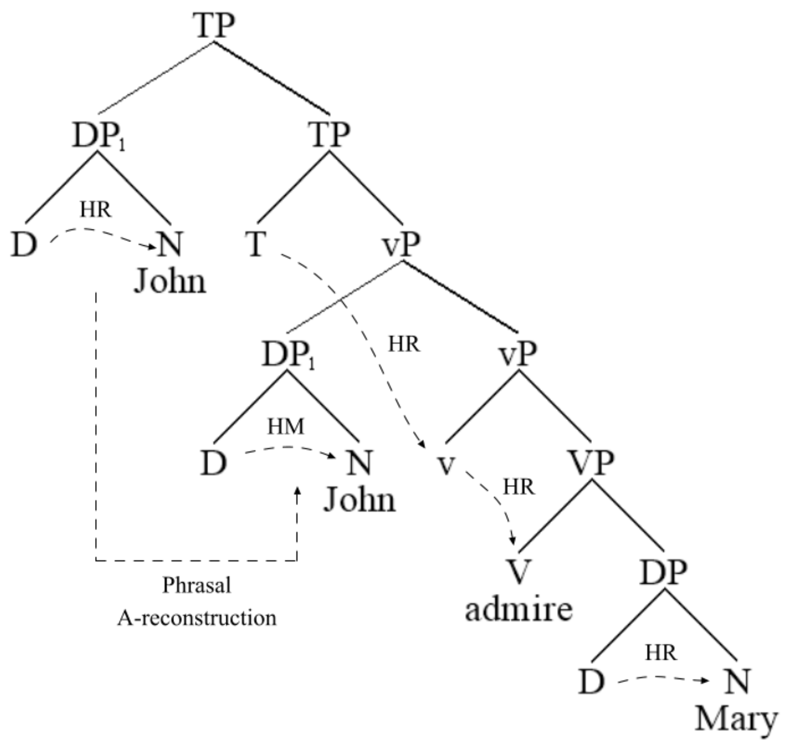


Figure 7. Calculated output for *John admires Mary*, HR = head reconstruction.

We will examine some of the details in Section 4; here I will focus on matters related to head reconstruction. Early experiments with this system revealed an interesting and unanticipated problem that concerns the way the parser processes complex heads. The matter is not directly relevant to the grammatical analysis but is relevant to the question of how a psycholinguistically realistic parser processes complex heads and what type of cognitive architecture supports these operations. Any solution [X(Y) + Z] branches into two possible derivations (a) [[X Y] Z] and (b) [X [Y Z]]. Nominals such as DP in general follow the former template (e.g., [[$_{DP}$ *John*] T...]) while verbal/predicate spreads the latter ([T [V...]]). The problem is that transfer is a one fell swoop cognitive reflex that does not branch. This required that some amount of parsing intelligence had to be added to the head reconstruction procedure. A left branch expansion (a) is created immediately upon [X(Y) + Z] by applying head reconstruction to X(Y), while the right edge expansion (b) is performed later when the same algorithm is applied to the hosting phrase structure object. The choice depends on whether the outcome generates something plausible. This is necessitated by the fact that head reconstruction can be nonlocal: when it is, we must leave X(Y) into the structure and reconstruct it when the required structure is available later.[24]

The model reacts automatically to all overt elements in the input. For example, in Finnish the sentential negation is marked by an auxiliary-like negative particle *e-* that occurs above the tense verb and exhibits full phi-agreement with a grammatical subject, if present (18). This sentence is number #8 in the simulation output.

(18) Pekka e-i ihaile Merja-a. (#8)

Pekka.NOM not-3SG admire Merja-PAR

'Pekka does not admire Merja.'

The output of the model is illustrated in Figure 8. Notice that the model will respond automatically to the presence of the negative word *e-* 'not' in the input by applying the same mechanisms as elucidated above: the word is retrieved from the lexicon and its components are forwarded to the syntactic module in the order they appear in the input and in the lexical decomposition (if any).

[24] The mechanism fails when both cases are possible (e.g. D(N) + P where P could attach either as complement to N or as an adjunct). This issue was corrected in a later study by inserting both choices (a-b) directly into the search tree, but this model was not used in the calculation of the present dataset.

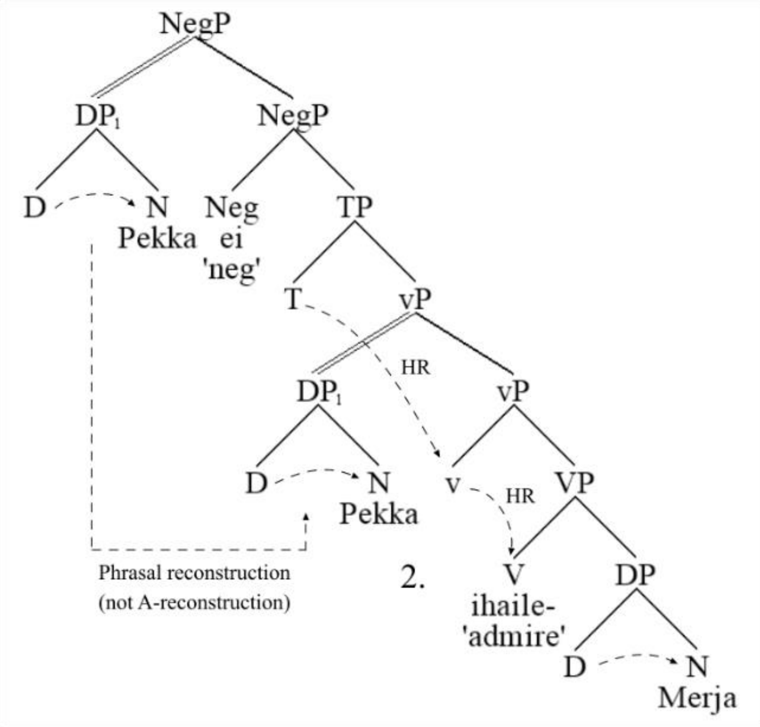
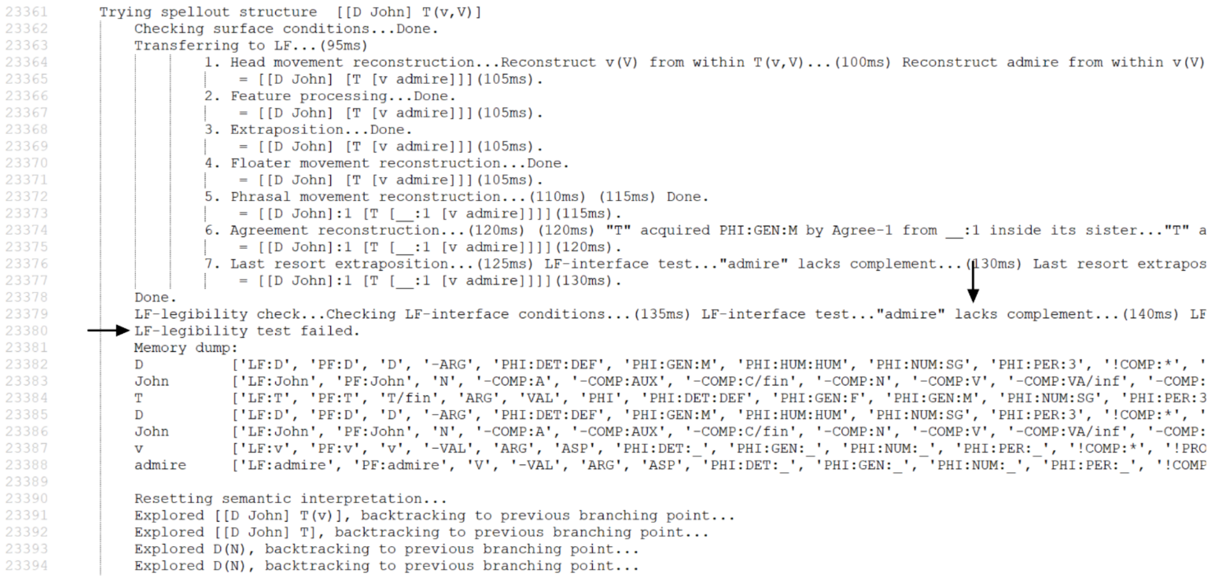


Figure 8. Output for Finnish negative clause.

The algorithm generates the negative head *e-i* 'not.3SG' into the correct position on the basis of its occurrence and position in the surface string. The grammatical subject is reconstructed directly into SpecvP, while the finite verb is reconstructed into the verbal spread T-v-V. Auxiliaries, both in Finnish and English, are treated in the same way. Notice that if we inserted the negation into some wrong position in the input, it would get merged-1 to a wrong position in the spellout structure, could not be selected at LF interface, thus the derivation would correctly crash. Ungrammatical inputs do not produce output solutions, but they leave computational steps into the derivational log file that contain reasons for rejection. For example, sentence #88 = *John admires* = [*John* [T [v V]]] is rejected at the syntax-semantics interface because the transitive verb V lacks a mandatory DP complement (line 23379 in the log file) and no other solutions are found. See Figure 9.



```
Trying spellout structure  [[D John] T(v,V)]
    Checking surface conditions...Done.
    Transferring to LF...(95ms)
            1. Head movement reconstruction...Reconstruct v(V) from within T(v,V)...(100ms) Reconstruct admire from within v(V)
            |   = [[D John] [T [v admire]]](105ms).
            2. Feature processing...Done.
            |   = [[D John] [T [v admire]]](105ms).
            3. Extraposition...Done.
            |   = [[D John] [T [v admire]]](105ms).
            4. Floater movement reconstruction...Done.
            |   = [[D John] [T [v admire]]](105ms).
            5. Phrasal movement reconstruction...(110ms) (115ms) Done.
            |   = [[D John]:1 [T [__:1 [v admire]]]](115ms).
            6. Agreement reconstruction...(120ms) (120ms) "T" acquired PHI:GEN:M by Agree-1 from __:1 inside its sister..."T" a
            |   = [[D John]:1 [T [__:1 [v admire]]]](120ms).
            7. Last resort extraposition...(125ms) LF-interface test..."admire" lacks complement...(130ms) Last resort extrapos
            |   = [[D John]:1 [T [__:1 [v admire]]]](130ms).
    Done.
    LF-legibility check...Checking LF-interface conditions...(135ms) LF-interface test..."admire" lacks complement...(140ms) LF
    LF-legibility test failed.
    Memory dump:
    D          ['LF:D', 'PF:D', 'D', '-ARG', 'PHI:DET:DEF', 'PHI:GEN:M', 'PHI:HUM:HUM', 'PHI:NUM:SG', 'PHI:PER:3', '!COMP:*', '
    John       ['LF:John', 'PF:John', 'N', '-COMP:A', '-COMP:AUX', '-COMP:C/fin', '-COMP:N', '-COMP:V', '-COMP:VA/inf', '-COMP:
    T          ['LF:T', 'PF:T', 'T/fin', 'ARG', 'VAL', 'PHI', 'PHI:DET:DEF', 'PHI:GEN:F', 'PHI:GEN:M', 'PHI:NUM:SG', 'PHI:PER:3
    D          ['LF:D', 'PF:D', 'D', '-ARG', 'PHI:DET:DEF', 'PHI:GEN:M', 'PHI:HUM:HUM', 'PHI:NUM:SG', 'PHI:PER:3', '!COMP:*', '
    John       ['LF:John', 'PF:John', 'N', '-COMP:A', '-COMP:AUX', '-COMP:C/fin', '-COMP:N', '-COMP:V', '-COMP:VA/inf', '-COMP:
    v          ['LF:v', 'PF:v', 'v', '-VAL', 'ARG', 'ASP', 'PHI:DET:_', 'PHI:GEN:_', 'PHI:NUM:_', 'PHI:PER:_', '!COMP:*', '!PRO
    admire     ['LF:admire', 'PF:admire', 'V', '-VAL', 'ARG', 'ASP', 'PHI:DET:_', 'PHI:GEN:_', 'PHI:NUM:_', 'PHI:PER:_', '!COMP

    Resetting semantic interpretation...
    Explored [[D John] T(v)], backtracking to previous branching point...
    Explored [[D John] T], backtracking to previous branching point...
    Explored D(N), backtracking to previous branching point...
    Explored D(N), backtracking to previous branching point...
```

Figure 9. A screenshot from the derivational log file containing part of the derivation for sentence **John admires*.

If we provided in the lexicon that *admires* is either an intransitive verb or ambiguous between transitive and intransitive readings, a solution would be found.

This is rather important feature, in that many or most ungrammaticality judgments result from a situation in which the transfer operation is too “unintelligent” and unable to reconstruct an element from a wrong position into the correct one, even when the grammatical distance between the two positions is small. Ultimately this stems from the fact that the minimal search algorithm that performs most reconstruction is limited in what it can do.

2.3 CP cartography mechanics

In most of the data examined in this work, the predicate moves to the left peripheral CP-domain while carrying a grammatical feature (or several) expressing one of the special interpretations associated with these operations. In the example (19), this involves yes/no interrogativization.

(19) Myy-dä-**kö**	Pekka	halus-i	_	koko	omaisuute-nsa?
sell-A/INF-**Q**	Pekka.NOM	wanted-PST.3SG		all	possessions-px/3sg

‘Was it selling that Pekka wanted to do to all his possessions?’

We examine how the analysis calculates sentences of this type. I will focus on syntax and discuss semantics in a separate subsection 2.5, a more detailed discussion of the C-features involved in the mechanism (largely ignored in this study) are elucidated in Section 3.6. We begin by considering a simple sentence (20), #102 in the output.

(20)	IHAILE-E	Pekka	Merja-a!
	admire-3SG.FOC	Pekka.NOM	Merja-PAR

’Pekka DOES admire Merja!’ or

’Pekka ADMIRES Merja!’

The fronted finite verb has been stressed prosodically to emphasize the contrastive focus reading (‘Pekka admires, not hates, Merja’ or ‘Pekka DOES admire Merja’). Prosodic information is transmitted to the syntactic component by prosodic features, which are (due to the lack of real prosody in the text files) attached to the input string as inflectional features. This is of course a simplification. The calculated output for this input is shown in Figure 10.

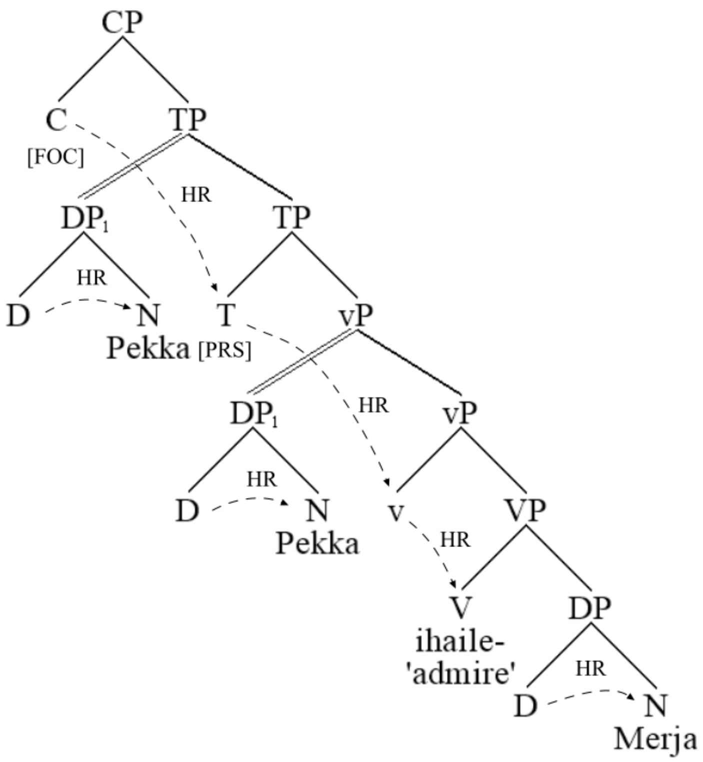


Figure 10. Calculated output for a simple T-to-C reconstruction

The head reconstruction algorithm leaves the highest head in situ and reconstructs the rest. The first step in the derivation is therefore (21)a-b.

(21) a.	Ihailee.foc	Pekka		Merja-a!
	admires-FOC	Pekka.NOM		Merja-PAR
b.	C_{foc}(__)	[Pekka	T(v, V)	Merja-a]

This will eventually result in the phrase structure shown in Figure 10, in which the clause is headed by a contrastive focus C-element. The mechanism correctly rejects sentences where the complex head with a C-feature occurs in a wrong position, generating the C-head into a position in which it cannot be selected (22).

(22) a.	*Pekka	ihaile-e-**ko**	Merja-a?
	Pekka	admire-3SG-Q	Merja-PAR
b.	*Pekka	Merja-a	ihaile-e-**ko**?
	Pekka	Merja-PAR	admire-3SG-Q

It would be easy to develop an algorithm that reconstructs C-elements from the wrong positions shown in (22) into the correct positions in (21). This would not match with the empirical facts. It is the *lack* of an operation of this type that, under the present hypothesis, explains why wrongly positioned words create ungrammatical outputs.

It is presupposed above that the prosodic focus feature [FOC] is not an inflectional element but an independent morpheme C_{foc}. This assumption was used in some early simulations, indeed it produces correct results in most cases. The discussion in the main article remained ambiguous on this issue. This was because what looks to be the same feature can occur in two other contexts: as attached to a predicate in situ, creating an in situ prosodic focus construction (23a), and inside a pied-piping phrase that is moved to SpecCP (23b).

(23) a.	Pekka	IHAILEE	Merja-a.
	Pekka	admires.3SG.FOC	Merja-PAR

’Pekka does admire Merja.’

b. [Merja-n SISKOA] Pekka ihaile-e.

Merja-GEN sister.FOC Pekka.NOM admire-3SG

’It is Merja’s SISTER, not mother, that Pekka admires.’

Assuming [FOC] is mapped to C creates a clause-internal CP-layer for (23a) and an extra CP-layer inside the pied-piped phrase for (b), both which would get rejected at the syntax-semantic interface. There is no empirical evidence for such layers, as pointed out in the main article. In addition, positing them would create empirical and theoretical problems. An additional complication is that due to its agglutinative nature it is possible to attach several features to one head or phrase. They do not generate separate C-heads but are accumulated inside the same head. Taken together these data suggest that these morphemes are inflectional criterial features, comparable to wh-feature, and do not as such project independent heads. (This was one of the reasons I rejected the hypothesis that morphological decomposition, especially when it comes to C-features, is represented in syntax by adjacent heads.) This means that when in the main article it was assumed that the highest C morpheme remains in situ during reconstruction, what we mean is that a head containing the C-feature comes into existence in syntax (e.g., foc $\rightarrow$ C^0(FOC)). There is independent evidence in favour of this hypothesis. Consider again (23b). The surface string contains a tensed verb preceded by two arguments and no overt C-head. The model must generate a phonologically null C-head to the structure on the basis of the criterial C-feature inside the moved phrase (Brattico and Chesi 2020)([...foc...] $\rightarrow$ [...foc...] C(FOC)). Thus, in both cases we have to map an inflectional feature into a head with that feature.

These mechanisms were tested in connection with negation, modal auxiliary, auxiliary verbs and with finite verbs that take infinitival complements. In such cases, the fronted element (negation, auxiliary, modal) is reconstructed to its correct gap position based on selection, while the C-element remains in situ and hosts the C-features. The mechanism was also tested with noncanonical word orders (Group 2.1.7 in the test corpus), an intervening PP argument (Group 2.1.8) and with pro-drop constructions (Group 2.1.9). These data too were calculated correctly. I will look at detailed results in Section 4.

2.4 Long head reconstruction

The Finnish long head movement pattern, in which a nonfinite verb moves over a finite verb (1), is perhaps not the norm, although long head movement constructions have been found from various languages as discussed in the main article (Lema and Rivero 1990; Rivero 1991; Roberts 1993, 2010). In English, head movement is local. Moreover, in Finnish the nonlocal pattern is restricted to cases which involve C-features; all other instances are local to my knowledge. Finally, there are examples in Finnish in which also head movement to C is local. Pure auxiliaries, for example, cannot move nonlocally (24).

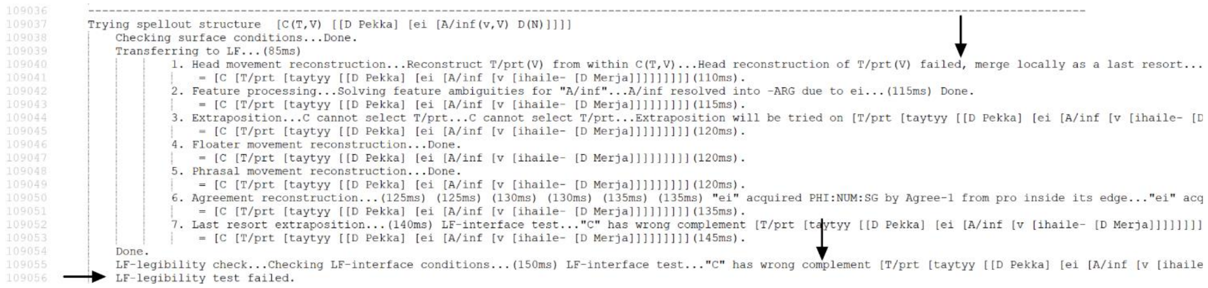
(24)	*Ollut$_1$-ko	Pekka	e-i	ole	__$_1$	tänään	koulussa?
	been-Q	Pekka.NOM	not-3SG	be		today	school

The distinction between local and nonlocal head movement seems to be associated with whether the moved element has enough lexical content to form a legitimate target for C-features. (Roberts 1993) noted that the same property distinguishes local head movement types in various languages (e.g. English and French). When an element such as a pure auxiliary moves, the interpretation does not target the specific lexical context of the auxiliary (if it has any) but the (truth of the) whole proposition (25).

(25)	O-n$_1$	Pekka	_$_1$	ollut	tänään	koulussa!
	be-PRS.3SG	Pekka.NOM		been	today	in.school

‘Pekka HAS been today in school.’

What is contrasted is the truth of the whole proposition against some prior claim in which the opposite is claimed, not the “lexical content” of the auxiliary as an independent predicate. To capture this pattern, it was assumed that head reconstruction cannot skip heads unless it is triggered by a special C*-feature, referring to the type of left peripheral features associated with both long head and phrasal movement in Finnish. Strict locality (in the sense of Travis 1984) applies to regular verbal spreads (T-v-V) as well as to auxiliary reconstruction (25) and imperatives.[25] This is captured at the level of code by using intervention. If a C*-feature is not present, then all functional heads cause intervention for head reconstruction, which becomes as local as possible, discharging everything into local head spreads.[26] This mechanism explains the lack of long head movement in cases that do not exhibit C*-features, both in English and Finnish. In such cases, the algorithm will attempt local head movement, but the resulting configuration is rejected at the syntax-semantics interface. This process is shown in Figure 11, which is a screenshot from a derivational log file for sentence #440 showing what happens when long head movement would be required for an element that does not have a C*-feature.



```
109036  -----------------------------------------------------------------------------------------------------------------
109037  Trying spellout structure  [C(T,V) [[D Pekka] [ei [A/inf(v,V) D(N)]]]]
109038      Checking surface conditions...Done.
109039      Transferring to LF...(85ms)
109040              1. Head movement reconstruction...Reconstruct T/prt(V) from within C(T,V)...Head reconstruction of T/prt(V) failed, merge locally as a last resort...
109041                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](110ms).
109042              2. Feature processing...Solving feature ambiguities for "A/inf"...A/inf resolved into -ARG due to ei...(115ms) Done.
109043                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](115ms).
109044              3. Extraposition...C cannot select T/prt...C cannot select T/prt...Extraposition will be tried on [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D
109045                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](120ms).
109046              4. Floater movement reconstruction...Done.
109047                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](120ms).
109048              5. Phrasal movement reconstruction...Done.
109049                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](120ms).
109050              6. Agreement reconstruction...(125ms) (125ms) (130ms) (130ms) (135ms) (135ms) "ei" acquired PHI:NUM:SG by Agree-1 from pro inside its edge..."ei" acq
109051                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](135ms).
109052              7. Last resort extraposition...(140ms) LF-interface test..."C" has wrong complement [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]
109053                 = [C [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile- [D Merja]]]]]]]](145ms).
109054      Done.
109055      LF-legibility check...Checking LF-interface conditions...(150ms) LF-interface test..."C" has wrong complement [T/prt [taytyy [[D Pekka] [ei [A/inf [v [ihaile
109056      LF-legibility test failed.
```

Figure 11. Failed head reconstruction. The gap is nonlocal, but head reconstruction cannot reach it (line 109030); hence the last resort option is selected. This configuration fails at the syntax-semantic interface (line 109056)

Provided that the reconstruction is triggered by a C*-feature in the complex head, long head movement becomes available. The head reconstruction algorithm travels downwards on the projectional spine of the phrase structure, following labelling and selection and searching for the first suitable position for

[25] Since it is unclear to me at present writing what the semantic effects of local T-to-C movement are (perhaps verum focus, at least in some cases), local T-to-C construction is derived by creating a regular finite C-head (C/fin) without C*-features. In English, fronting an auxiliary creates an interrogative sentence, but the interrogativization targets the whole proposition (thus, what is created is finite C plus some type of interrogative feature). Some languages seem to exhibit a formal root level fronting, or this corresponds to some root level feature. In Finnish, imperatives can be created by moving a local finite imperative verb.

[26] A functional head is defined by feature [!COMP:*] = 'must have some complement' which in this case functions as the intervening feature. The feature intervention approach was adopted, following (Roberts 1993, 2010), in order to leave room for long head movement constructions that do not follow the A-bar profile exhibited by Finnish. Nevertheless, the approach, although based on a fairly popular approach going back to Rizzi (1990), still feels ad hoc when examined at the level of actual code.

the reconstructed object. The process is called *minimal search*, suggesting a similarity to the minimal search system proposed in Chomsky (2008) but interpreted as a top-down reconstruction. The essence of the search algorithm is (26).[27] Suppose α is targeted for search, then

(26) *Minimal search*

if α is primitive, terminate search; otherwise search X in $\{_{XP}$ X Y$\}$, but if X is primitive, search Y.

The algorithm dodges left phrases (specifiers, subjects) and right adjuncts and enters complements when X is primitive. This captures CED effects in the dataset. Most reconstruction, whether head or phrasal, uses minimal search. It is interesting to observe in this connection that both left branches and right adjuncts are transferred independently to the semantic component as phases, hence this could ultimately provide a deeper explanation for (26).[28] However, no such unification was pursued in this study. Figure 12 shows the calculated results for a long-distance head movement sentence (27).

(27)	IHAILLA$_1$	Pekka	sanoo	että	Pekka	haluaa	__$_1$	Merja-a. (#448)
	admire.A/INF.FOC	Pekka	says	that	Pekka	wants		Merja-PAR

[27] The intervention condition is currently provided in the head reconstruction algorithm, but a deeper explanation would include it into minimal search itself. It would then generalize to all reconstruction operations that use this algorithm.

[28] We could perhaps reduce the definition into 'search what is currently in the active working memory', assuming that left branches and right adjuncts have been transferred independently (as they are).

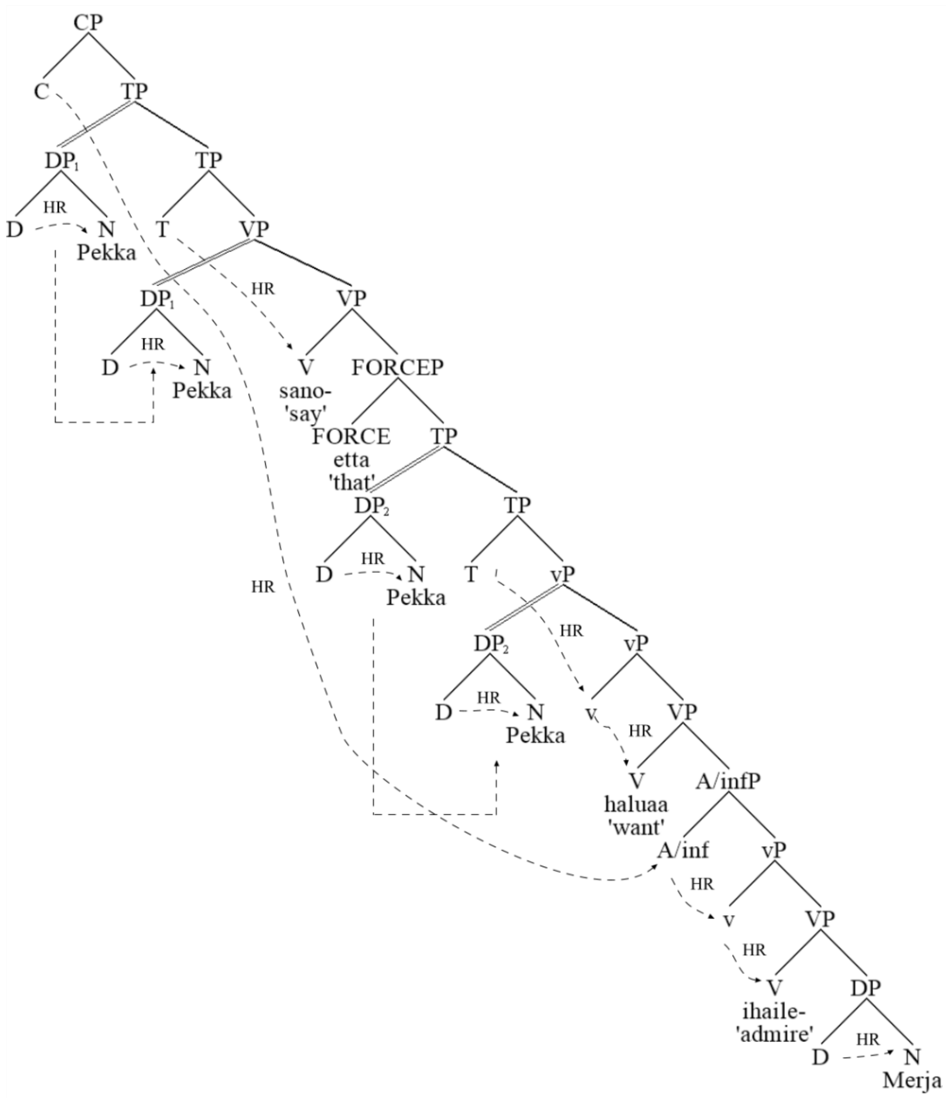


Figure 12. Calculated output for sentence (27)

The model reconstructs A/inf(V) from within the fronted head, performs (in principle unlimited) search and finds the position indicated by the arrow, where the A-infinitival can be selected. All long head movement constructions are derived in the same way.

In the main article some considerations were presented suggesting that the reconstruction dependencies are those of the Ā-system. The C*-features associated with the fronted head are also involved in Finnish phrasal Ā-movement.[29] Both cases are handled by the same system, an idea that

[29] With few exceptions: imperatives are only possible in connection with verbal head movement, whereas wh-features cannot be attached to verbal heads.

was to my knowledge first explored by Roberts (1993, 2010), where it was proposed that the Ā-system can target also (verbal) heads, not only phrases.[30] Suppose we consider regular phrasal A-bar movement that involves pied-piping, such as (28).

(28) [Peka-n-ko auto] hajosi?

Pekka-GEN-Q car broke

'Was it Pekka's (and not Merja's) car that broken down?'

Once the pied-piped phrase is reconstructed, the syntax-semantics interface sees an operator-scope dependency (29).

(29) C(ko). . . T . . .[[Peka-n-ko auto] broke]

C(Q). . . T [[Pekka-GEN-Q car broke

There is a dependency between C(Q) and a word *Pekka*. Head movement constructions exhibit the same dependency, this time between C(Q) and some verbal head, such as the A-infinitival. As pointed out by Roberts, the only difference between these two cases appears to be that the former involves pied-piping as an additional mechanism. Although many problems remain, this to me seems to be the correct.

I will conclude this section by considering local head movement. Finnish exhibits HMC-compliant local head movement if no C*-feature is involved. This concerns Neg-to-Force movement, pure auxiliary movement to C, all cases of head ordering that appear without C*-feature and the standard cases of morphological decomposition, independent of the lexical categories of the heads. All these cases, as well as cases reported in the literature to my knowledge, follow (43A-B) for reconstruction with the additional property that movement is HMC-compliant. Thus, the case of Finnish is relatively

[30] Roberts (1993: Chapter 1) posits a distinction between A and Ā heads (directly mirrored by the C/C*-features in the main article and C/FIN vs. C/OP at the level of implementation) and furthermore captures the corresponding head movement locality restrictions (HMC vs. LHM) by relying on feature intervention, adopted here as well. Feature intervention allows us to posit several varieties of LHM, which seems reasonable in light of data crosslinguistic LHM data currently available. He also noted that auxiliary and lexical content verbs behave differently, a distinction I assumed here as well. Finally, one of the Roberts' head movement configurations (substitution X^0-Y^0) corresponds closely to the X(Y) model adopted here.

simple. Roberts' (1993) hypothesis that the LHM is restricted to cases of Ā-reconstruction/movement with verbs with lexical content seems correct for Finnish; some type of predicate clefting seems at issue. This model does not generalize to all languages, however. In some languages (e.g. V2), head movement to C seems to have little semantic import; in some, it has semantic relevance (e.g., interrogativization) but we observe the auxiliary/lexical distinction. Finally, some languages exhibit the canonical LHM signature ("movement of an infinitival over a finite verb") with semantic import, but the operation is more restricted than it is in Finnish. In all these cases, rule (43A-B) seems to apply, but is syntactically more limited in application and/or the features involved differ. To capture the "more narrow" LHM signature we could either parametrize Ā-LHM, perhaps by using the intervention feature, as is the case here, or parametrize HMC by allowing the operation to skip heads. To test any of these models requires, however, that we construct a representative test corpus for the relevant languages.

2.5 Semantics and the operator-variable system

Before addressing the semantic interpretation of the head movement constructions specifically, I will provide a brief description of the overall semantic architecture assumed at the present writing to make navigating the source code easier.

When a phrase structure object is transferred to the semantic component, it first arrives at the syntax-semantic interface (LF interface) that mediates most of the communication between syntax and semantics.[31] Syntax-semantics interface objects are phrase structure objects arriving from transfer. Most of the phrase structures illustrated in this document and in the main article are syntax-semantic interface objects in this sense. All intermediate objects are available in the derivational logs. Finally, the syntax-semantics objects are not transformed or manipulated in any way; they are only interpreted and translated into another format that the language external cognitive systems (conceptual-intentional systems) can access.

[31] There is an additional pragmatic interface that reacts to noncanonical word orders, but this interface is not relevant to the issue at hand and does not affect syntactic processing in any way.

Information at the syntax-semantic interface is interpreted by a larger system called *narrow semantics* which translates information arriving from the syntactic pathway into "instructions" understood by the conceptual-intentional systems. Narrow semantics consists of several subsystems reacting to various features in the input; I consider many grammatical features to be "grammaticalized instructions" that these external systems use. These extensions are shown in Figure 13.

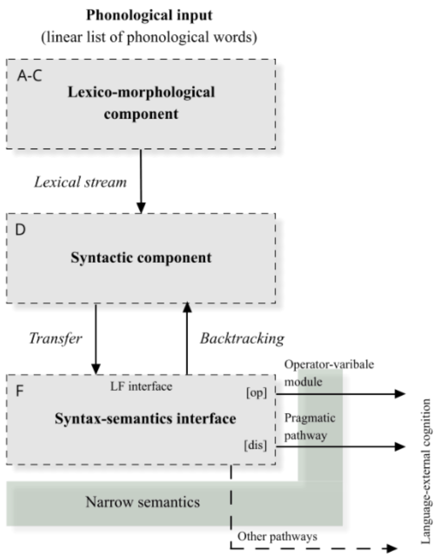


Figure 13. Internal structure of the semantic component. Narrow semantics wraps the syntactic processing pathway and mediates most of the information flow between syntax and language-external cognition.

The various subsystems existing inside narrow semantics respond to particular feature types. One of these systems, called the operator-variable module, responds to C*-features.[32] It handles both phrasal operator constructions and head operator constructions. An operator is interpreted by linking it with *propositional scope*. Propositional scope is determined in one of two ways. If the input string contained an overt scope marker, then the scope will be marked by the finite head that contains the same operator feature, copied from the overt scope marker during transfer, as explained above. If the scope is not

[32] At the level of code C*-features are denoted by OP:F, with F replacing the feature (e.g., WH, Q) and OP diverging the processing to the operator-variable module.

marked overtly, then any c-commanding finite head can be selected. Thus, in the case of a regular interrogative (30a), the reconstructed operator *who* is linked with the propositional scope as indicated by the original surface position of the same operator.

(30)	a.	Who + does + John + admire?		(Input)
	b.	[who$_1$ [T . . .	who$_1$]]]]]	(Syntax-semantics interface object)
		FIN, OP:WH	OP:WH	(Scope)

This is interpreted so that the scope of the question is the material covered by the scope dependency, shown by the line in (30). The operator feature that appears at T is copied during transfer, as explained in Section 2.3. If the operator were inside an embedded clause and the scope in the main clause, a wide scope reading results (31). In this case the scope dependency covers more material that it does in (30).

(31) Who$_1$ did John claim that Mary admires __$_1$?

The same system interprets Finnish long head movement constructions, in agreement with Roberts' idea that long head movement and long phrasal movement are based on the same system. Thus, (32) is interpreted so that the fronted A-infinitival constitutes a yes/no operator associated with a predicate with a scope spanning the whole clause. The long head movement tells the algorithm what the scope is, as the C-head remains in situ while the rest reconstructs (Section 2.3). This matches with the semantic interpretation elicited from native speakers. The speaker is asking if it's true that Pekka wanted to sell (and not, say, loan) his possessions.

(32)	Myy-dä-kö	Pekka	halus-i	_	omaisuute-nsa?
	sell-A/INF-Q	Pekka.NOM	wanted-PAST.3SG		possessions-PX/3SG
	C			A/inf V	
	OP:Q, FIN			OP:Q	

Figure 14 shows the outcome for a structurally similar sentence (33) that can be found from the test materials (sentence number #266).

(33) Ihailla-ko	Pekka	haluua	_	Merja-a?
admire.A/INF-Q	Pekka	wants		Merja-PAR

‘Does Pekka want to admire (and not hate) Merja?’

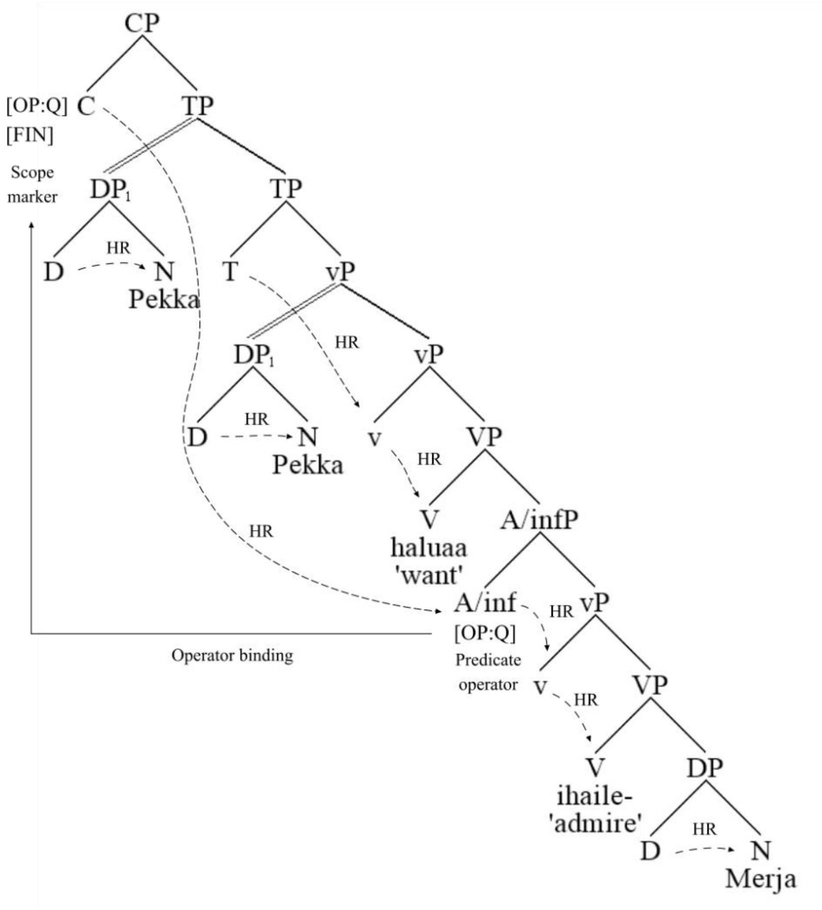
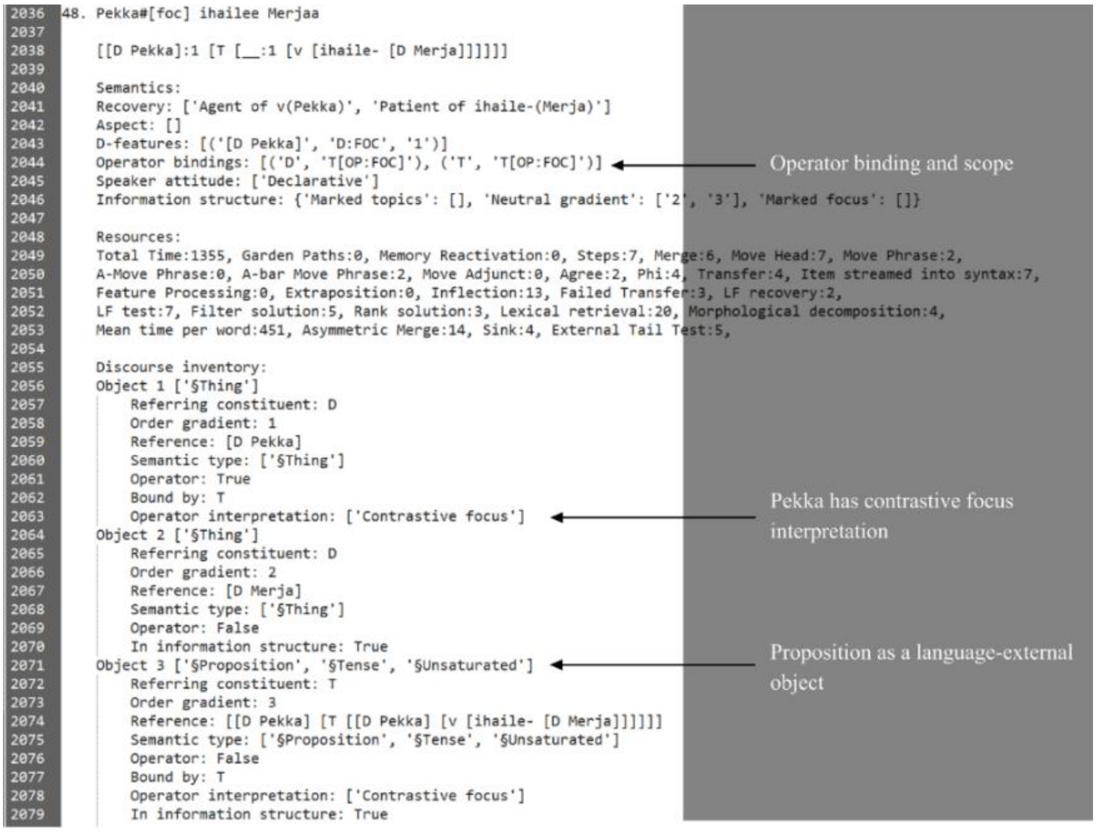


Figure 14. Head reconstruction and operator binding for a long head movement construction in Finnish.

This information is present in all output from the algorithm, and for all grammatical input sentences. Figure 15 shows the same information in a text format for sentence *Pekka#foc ihailee Merjaa* ‘PEKKA admires Merja’, where the grammatical subject has been focussed contrastively.



```
48. Pekka#[foc] ihailee Merjaa

    [[D Pekka]:1 [T [__:1 [v [ihaile- [D Merja]]]]]]

    Semantics:
    Recovery: ['Agent of v(Pekka)', 'Patient of ihaile-(Merja)']
    Aspect: []
    D-features: [('[D Pekka]', 'D:FOC', '1')]
    Operator bindings: [('D', 'T[OP:FOC]'), ('T', 'T[OP:FOC]')]
    Speaker attitude: ['Declarative']
    Information structure: {'Marked topics': [], 'Neutral gradient': ['2', '3'], 'Marked focus': []}

    Resources:
    Total Time:1355, Garden Paths:0, Memory Reactivation:0, Steps:7, Merge:6, Move Head:7, Move Phrase:2,
    A-Move Phrase:0, A-bar Move Phrase:2, Move Adjunct:0, Agree:2, Phi:4, Transfer:4, Item streamed into syntax:7,
    Feature Processing:0, Extraposition:0, Inflection:13, Failed Transfer:3, LF recovery:2,
    LF test:7, Filter solution:5, Rank solution:3, Lexical retrieval:20, Morphological decomposition:4,
    Mean time per word:451, Asymmetric Merge:14, Sink:4, External Tail Test:5,

    Discourse inventory:
    Object 1 ['§Thing']
        Referring constituent: D
        Order gradient: 1
        Reference: [D Pekka]
        Semantic type: ['§Thing']
        Operator: True
        Bound by: T
        Operator interpretation: ['Contrastive focus']
    Object 2 ['§Thing']
        Referring constituent: D
        Order gradient: 2
        Reference: [D Merja]
        Semantic type: ['§Thing']
        Operator: False
        In information structure: True
    Object 3 ['§Proposition', '§Tense', '§Unsaturated']
        Referring constituent: T
        Order gradient: 3
        Reference: [[D Pekka] [T [[D Pekka] [v [ihaile- [D Merja]]]]]]
        Semantic type: ['§Proposition', '§Tense', '§Unsaturated']
        Operator: False
        Bound by: T
        Operator interpretation: ['Contrastive focus']
        In information structure: True
```


Figure 15. Result summary for sentence *Pekka#foc ihailee Merjaa* 'PEKKA admires Merja'.

The field "semantics" in Figure 15 provides a summary of the semantic interpretation, while the field "operator bindings" shows that the D-operator (head of DP = *Pekka*) was bound by finite T with feature OP:FOC.[33] The lower part of the results summary contains a list of language-external objects that were projected into existence during processing. Notice that also the proposition is projected into the global discourse directory when it is identified by the operator it contains. The idea is that it becomes accessible for the language-external cognitive system.

This approach to semantic interpretation was originally inspired by Chomsky 2008, where the author proposes a "duality of interpretation" approach in which two distinct semantic systems interpret

---

[33] Finite T will also "bind itself," as a consequence of the assumptions made here. This could be ruled out by blocking reflexive binding, yet the matter is obviously more complex because T can function as an operator (T-to-C movement with an operator feature at T). Another reason why this mechanism was left into the model was because it could be useful in the explanation of the genesis of verum focus interpretation that targets the whole proposition. Perhaps in such cases the operator and the scope marker are the same element?

different aspects of the incoming syntactic structure. The idea, to the extent that my interpretation of it is correct, was implemented by assuming that the “second system” responsible for operators and their propositional scopes reacts to specific feature types in the input. We can think of the linguistic structure as “grammaticalizing conceptual realms” that are part of the extralinguistic global cognition. I do not restrict the number of semantic processing pathways into two, however. For example, noncanonical word orders are interpreted by a separate pragmatic pathway to represent discourse-configurational topic-focus structures (see line 2406 in Figure 15)(Brattico 2021a).

Notice that the algorithm that calculates any operator’s scope relies on finiteness. The scope is determined by finiteness + a copy of the operator feature itself. This links grammatical finiteness directly with the semantic notion of proposition.[34]

3 ADDITIONAL TOPICS

3.1 Performance

We can collect detailed performance metrics of the operation of the algorithm. This makes it possible to compare computational costs and the psycholinguistic plausibility of alternative hypotheses and proposed computational mechanisms. We can also compare different solutions in terms of their economy properties if considered relevant.[35] Performance data concerning the simulations run in the present study are summarized in Table 1 and explained further below.

---

[34] What remains to be implemented and to some extent problematic is the restriction limiting the operator interpretation to predicates that have “sufficient lexical content.” The syntactic background model assumes that auxiliary verbs and similar elements lacking lexical content lack a feature [ASP] projecting an independent event structure and thematic roles, so an implementation could be done on such basis, but the problem is that the operator dependency is established between C and the highest functional head in the raised predicate that does not alone seem to be able to provide the distinction we need. The relevant unit is the functional head plus the verb (T + V), not T alone. The information is available in the input in the fronted complex predicate, but is not transmitted to the LF interface by the current implementation. This is connected to a larger problem that remain unaddressed: what causes the components of the predicate coalesce together in production? Is it a phonological feature of some type, or is semantics (LF interface) involved? Roberts (1993) assumes that some head movement operations are triggered by head-internal selection features.

[35] Complexity metrics have played a theoretical and/or empirical role in the minimalist program (Chomsky 2000:104-105, 110-111), where the putative empirical evidence supporting the use of such notions in linguistic theorizing has resulted in them being used also as theoretical constraints or goals.

Table 1.
Performance metrics (mean processing times, number of operations) for the present dataset.

	Group						
	1	2	3	4	5	6	All
Means per relevant unit							
Total PPT/sentence (ms)	1878.9	1791.6	2048.2	2163.8	3511.0	4860.0	2042.8
PPT/word (ms)	447.0	493.6	497.6	432.8	505.1	1185.0	495.1
N of garden paths/sentence	0.2	0.0	0.1	0.0	0.0	2.4	0.1
N of Merge-1/sentence	8.4	8.4	9.6	10.6	14.7	18.0	9.4
N of Move head/sentence	18.2	15.9	17.2	21.9	49.3	35.4	19.1
N of Move phrase/sentence	3.0	3.1	3.1	2.2	15.1	9.4	3.5
N of Agree/sentence	6.6	7.3	8.6	14.3	24.5	4.4	9.0
N of Asymmetric Merge/sentence	55.8	66.0	80.7	74.4	228.7	82.4	75.1

Groups: 1 = baseline tests, 2 = Local T-to-C movement, 3 = LHM, 4 = LHM over two V-elements, 5 = Super (long distance) LHM, 6 = VP fronting.

The columns in Table 1 represent the experimental groups of the test corpus (1 = baseline tests, 2 = Local T-to-C movement, 3 = LHM, 4 = LHM over two V-elements, 5 = Super (long distance) LHM, 6 = VP fronting; 7-10 contained ungrammatical sentences and are ignored). The user can use whatever groups and grouping relevant for the research agenda or hypothesis, here I use the groups that were readily available in the test corpus. The rows show summaries of the performance metric calculated over all sentences in each group. The first row represents total PPT per sentence, the second the mean PPT per word. PPT refers to predicted processing time. Predicted processing times are calculated in the following way. Each independent syntactic operation projected by the model (e.g., Merge, Move) is associated with a predicted cognitive cost (in milliseconds), corresponding to an estimated neuronal processing cost in the human brain of that operation, which are summed up during real-time processing to yield total predicted processing times for each input sentence. The PPTs shown in Table 2 are computed in this way. For example, the current model predicts that the average comprehension speed for sentences in groups 1-5 is approximately 500ms per word, whereas in group 6 it is much longer. These results are obtained when most elementary operations projected by the model are associated with the PPT of 5ms. They should ultimately reflect real physiological properties of the human brain, although currently they are only used in a relative sense to compare the computational costs of various algorithmic language comprehension models.[36] Notice that these numbers include not only operations that were done overtly and recorded into the log files, but also all covert operations that were used when for example evaluating candidate solutions.

Below the two PPT estimates is a row recording the mean number of garden paths (row "N of garden paths/sentence" in Table 1). The algorithm generates garden paths in connection with 26 test sentences in this test corpus, almost all which occur in connection with VP-fronting (Group 6 in the test corpus, #479-485) that are (rightly or wrongly) hard for the model. In all other cases the first pass parse corresponds to an acceptable (and correct) interpretation. Thus, there is very little garden pathing. The model uses several cognitive heuristic parsing principles to achieve this efficiency.[37] The researcher can activate parsing heuristics freely when configuring the simulation. In the main article it was noted that the grammaticality and semanticality judgments involving VP-fronting were unclear and difficult to do. This could be because they are associated with an extra processing cost, influencing acceptability.

Raw performance metrics are provided in the output file LHM_corpus_resources_FINAL.txt, which contains a comma-delimited text file listing each input sentence followed by all performance metrics associated with the processing of that sentence, as recorded during processing. This file can be loaded into an external program such as Excel or SPSS for analysis.[38]

---

[36] Merge^{-1} ("inverse Merge") in Table 1 refers to an operation in which a primitive or phrasal constituent is attached to the right edge of an existing partial phrase structure during parsing, including when garden pathing (backtracking). It does not include more primitive Merge operations that are executed during reconstruction, for example. Asymmetric Merge includes all elementary bottom-up Merge operations in which two constituents are combined together anywhere in the system. This operation occurs in connection with Merge-1 as well as with any phrasal or head reconstruction, thus the number is higher. Asymmetric Merge and the number of garden paths provides the best overall estimation of the computational cost of any proposed solution.

[37] If these principles are knocked out, efficiency drops dramatically. Processing of a sentence with one embedded that-clause (e.g., *John said that Mary admires him*) consumed one hour from a standard desktop computer and generated 134795 garden paths in one simulation study; in the present study the number of garden paths for sentences like this was zero.

[38] At the present writing no published studies on the performance aspects of the syntactic background theory exists, and therefore the numbers provided by the algorithm, although plausible on the face of it, should be regarded with some caution.

3.2 Transfer and transformations

Transfer is executed when a candidate solution, either a whole sentence or some part of it, is submitted to the syntax-semantic interface for evaluation. Head reconstruction is part of transfer.

One way to see why a mechanism of this type must exist is to consider the Finnish free word order profile. In a basic ditransitive clause virtually all logically possible word order permutations are also grammatical, yet the attested word forms themselves are under strict grammatical control.[39] The A-infinitival, for example, can be selected only by certain kinds of words, independent of where they happen to be in the input. Because word order is relatively free, these constrains must operate independent of the surface positions of any of these elements. Furthermore, the restrictions are closely matched or identical with conditions on semantic interpretation. Thus, the intuitive reason why the Finnish A-infinitival can occur only in specific grammatical environments has to do with the way it is interpreted as denoting an 'object event' of some other predicate ('wants + to sell' but not 'to see wants'). If we assume that the conceptual system is universal, then there has to be a mechanism that neutralizes the attested surface variations and produces canonical structures that the universal semantic system can interpret. Head reconstruction follows the same logic: it analyses contents of complex phonological words so that the selectional restrictions between its internal constituent lexical items can be evaluated and interpreted by principles that do *not* presuppose that the input language is agglutinative.

I have found that the best match between empirical data and the model can be attained if we adopt a broadly generative framework and assume that what have traditionally been described as transformations in that theory are interpreted as a mirror image of transfer. This explains why the model produces what look like inverted transformational chains. This assumption is empirical. I do not know anything else in the current cognitive science literature that comes even close to capturing the complex properties of Finnish long head reconstruction. This claim should be evaluated against the type of evidence discussed in Section 1.2 and in the main article. It is, however, in principle easy to imagine

[39] Case suffixes in the case of arguments, specific verb forms in the case of predicates.

and develop more elegant alternatives and then demonstrate that they are superior by embedding them into some algorithm and calculating the correct empirical properties.

3.3 A note on phonological content at C and the V2 property

The test corpus includes examples in which C*-features are attached to a wrong head or the host head is not in a fronted position. An earlier model overgeneralized examples such as (34), judging them grammatical.

(34)	*Pekka	ihaile-e-ko	Merja-a? (Wrongly judged grammatical)
	Pekka.NOM	admire-T/FIN.3SG-Q	Merja-PAR

These sentences are ungrammatical because it is not possible to fill in SpecCP and C at the same time. Capturing this restriction turned out to be nontrivial. We must relate the availability of the SpecCP position in some way to the phonological content at C, yet phonological content was up to this point unavailable inside narrow syntax. It has never been needed for the derivation of any other data. The problem is possibly more general, however, in that the model was also accepting (35) where the subject appears above the high complementizer *että* 'that' that no phrase can fill under any circumstances in Finnish.

(35)	*Jari	sanoi	[$_{CP}$Pekka$_1$	että	__$_1$	[ihaile-e	Merja-a.]]]
	Jari	said	Pekka.NOM	that		admire-T/FIN.3SG	Merja-PAR

In order to connect the availability of SpecCP to the phonological content at C, a mechanism was created which embeds a negative specifier selection feature to the C-head when it is created from the phonological string. The operation is performed inside the presyntactic lexico-morphological component that has access to phonological material. The same feature is part of the lexical content of the high complementizer *että* 'that'. The solution is not very satisfactory, but perhaps not completely implausible either since this phenomenon is in some ways reminiscent of the V2 profile, thus possibly not an isolated anomaly.

3.4 Standard cyclic theory and the present analysis: some comparisons

Before trying to compare the present analysis with standard cyclic theories, it is perhaps important to point out that most or all standard theories, whether they have been developed within the generative perspective or within some other formalism or framework, are typically interpreted as analyses of grammatical competence, thus they are descriptions of what is possible and impossible in a human language. They therefore take a neutral or agnostic stance on how the described limitations are operating during generation or recognition of linguistic forms by native speakers when they use language for communication or for some other purpose. A cyclic, bottom-up derivation such as that proposed in Roberts 2010, for example, does not describe the actual causal sequences that occur when speakers generate utterances. Thus, Chomsky points out that

> it is perhaps worth while to reiterate that a generative grammar is not a model for a speaker or a hearer. It attempts to characterize in the most neutral possible terms the knowledge of the language that provides the basis for actual use of language by a speaker hearer. When we speak of a grammar as generating a sentence with a certain structural description, we mean simply that the grammar assigns this structural description to the sentence. When we say that a sentence has a certain derivation with respect to a particular generative grammar, we say nothing about how the speaker or hearer might proceed, in some practical or efficient way, to construct such a derivation. These questions belong to the theory of language use - the theory of performance. (Chomsky 1965: 9).

A pure competence theory can be seen as providing boundary conditions that a more detailed information processing theory must respect. In the same way, although I interpret the present analysis in terms of the two levels in Marr's system, we could retreat from this position and claim that the description specifies only a characteristic function for the set of grammatical expressions. Once we do this, it is possible to compare the two approaches.

For example, what in the cyclic bottom-up theories are described as case *assignment* must be described as *checking* in the inverse model, since all agreement and case features are already present in the input. We must verify them for correctness. Thus, valuation and assignment translate into checking.

The two approaches are not notational variants. Case suffixes reflect the grammatical contexts of the canonical positions of arguments that carry them and can be used for reconstruction purposes. In general, morphological word forms play a major role in reconstruction computations, as hierarchical structure is absent from the sensory input and must be inferred from something, most likely at least in part from the overt morphological forms. Thus, what in the cyclic bottom-up theories are sometimes described as idle morphosyntactic by-products of the generation – virus features that need elimination[40] – become essential ingredients of the recognition algorithm. The same reasoning applies to linear order. Whereas in the cyclic generative approaches linear order can be considered a relatively unimportant ancillary phenomenon, a recognition model cannot treat it as irrelevant.

The same reasoning can perhaps be applied to the question of what might ultimately motivate displacement itself, a problem that has animated research at least for six decades. In the present dataset the dislocated predicate signals the scope of the operator it carries. The scope, as elicited from native speakers, always corresponds to the material that falls under the dependency (36).

(36)	Myy-dä-kö	Pekka	aikoi	$\_\__1$	kaiken	omaisuutensa?
	sell-A/INF-Q	Pekka	planned		all	possessions

In the cyclic generative model, on the other hand, scope is part of the intended interpretation and comes essentially for free. It is where the speaker intends it. Thus, the operation copying criterial features to a local head functions as the scope marker for the reconstruction operation and therefore communicates the propositional context of the operator through the sensorimotoric interface.[41] The same reasoning

[40] Thus, the actual overt morphosyntactic features can be viewed as "redundant" in having "no semantic interpretation," thus "they must be deleted before they reach the semantic interface for the derivation to converge" (Chomsky 2008: 154). See also (Chomsky 2000).

[41] It is presupposed here that (i) the properties of any head H can be reconstructed from its own overt content and from the material in its specifier(s) and that (ii) noncanonical phrases are reconstructed to their thematic canonical positions by transfer. Condition (i) copies criterial operator features to the finite C-head, while condition (ii) repairs noncanonical orders. As a consequence, the input to the semantic module interpreting operator-variable constructions generates the correct entities and dependencies to the global discourse inventory.

can perhaps be applied to other forms of displacement. Thus, we assume that they trigger semantically interpretable reactions by transfer. Of course, these speculations would be useless unless it can be shown that they provide correct predictions also in the case of ungrammatical input sentences. In some range of core cases they do. For example, if the complex word containing the operator feature occurs in situ, an ill-formed phrase structure object that hosts a C-element in the middle of the clause would arrive to the syntax-semantic interface in such position, leading into a crash. An example is provided by sentence #558 in the test corpus, and a screenshot of the relevant derivational log file is below[42]:

```
142398   --------------------------------------------------------------------------------------------------------------------
142399   Trying spellout structure  [C [[D Pekka] [C(T,v,V) D(N)]]]
142400       Checking surface conditions...Done.
142401       Transferring to LF...(85ms)
142402               1. Head movement reconstruction...Reconstruct T(v,V) from within C(T,v,V)...Must reconstruct D(N) first...Reconstruc
142403                  = [C [[D Pekka] [C [[D Merja] [T [v ihaile-]]]]]](105ms).
142404               2. Feature processing...Done.
142405                  = [C [[D Pekka] [C [[D Merja] [T [v ihaile-]]]]]](105ms).
142406               3. Extraposition...C cannot select C...C cannot select C...Extraposition will be tried on [C [[D Merja] [T [v ihaile
142407                  = [C [[D Pekka] [C [[D Merja] [T [v ihaile-]]]]]](110ms).
142408               4. Floater movement reconstruction...[D Pekka] failed to tail [ARG] [FIN] [VAL] ...Dropping [D Pekka]...<D Pekka> wa
142409                  = [C [<D Pekka>:2236 [C [<D Merja>:2237 [T [<__>:2236 [v [<__>:2237 ihaile-]]]]]]]](150ms).
142410               5. Phrasal movement reconstruction...Done.
142411                  = [C [<D Pekka>:2236 [C [<D Merja>:2237 [T [<__>:2236 [v [<__>:2237 ihaile-]]]]]]]](150ms).
142412               6. Agreement reconstruction...(155ms) (155ms) (160ms) (160ms) (165ms) (165ms) "T" acquired PHI:NUM:SG by Agree-1 frc
142413                  = [C [<D Pekka>:2236 [C [<D Merja>:2237 [T [<__>:2236 [v [<__>:2237 ihaile-]]]]]]]](165ms).
142414               7. Last resort extraposition...(170ms) LF-interface test..."C" has wrong complement [<D Pekka>:2236 [C [<D Merja>:22
142415                  = [C [<D Pekka>:2236 [C [<D Merja>:2237 [T [<__>:2236 [v [<__>:2237 ihaile-]]]]]]]](175ms).
142416       Done.
142417       LF-legibility check...Checking LF-interface conditions...(180ms) LF-interface test..."C" has wrong complement [<D Pekka>:223
142418       LF-legibility test failed.
```

Figure 21. A derivation that crashes due to a sentence-internal CP-element. Notice that extra CP-layer on line 142399

Linear order plays a key role in any explanation of this type.

The standard cyclic theory derives expressions by applying an iterative operation Merge, which takes two elements α, β as input and yields a new element $\{\alpha, \beta\}$ (Chomsky 2008: 138). The assumption that Merge creates sets requires an extra axiom which restricts the size of the set to two. Implementing this requirement looks suspicious. It implies that sets have to be counted when merge is attempted. A set should not, by its nature, be limited in that way, which raises the question if merge really generates sets. Second, and more importantly, we cannot ignore linear order when interpolating syntactic structures from sensory input. These two problems were solved in this study by assuming that Merge

[42] I found one edge case in which a sentence of this type (e.g., #579) crashed because the algorithm was unable to discharge all internal components of the wrongly positioned complex predicate. This is discussed in Section 4.3.5.

creates asymmetric binary-branching structures [α, β] with the left and right constituents being defined axiomatically. This assumption simplifies labelling and derives automatically the limitation that phrasal constituents must have exactly two daughters. It provides axiomatic and simple definitions for the notions of left and right edge. An additional difference between the cyclic theory and the present model is that a recognition model creates phrase structures from the sensory input, hence we require a more complex operation (at least when working with syntactic objects that reside at the syntax-lexicon interface) which targets nodes α in the existing phrase structure and substitutes them with [α, β], β being the incoming lexical item. The target + Merge + substitute was called Merge^{-1}. Instead of expanding the structure at the top, it is expanded at the right edge. Yet, right edge expansion presupposes cyclic Merge (called *asymmetric Merge* in the implementation), which is therefore part of the model as well.

It should be noted in this connection that the right edge expansion does not force one to adopt the comprehension perspective. It is possible that language is produced or at the very least can be produced by using the same mechanism, thus by creating sentences in the same order as they are uttered. It is also possible, at least in theory, that the motoric planning stage is executed by creating spellout structures and not syntactic-semantics interface objects. While speculation, this point is relevant in the sense that we do not need to associate any generative model with any concrete causal information processing model. We can consider the present model, as well as any cyclic system, as a formally rigorous way of specifying the “logic” of recursive composition without taking a stance of what it means from the point of view of actual comprehension or production.

Ultimately the matter comes down to the question of what kind of phenomenon language is. Chomsky argues that “the primary contribution to the structure of [faculty of language] may be optimization of mapping to the C-I interface. . . while the mapping to the [sensorimotoric] interface is an ancillary process” that does not enter in the any “principled explanation” (Chomsky 2008: 136). This represents the opposite of what I have claimed here. Transfer exists to process sensory objects (sensorimotoric interface objects) into a format that can be understood by the syntax-semantics interface, and therefore we could conclude that they are both equally important components in any "principled explanation." It is clear that much of the universal limitations of language are derived in the present model from the concrete sensorimotoric properties (linear order, overt morphology, displacement). For example, the fact that a wrongly positioned complex predicate with a C*-feature leads into ungrammaticality is explained as a consequence of the very narrowly construed and thus limited operation that generates syntactic structures and semantic interpretation from the linear order itself. The mechanism is so "unintelligent" that it insists on positioning the element into the middle of the clause as directly cued by its position in the surface string. In the same way, the system refuses to consider hypotheses in which the highest morpheme of a complex word would not belong to the exact position in which it appears in the surface string.

3.5 Lexicon, crosslinguistic variation and parameters

The issue of modelling crosslinguistic variation is nontrivial both from an empirical and implementation point of view.

The lexicon is created at runtime by using three sources, all which exist as independent text files. One file contains a list of universal morphemes, such as tense ('past, present'), interrogativization ('who, what'), transitivity ('admires Mary'), conjunction ('and') and many others that are assumed to be grammaticalized in many or perhaps in all languages.[43] Another lexical file contains language-specific lexical items, mostly open-class content words. Finally, there is a third file that contains the lexical redundancy rules. A lexical redundancy rule can be thought of as specifying a cluster of default features. For example, it is specified in the lexical redundancy rule list that any preposition must have a DP-complement. Each lexical redundancy rule is defined by using a formula $f_1 \ldots. f_n \rightarrow g_1 \ldots g_n$, in which the presence of features $f_1 \ldots. f_n$ in a lexical item triggers the presence of features $g_1 \ldots g_n$. If a lexical redundancy rule is violated in the language-specific lexicon, the latter wins; hence lexical

[43] It does not matter if there are languages that do not use some of these items, or if there are languages that use items that are not in this list; the reason they are contained in a separate file is to make finding and editing them easier. Also, by putting an item into this list the researcher is tentatively claiming that it should be found from several or all languages, in some form. Most or in fact all of these elements are functional heads and inflectional features.

redundancy rules specify default behavior that can be overridden by the language specific properties. When a lexicon or a lexical element is created at runtime, its properties are first retrieved from the language specific lexicon and then processed through lexical redundancy rules.

This architecture raises the question, however, of how to define language specific features for functional elements. English finite T, for example, must contain a feature requiring strict EPP behavior, but this feature should not be present if the language has a consistent noncanonical VSO profile. In Finnish, finite verbs exhibit EPP behavior but the system is based on definiteness or topicness, not grammatical subjecthood.

In some of the early implementations this problem was addressed by using a third lexical layer or a "computational prism" of sorts that applied parametric rules to lexical elements when they were retrieved from the lexicon. A parametric rule was a rule that attributed behaviours to functional lexical elements during lexical retrieval on the basis of the input language. The whole component looked completely ad hoc, however, and it was impossible to me to think that crosslinguistic parameters were based on an explicit lexical transformation "tunnel." Thus, the system was replaced by a mechanism, used in the present study as well as in several others, in which lexical redundancy rules could be specified in such a way that they applied only to selected languages. For example, the English EPP rule could be stated as a lexical redundancy rule 'LANG:EN, T, FIN → EPP', which would generate an EPP feature to all finite T elements in English during lexical retrieval. This requires that each input sentence is categorized as belonging to some language, guiding the retrieval of functional items.

While this system can be used to describe crosslinguistic variation, it fails to capture the fact that many language-specific lexical redundancy rules apply in clusters – their basic parametric quality. In Finnish, for example, there is a generalization according to which any phrase that is pied-piped to the scope position at SpecCP must have the operator feature at its edge (Huhmarniemi 2012). This means that all functional items that head pied-piped phrases in Finnish must project the required syntactic specifier position, a behavior that is absent in English. Unless we structure the lexicon in such a way that these properties are honoured, either Finnish or English operator constructions will be derived wrongly. While it is indeed possible to describe such behavior in individual lexical redundancy rules, the underlying assumption would then be that they could be distributed also randomly throughout the functional lexicon, which does not seem to be the case. In other words, lexical rules are stipulated formulas – there is no natural way to capture properties that cluster together or exhibit some form of internal logic or a general pattern. In this sense the third layer architecture was not entirely incorrect; it forced the processing into a narrowly defined pathway, even though this forcing looked mysterious.

The problem remains unsolved. It is connected to a large problem: what ultimately determines the possible forms of the lexical redundancy rules? Arbitrary lexical rules are unrealistic also from the point of view of language acquisition. It is not realistic to assume that the child generates arbitrary redundancy rules into an empty storage medium as a function of linguistic experience; the system must be constrained by something. The principles and parameters framework used grammatical parameters to achieve the required constraints, but what else expect some type of ad hoc code could force the grammar into some parametric space? A brute ad hoc parametrization of the grammatical rules or algorithms themselves would not solve this problem since the parametric properties must be reflected also in the lexicon.

These questions raise another problem: how to implement the notion of 'language of a sentence' and direct the processing of the sentence to a correct model/lexicon. This is currently addressed by creating separate models for speakers of all languages present in the lexicon, and then by selecting the model based on the language of the words that appear in each input sentence. There is a separate language guesser module that accomplishes this. Thus, an input sentence that contains mostly English words will be directed for a parser model that uses the English functional lexicon, hence to a parser that we can think of "modelling an English speaker." This means that even if the input sentence would contain some words from Finnish, it would still get processed by using the English functional lexicon/parametrization; it is not possible to shift the language in the middle of the sentence.

3.6 Left peripheral C-features in Finnish

Finnish data suggest that the long head movement option is restricted to cases of left peripheral C-features, themselves unified by the fact that they are associated with the left peripheral C-domain, both in root and non-root contexts. This was accomplished technically by associating overt C-features with operator feature components in the lexicon, which then trigger Ā-reconstruction and the operator-variable interpretation seen in the output files and generated by the semantic system(s). These assumptions were sufficient to calculate the present dataset, but they do not exhaust syntactic and semantic behaviours of the C-features themselves. This matter was examined in more detail in another study (Brattico 2021a) that was in turn based on an earlier investigation by (Brattico et al. 2013). I will recapitulate the main points of that study and then move on to discuss some additional empirical facts.

The overall cognitive architecture handling grammaticalized features, including features involved in the C-domain, is based on the idea that the syntactic processing pathway (narrow syntax) generates “instructions” for semantic systems, where the term “instruction” is interpreted as referring to feature types that some cognitive module can respond to. Thus, the operator-variable system reacts to operator features that are distinguished from other features by their type marking. In order to model the behavior of the whole C-domain in Finnish, three feature types were required: force features (e.g., declarative, interrogative, imperative), which are involved in selection and compositional semantics; discourse features, which are interpreted by a pragmatic pathway; and operator features that are the subject matter of the present study and which were first modelled in (Brattico and Chesi 2020). This architecture is illustrated by Figure 13, with all three relevant feature types are listed in Table 2 (Brattico 2021a).

Table 2. Three types of features of Finnish left periphery (Brattico 2021a).

Force feature	Operator feature	Discourse feature	Examples
+	–	–	Pure force features: local Aux-to-C movement for interrogativization, *that* complementizer
–	+	–	Pure operator feature: relativization, possibly also internal *wh*-movement
–	–	+	Pure discourse feature: *wh*-features in connection with an echo-question
+	+	–	Movement plus clause typing: standard embedded interrogativization
+	–	+	Verb movement in a root imperative clause triggered by discourse feature
–	+	+	Left peripheral discourse features: *hAn*, *pA* particles; complex cartographies
+	+	+	Movement, clause typing and discourse together: root interrogativization

A force feature is connected to clause typing and affects the force of the clause (e.g., interrogative, imperative, declarative). They are involved in selection at LF interface, which can perhaps be identified with compositional semantics and/or a thematic theory of some type.[44] An operator feature is interpreted by the operator-variable module and involves the notion of propositional scope, often but not necessarily expressed by movement. This module (alone) understands Ā-dependencies. Discourse features are interpreted by the pragmatic pathway and are interpreted against a communicative move involving a thinker, proposition, and a propositional attitude, making them root phenomena.

All sentence examples of long head movement examined in this study contained overt forms that were associated with an operator feature, whereas examples of local head movement were connected with force (imperative, interrogativization) or root-level discourse features. Any overt form can be associated with zero to several features from each category. For example, an embedded interrogative pronoun or head is interpreted as containing an interrogative operator feature, an interrogative force feature, but no interrogative discourse feature as it does not represent an interrogative move by the speaker; a root interrogative typically has all three features. It represents interrogative clause type, interrogative communication, and an operator. We can now provide a classification of the Finnish overt C-features appearing in the sentence examples in the main article, by using the features in Table 2. This classification is provided in Table 3.

[44] Projection principle and selection operate at the LF interface, thus there is currently no external module that handles them.

Table 3. Summary of properties of left peripheral C-features in Finnish

	Force feature/ clause typing	Operator feature	Discourse int (root)	In situ option
Root imperative	+	–	+	–
Contrast by prosodic stress	–	+	+	+
Wh-feature	+ (not in situ)	+	+	+
Relative pronoun	–	+	–	–
Yes/no particle -*kO*	+ (not in situ)	+	+	–
Familiarity topic -*hAn*	–	+	+	–
Polarity topic -*pA*	–	+	+	–

Imperatives were excluded from both studies, but they are formed in Finnish by fronting a special imperative form of the main finite element (auxiliary, verb, negation) to the C-domain. Only local movement is possible. The imperative has an additional discourse interpretation in root contexts, in that the speaker is making an imperative communicative move; this interpretation is absent from embedded contexts. Since the imperative is associated with a special imperative form, the imperative feature can be listed directly in the lexicon and recognized directly from the input.

The reason discourse interpretation is absent in all embedded contexts, including embedded imperatives, is because the pragmatical pathway that responds to discourse features does not have access to embedded domains.[45] It operates with the notions of thinker (speaker) and main proposition and reads and interprets the discourse features from the latter. Interpretation of force and operator features, in

[45] Perhaps it does not access even recursion. The larger question here is if the language-external modules, all or some, access recursion independent of language.

contrast, are not limited to root contexts. The implication of these assumptions is that although all interrogative pronouns are linked with a discourse feature in the lexicon, they do not always trigger reactions inside the pragmatic pathway (see Figure 26) and hence do not have any semantic effect. Therefore, a lexical feature may remain inert with respect to semantic interpretation if the module that reacts to it has no access to the syntactic structure in which it occurs. This is currently possible.

Prosodic stress expresses contrast or contrastive focus in Finnish and English. Finnish employs two strategies, movement (37)(a) and in situ (b).

(37) a.	MYY-DÄ	Pekka	halua-a	_	kaiken	omaisuutensa!
	sell.FOC-A/INF	Pekka	want-PRS.3SG		all	possessions
b.	Pekka	halua-a	MYY-DÄ		kaiken	omaisuutensa!
	Pekka	want-PRS.3SG	sell.FOC-A/INF		all	possessions

These sentences differ in that the scope of (a) is unambiguous, whereas in (b) the scope could be ambiguous if the sentence was inside an embedded context. In both cases, the interpretation, generated on the basis of the LF interface representation by the operator-variable module, is (38).

(38) C(FOC, FIN) Pekka wants to sell(FOC) all possessions

The operator-variable module reads the operator [FOC] from the prosodically stressed predicate at LF interface (after transfer and reconstruction) and binds it with the scope. If the stressed operator was reconstructed, reconstruction leaves a copy of the feature at the surface site which will then mark the scope; if scope information is missing, any c-commanding finite head can be selected (and are returned as a list). The in situ strategy must be specifically licensed by a lexical feature, as it is not available in connection with all C-features. In English, only the in situ strategy is available with the exception of local T-to-C movement.

Notice that most examples of this type occurring in the main article were translated by using clefting (39)a, but an alternative would be a sentence with prosodic stress on an in situ element (b).

(39) a. It was selling (and not loaning) that Pekka wanted to do with all his possessions.

b. Pekka wanted to SELL (not loan) his possessions!

The operator-variable module targets the lexical content of the predicate ‘to sell’, hence there is a presupposed context in which some other predicate, such as ‘to loan’, was used. If the stressed element is inside a pied-piped phrase, the mechanism is the same. The pied-piped phrase is reconstructed, and the targeted predicate is linked with the main clause scope generated during transfer and reconstruction.

(40) a.	TÄMÄN	talon	C	Pekka	osti	[tämän…]	(ei tuota)!
	THIS	house		Pekka	bought	this	(not that one)!

‘Pekka bought THIS house, not that one!’

b.	Tämän	TALON	C	Pekka	osti	[…talon]	(ei ovea)!
	this	HOUSE		Pekka	bought	house	(not door)!

‘Pekka bought this HOUSE, not (just) the door!’

Wh-features can be associated with lexical elements in several major lexical categories, but the case for verbs is not clear. Are there interrogative verbs? In English, the fronted auxiliary is linked with interrogativization, but the way this linkage is achieved remains unclear and has not been modelled, apart from trivial strategies of assuming ambiguity or surface disambiguation as was done in (Brattico and Chesi 2020). In Finnish, the wh-feature is normally linked with a force feature, operator feature and a discourse feature (the latter activated only in root context), but also the in situ strategy is available and leads into the echo-reading. Echo reading results when the operator-variable module links the operator with finite scope while the interrogative force feature is not generated due to the lack of reconstruction. The in situ option is licensed by a lexical feature associated with Finnish interrogativization. Here the in situ interrogative pronoun is bind by finite T.

(41)

Pekka	ihaile-e	ketä?
Pekka	admire-PRS.3SG	who.PAR

(Only echo-reading)

The yes/no particle -kO expresses yes/no interrogativization and differs from the wh-feature in two ways: it does not have the in situ strategy, therefore scope must be generated by reconstruction, and it can be combined with almost any element including verbs. Indeed, most of the example sentences in the main article used the yes/no particle due to its straightforward properties. Notice that because the yes/no question particle can be suffixed both with verbs and other lexical elements, it exhibits both head movement (both HMC and LHM) and ordinary pied-piping.

(42)

a.	Myy-kö	Pekka	_	kaiken	omaisuutensa?		
	sell.PRS.3SG-Q	Pekka		all	possessions		
	'Was it selling that Pekka does with all his possessions?'						
b.	Myy-dä-kö	Pekka	halua-a	_	kaiken	omaisuutensa?	
	sell-A/INF-Q	Pekka	want-PRS.3SG		all	possessions	
	'Was it selling that Pekka wants to do with all his possessions?'						
c.	[Kaiken	omaisuutensa-ko]	Pekka	halua-a	myy-dä	_?	
	all	possessions-Q	Pekka	want-PRS.3SG	sell-A/INF		
d.	[Kaiken-ko	omaisuuden]	Pekka	halua-a	myy-dä	_?	
	all-Q	possessions	Pekka	want-PRS.3SG	sell-A/INF		
e.	*Pekka	halua-a	myy-dä-kö	kaiken	omaisuutensa?		
	Pekka	want-PRS.3SG	sell-A/INF-Q	all	possessions		

As above, the yes/no particle targets a specific predicate to which it is attached as a suffix (e.g., 'sells', 'to sell', 'possessions', 'all'). That predicate is interpreted as an operator at the reconstructed position by the operator-variable module and linked with scope.

It should be noted that sentences involving HMC compliant movement (a) are ambiguous. This sentence can be interpreted either as targeting a specific finite predicate 'sells' or as making a yes/no question concerning the whole proposition. The latter interpretation is less likely (or perhaps absent) in (b). This difference becomes clearer when the moved element is an auxiliary. A fronted auxiliary can only create a verum focus interpretation, while long head movement is also absent, suggesting that the Ā-operator interpretation is absent. In the main article it was speculated that this is because the operator-variable module needs lexical content in order to create an interpretation. The restriction has not yet been implemented at the level of computer code.

It is possible to combine the yes/no particle with an interrogative pronoun. The effect is unclear: both interrogatives (*ketä* 'who.PAR', *ketä-kö* 'who.PAR-Q') can be used in almost any context with the exception that the presence of the yes/no particle eliminates the in situ strategy, suggesting that this involves some type of feature redundancy.

Finally, the familiarity topic particle -*hAn* and the polarity topic particle -*pA* behave like the yes/no question particle -*kO* with the exception that they do not have force features, which also makes it possible to combine them with suffixes which do have force implications. Their interpretation relies on discourse, with -*hAn* being associated in some cases and contexts with familiarity topic (something familiar from context) and -*pA* reinforcing the polarity of the clause. The exact semantics are murky; quite likely these particles have multiple uses. Both are operators and do not have the in situ option.

(43) a. Merja-a-han Pekka ihailee _.
Merja-PAR-FAM Pekka admires

'It is Merja (who we presuppose to know) that Pekka admires.'

b. Merja-a-pa Pekka ihailee _.
Merja-PAR-POL Pekka admires

'But it is Merja who Pekka admires.'

Both particles are associated discourse interpretation and cannot therefore be combined with the relative pronoun, as relative clauses are not interpreted as full saturated propositions. The operator module links both particles with the finite scope of the relative clause, but the pragmatic module will fail to assign them with a thinker and proposition. Both particles can be combined with each other as well as with interrogative pronouns, imperative verbs and the yes/no particle. It is unclear what their semantic effects are when combined with other C-features. Sentences (44)a-b are very close if not complete synonyms, perhaps the only clear effect being that the presence of the familiarity topic particle in (b) eliminates the in situ strategy and hence the corresponding echo-reading.

(44) a.	Kuka	ihailee	Merjaa?
	who	admires	Merja
	’Who admires Merja?’		
b.	Kuka-han	ihailee	Merjaa?
	who-FAM	admires	Merja
	‘Who admires Merja?’		

Perhaps an additional difference is that (b) is not felicitous in a formal context, it has an aura of informality or colloquiality. An alternative translation could be ‘who could it be that admires Merja?’ or ‘guess who admires Merja?’ Regardless, all C-features are processed independently. Thus, sentence (b) involves two operators WH and FAM each bound by its own clause-mate finite C.

The order at which the various C-features can be combined with a root stem is regulated by rules that were not modelled. This question was left unsolved, because it was and still is unclear whether the restrictions are part of the lexico, morphology, syntax or semantics (or some combination of these). The model therefore accepts both *Pekka-ko-han* ‘Pekka-Q-FAM’ and **Pekka-han-ko* ‘Pekka-FAM-Q’ with the same output. It will also (correctly?) accept ?*Pekka-han-han* ‘Pekka-FAM-FAM’ but also **Pekka-han-han-han* ‘Pekka-FAM-FAM-FAM’. Some possible combinations are borderline marginal, such as ??*Pekka-ko-pa* ‘Pekka-Q-POL’. This topic must be investigated further.

I will conclude this section by outlining few problems in elucidating the semantic interpretations for the left peripheral C-features and creating English translations for them. The most natural linguistic device for expressing in English the meaning of Finnish LHM constructions and the discourse-based left peripheral particles appears to be prosody. For example, when a complex infinitival is suffixed with the yes/no morpheme and fronted in Finnish (1), the corresponding English sentence would have the infinitival in situ but use special prosody indicting speaker's surprise for the choice of that particular predicate. Contrastive focus would be expressed by using heavy prosodic stress. The same is of course true of interrogativization and imperatives as well. The English clefting construction can in some cases substitute prosody to emphasize the content of the moved element, but the construction is different, and Finnish has a separate clefting construction that has different properties (see footnote 4). Another problem is ambiguity. Consider (45).

(45)	Myy-hän	Pekka	_	omaisuutensa.
	sell.PRS.3SG-FAM	Pekka		possessions

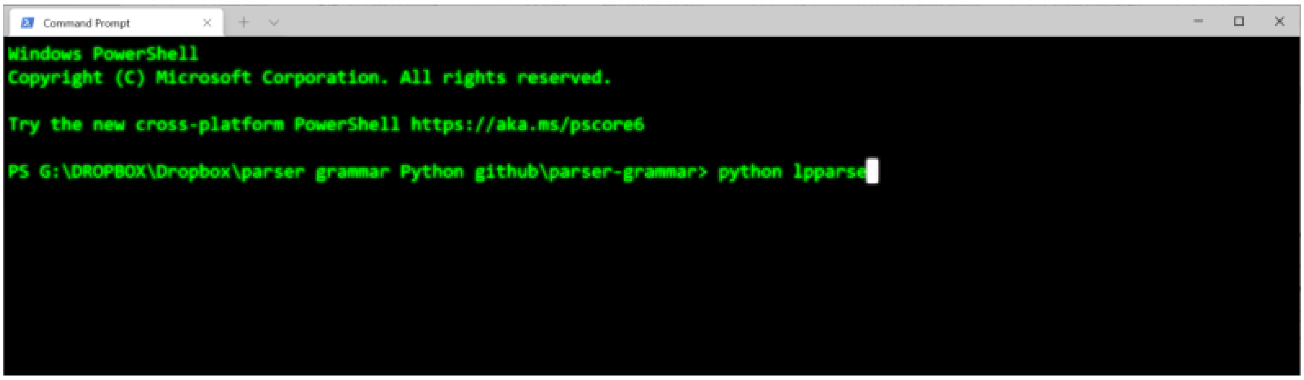
This sentence can be interpreted in several ways. If we apply (here implicit) prosodic stress to the fronted verb, then the interpretation is 'but Pekka does sell his possessions', that is, something in the previous context was contrasted. The most natural interpretation is that it was claimed that Pekka does not sell his possessions, which (45) counters. To me this interpretation emerges also if the stress is canonical. But the sentence can also be interpreted as an interrogative 'Does Pekka sell his possessions?' or 'Pekka sells his possessions, doesn't he?', especially if the interrogative interpretation is emphasized by prosody and the fronted element is *not* stressed. This interpretation appears to be based on the same mechanism generating interrogativization out of Aux fronting in English, because it disappears if the fronting is LHM.

3.7 Source code and source files

The implementation was done in Python programming language (version 3.x) without external libraries with the exception of the pyglet library that is used in drawing the phrase structure trees. Thus, if the user wants the algorithm to provide phrase structure trees (shown in this document), this library must be currently installed in the local machine alongside Python itself. The version of the comprehension algorithm that was used to simulate the data in this study is 9.0. Python can be installed by following the instructions at https://www.python.org/. It goes without saying that implementation details are subject to change when the model is developed.

The source code is stored currently at https://github.com/pajubrat/parser-grammar and can be installed from there into a local machine by following regular installation instructions, also provided in the software documentation (Brattico 2019a) that is available in the /docs subfolder. Standard installation (not cloning) means that the user copies the material from the above address to a local machine, into some local directory that the user can select freely. That directory then becomes the root directory of the program. The source code that the user downloads from the repository consists of the latest version of the program, also called the master branch. This is typically a functional version of the code and should do something meaningful when downloaded; usually it processed all sentences from the *default corpus*. The exact version that was used in this study is stored in the branch *long-head-movement-(Study-4d)*.[46] Branches *4e*, *f* and so on are used if improvements or corrections are required, or if the same dataset is processed by a more advanced model. If we consider the algorithm as a rigorous justification for the hypothesis, then it is sufficient that an algorithm that incorporates the proposed hypothesis and calculates all the data exists. An actual implementation is a realization of this requirement.

The source code, when downloaded to the local machine, can be executed by writing “python lpparse” into the command prompt that is in the selected root directory in the local machine. The user must have Python installed and configured on the local machine.



[46] Thus, to replicate the present study the user should download the code and files from this branch.

Figure 16. Execution of the program. The program package must be installed into some directory in the local machine, freely selected and/or created by the user. Thus, the path shown above corresponds to the location where it is in one of my own local machines.

This will execute the program package. The source code consists of ordinary text files that are read by a Python interpreter, which translates them into concrete actions performed by the computer (calculating results, reading, and writing files). Hence the user can examine the program with any text editor, such as Windows notepad. The source code is in the folder /lpparse. The .py extension refers to a Python source code module. For example, to examine how head reconstruction works, the user can open the module head_reconstruction.py with Windows notepad and find the actual code.

When then user writes the above command into the command prompt, the program will attempt to read the __main__.py module from the folder /lpparse, which then executes the master script called main.py.[47] This module in turn has a function run_study(), which runs one whole study. The code for run_study() is provided in Figure 17.

[47] The module __main__.py parses command line arguments. It can be used to divert processing to special purpose programs or functions, such as macro scripts that run several studies.

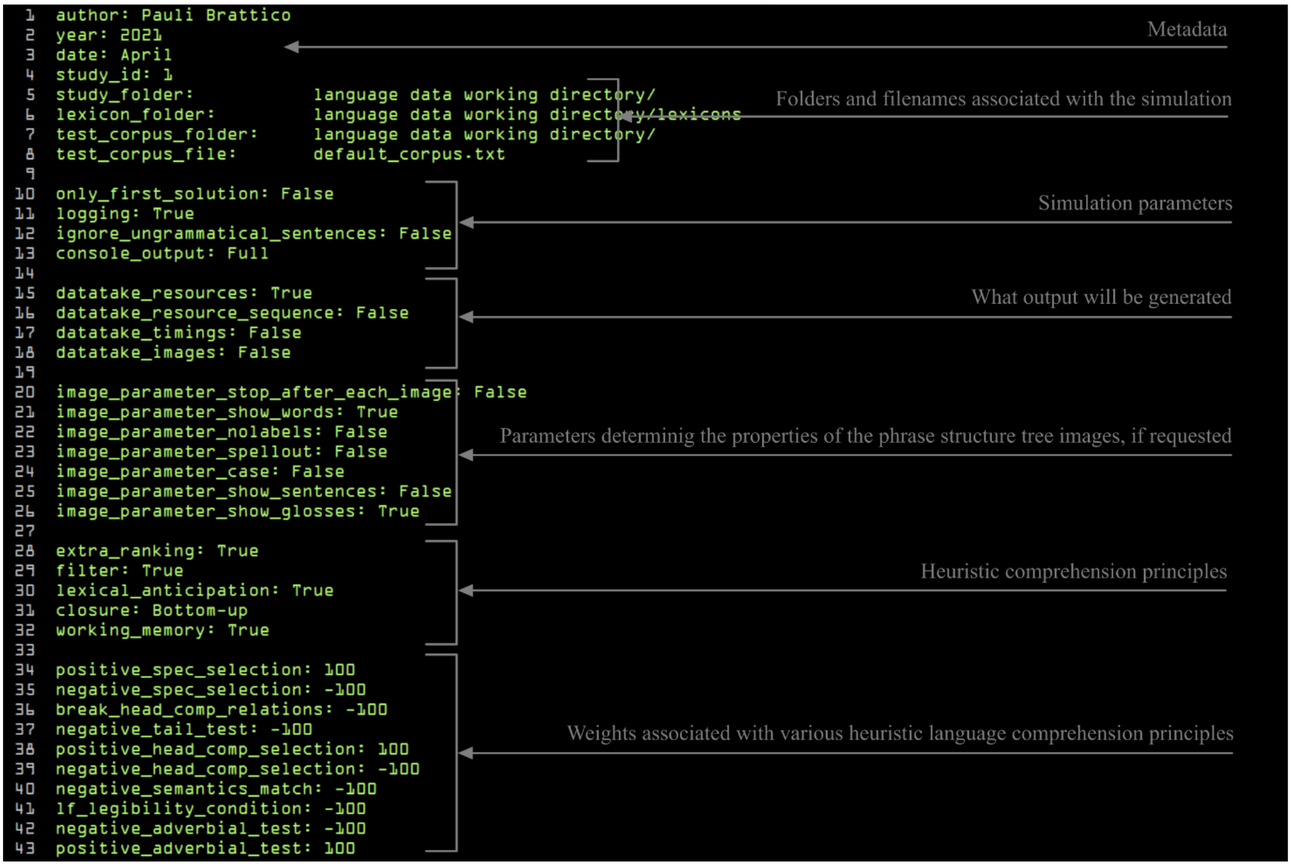
```
6  def run_study(args):
7      sentence = args.get('sentence', '')
8
9      local_file_system = LocalFileSystem()
10     local_file_system.initialize(args)
11     local_file_system.configure_logging()
12
13     parser_for = {}
14     lang_guesser = LanguageGuesser(local_file_system.external_sources["lexicon_file_name"])
15     for language in lang_guesser.languages:
16         parser_for[language] = LinearPhaseParser(local_file_system, language)
17         parser_for[language].initialize()
18
19     if not sentence:
20         sentences_to_parse = [(sentence, group, part_of_conversation)
21                               for (sentence, group, part_of_conversation)
22                               in local_file_system.read_test_corpus()]
23     else:
24         sentences_to_parse = [([word.strip() for word in sentence.split()], '1', False)]
25
26     sentence_number = 1
27     for sentence, experimental_group, part_of_conversation in sentences_to_parse:
28         if not is_comment(sentence):
29             language = lang_guesser.guess_language(sentence)
30             local_file_system.print_sentence_to_console(sentence_number, sentence)
31             parser_for[language].parse(sentence_number, sentence)
32             local_file_system.save_output(parser_for[language],
33                                           sentence_number,
34                                           sentence,
35                                           experimental_group,
36                                           part_of_conversation)
37             if not part_of_conversation:
38                 parser_for[language].narrow_semantics.global_cognition.end_conversation()
39             sentence_number = sentence_number + 1
40         else:
41             local_file_system.parse_and_analyze_comment(sentence)
42             local_file_system.write_comment_line(sentence)
43
44     # Finish processing
45     local_file_system.save_surface_vocabulary(parser_for["LANG:EN"].lexicon.surface_vocabulary)
46     local_file_system.close_all_output_files()
```

Figure 17. The code for the main.py in the current master branch. The logic is explained in the main text.

Lines 9-11 initialize the simulation. The most important function of these operations is to read the configuration file that sets a variety of internal parameters for the simulation. The configurational file is currently named config_study.txt and is located in the root directory (missing files and parameters are substituted by default behavior, ignored here).[48] Figure 18 shows a screenshot of this file at present writing together with explanations of the most important parameters.

---

[48] Any parameter can also be given as a command line argument, which are convenient when writing scripts that execute several studies.



```
 1  author: Pauli Brattico
 2  year: 2021
 3  date: April
 4  study_id: 1
 5  study_folder:          language data working directory/
 6  lexicon_folder:        language data working directory/lexicons
 7  test_corpus_folder:    language data working directory/
 8  test_corpus_file:      default_corpus.txt
 9
10  only_first_solution: False
11  logging: True
12  ignore_ungrammatical_sentences: False
13  console_output: Full
14
15  datatake_resources: True
16  datatake_resource_sequence: False
17  datatake_timings: False
18  datatake_images: False
19
20  image_parameter_stop_after_each_image: False
21  image_parameter_show_words: True
22  image_parameter_nolabels: False
23  image_parameter_spellout: False
24  image_parameter_case: False
25  image_parameter_show_sentences: False
26  image_parameter_show_glosses: True
27
28  extra_ranking: True
29  filter: True
30  lexical_anticipation: True
31  closure: Bottom-up
32  working_memory: True
33
34  positive_spec_selection: 100
35  negative_spec_selection: -100
36  break_head_comp_relations: -100
37  negative_tail_test: -100
38  positive_head_comp_selection: 100
39  negative_head_comp_selection: -100
40  negative_semantics_match: -100
41  lf_legibility_condition: -100
42  negative_adverbial_test: -100
43  positive_adverbial_test: 100
```


Figure 18. Screenshot from the configuration file determining the input parameters.

The most important parameters are the file and folder names (lines 5-8) which the program uses to read all inputs (test corpus and lexicons) and where it produces the output (parameter study_folder). Each individual study should use its own folder that contains all the required materials, perhaps with the exclusion of the lexicons that I currently load form a master lexicon folder and which remains mostly the same from one study to the next.[49]

Next the function prepares a "brain model" for the speaker of each language present in the lexicon (lines 13-17) and initializes them. These models are matched with the languages of the test sentences at runtime. Finnish sentences will be processed by a Finnish speaker, English sentences by English

[49] The lexicon used in connection with any particular study should still be stored together with the rest of the files.

speaker, and so on. Bilingual speakers are assumed to shift from one model to another based on the input. Separate models are required for each language because the properties of functional items differ from one language to the next (at least in the current version). The program then reads the input sentences from the test corpus (lines 22-24), as defined in the configuration file (Figure 19), and sends them, one by one, to the parser model based on the language (lines 27-42).

A parser model is defined in the class LinearPhaseParser, which uses parse() for parsing the sentence it receives. This function is called for each input sentence (line 31). This module and its functions define what could be characterised as executive control for the parsing process, thus step-by-step instructions on what to do with each input and in which order. All submodules of the system (lexico-morphological module, transfer, syntax-semantic interface and others) are instantiated inside this class from their own modules, so that the LinearPhaseParser class can be thought of as a general model of the speaker-hearer (of each language).[50] It will also perform many auxiliary tasks, such as recording performance metrics. The parsing function performs necessary initialization and then calls a recursive parsing function (parse_new_item()) which consumes words from the input and sends them through the pipeline as defined in Figure 2. The phrase structure argument of this function represents the current syntactic analysis in current working memory. Once all words have been consumed, the function attempts to complete processing (function complete_processing()) which tries to transfer the result into the syntax-semantics interface, creates a semantic interpretation if successful and performs several logging functions. Control is then handed back to the main loop which backtracks (exits from the recursion in a well-defined order).

Head reconstruction is defined in its own module head_reconstructions.py and is part of the transfer (transfer.py). The core content of the analysis is contained in two functions. One is reconstruct_head_movement(), which performs minimal search, detects complex heads and reconstructs them. It is shown in Figure 19.

---

[50] An alternative is an architecture in which the parser module is embedded inside some still more large executive system. This could be used if the output of the syntactic pathway were used for some other purpose, such as reasoning or communication.

```python
59  def reconstruct_head_movement(self, phrase_structure):
60      """
61      Reconstructs all head movement from phrase structure [ps]
62
63      Performs minimal search on the right edge of the phrase structure and detects complex left heads.
64      Once detected, the head is reconstructed. Once reconstructed, the algorithm continues the process
65      from where it left. Minimal search is terminated by intervention feature that is determined by
66      the type of the head, which is used to explain why C-heads tolerate long distance reconstruction
67      while A-heads do not. This implements rules (43A-B).
68      """
69      # ----------------- minimal search ------------------------------------------#
70      for node in phrase_structure:
71          if self.detect_complex_head(node):
72              complex_head = self.detect_complex_head(node)
73              log(f'Reconstruct {complex_head.right_const} from within {complex_head}...')
74              intervention_feature = self.determine_intervention_feature(complex_head)
75              # Inverse head chain is created here (Rule 43A-B)
76              self.create_head_chain(complex_head, self.get_affix_out(complex_head), intervention_feature)
77              log(f'={phrase_structure.top()}...')
78      #-------------------------------------------------------------------------------#
79      return phrase_structure.top()
```

Figure 19. Implementation of minimal search.

The algorithm implements minimal search (line 70) by enumerating the phrase structure on its right edge by following labelling, and then detects complex heads X(Y) (line 71). Line 74 defines the intervention feature (notice the stipulative character of this component) and line 76 triggers head chain creation which reconstructs the head X(Y) → [X [...Y...]]. Head reconstruction is defined in function create_head_chain() and is provided in Figure 20.

```
98  def create_head_chain(self, complex_head, affix, intervention_feature):
99      """
100     Creates a head reconstruction chain for one complex head [complex_head] H.
101
102     If H has no sister XP that reconstruction could use as a target, then H will be
103     reconstructed into its own sister, H(X) = [H X]. Else a minimal search is called for
104     the right edge of its sister XP. If minimal search encounter intervention feature,
105     it will trigger the last resort reconstruction. If it is allowed to continue, it will
106     try to merge the reconstructed affix into this position and, if the operation succeeds,
107     the affix will remain at that position. If the operation does not succeed, the affix
108     will be removed and the procedure continues. If it reaches the end of the structure,
109     one more solution will be attempted and, if that does not work, then last resort.
110     """
111     if self.no_structure_for_reconstruction(complex_head):
112         self.reconstruct_to_sister(complex_head, affix)
113     else:
114         phrase_structure = complex_head.sister()
115         node = None
116         # -------------- minimal search --------------------------------------------#
117         for node in phrase_structure:
118             if self.causes_intervention(node, intervention_feature, phrase_structure):
119                 self.last_resort(phrase_structure, affix)
120                 return
121             node.merge_1(affix, 'left')
122             if self.reconstruction_is_successful(affix):
123                 self.controlling_parser_process.consume_resources("Move Head")
124                 return
125             affix.remove()
126         # --------------------------------------------------------------------------#
127         # Still no solution
128         if self.try_manipulate_bottom_node(node, affix, intervention_feature):
129             return
130         else:
131             affix.remove()
132         self.last_resort(phrase_structure, affix)
```

Figure 20. Implementation of the head reconstruction algorithm.

Lines 111-112 implement the special edge case in which the complex head is itself the bottom right constituent, in which case there is no structure that could be targeted; it targets the sister (X(Y) → [X Y]). We perform minimal search (lines 116-126) and examine each possible position for the head (line 121) and evaluate if the position satisfies the conditions for gap (line 122). If it does, the head remains in that position and the operation terminates (line 124); if not, search continues. Intervention is calculated at line 118, which, if it happens, causes the algorithm to implement the last resort solution X(Y) → [X [Y...]] defined in its own function. Lines 126-131 handle an edge case in which the gap position exists inside the bottom right node, for example, if Y must be selected by Z but Z is still inside the bottom right node, W(Z).

In addition to the source code in the /lpparse directory, the simulation uses several input files and generates several output files. These files can be found also from the master branch. The *test corpus* is in the file LHM_corpus.txt which is located in the /language data working directory/study-4_d-LHM subfolder. The same folder contains also several output files as follows: LHM_corpus_grammaticality_judgments_FINAL.txt contains input sentences together with the grammaticality judgments predicted by the model; LHM_corpus_log_FINAL.txt contains the complete derivational log; LHM_corpus_resources_FINAL.txt contains data concerning the computational resources that were consumed per each input sentence; LHM_corpus_results_FINAL.txt contains summaries of the main results (syntactic analyses, semantic interpretation, resource consumption and extralinguistic semantic objects). Phrase structure images generated by the model are provided in a compressed folder *Phrase Structure images*.

4 RESULTS

4.1 Introduction

In this section I will examine the results in some more detail. I will limit the discussion to two concerns, observational and descriptive adequacy. The former is concerned with grammaticality judgments only. An observationally adequate model must partition the set of input sentences correctly into grammatical (possible) and ungrammatical (impossible). Descriptive adequacy imposes a further requirement for the grammatical sentences: the semantic interpretations assigned to them by the model must match with semantic intuitions elicited from native speakers, and the internal representations generated by the algorithm must be plausible under a linguistic and/or psycholinguistic theory. Explanatory adequacy and psycholinguistic plausibility are ignored in this report, although they played a role in the construction of the hypothesis.

Notice the expression “plausible under a linguistic theory.” Since we do not know what the properties of the real language faculty are, we can demand neither complete agreement with a theory nor plausibility under any specific approach.

4.2 Observational adequacy

Observational adequacy is evaluated by comparing grammaticality judgments of the model with those elicited from native speakers. A feasible and reliable way of performing the required comparisons is by preparing a file containing the native speaker judgments and by comparing that file with the model output by automatic file comparison tools. The comparison reached 100% match with the final version.[51]

It is crucial that no further modification to the theory or hypothesis comes at the cost of undoing previously established results. We expect that future development can cause issues, but not wholesale undoing that cannot be corrected without a complete rewriting of the hypothesis. To this aim, most of the head movement data examined in this study were added to a *default corpus* that hosts examples of all construction types examined so far with the model. When further modifications to the hypothesis are eventually considered, developed, and tested, most of the head movement data will then be automatically included into this default test set.[52]

The fact that the model reached observational adequacy as judged by the author has often raised concerns that a confirmation bias could play a role when the model is evaluated. Moreover, some of the data is hard to judge, raising concerns that it might be judged in a biased way to give an unfair advantage to the model and/or by downplaying possible problems. This criticism, while legitimate in all theoretical linguistics, misconstrues the verification process. The model is not verified by running one critical experiment with a model that appears out of the blue as a complete product. It is developed by a trial-and-error procedure, in which various versions of model are compared with the data by simulation and

---

[51] The two files are LHM_corpus_grammaticality_judgments_FINAL.txt (model output) and LHM_corpus_grammaticality_judgments_NATIVE.txt (native speaker).

[52] The current version of the model is always the latest in the master branch, and the default corpus that I use in its development is in the directory /language data working directory. The user can therefore test the current version with most of the head movement data by running the algorithm over the default corpus and examining the results for the relevant test sentences. At present writing the relevant test sentences are in groups 3.4 (left peripheral C-feature tests), 3.5 (selection and C-features), 3.6 (in situ operators), 8.4 (wrong head orders) and 9 (head movement). As this article goes to print, the current version (10) runs the binding corpus; head reconstruction has not been tested but will be in the near future.

then revised accordingly. This process continues until a good or perfect match is reached. Thus, the final model was preceded by several hypotheses and models which produced either completely wrong results or results that were only partially correct. Many of these attempts are, moreover, available in the source code repository, and some simulation results with deficient models are available as well. This development shows that wrong output, rather than data munging, guided discovery. Problems of data interpretations represents a relatively minor concern when the main goal is often to get the formula to calculate even the core of the data correctly. Of course, this does not mean that disagreement over data cannot exist. The model must be adjusted to reflect data that is considered robust and correct by a consensus view. Moreover, unclear data should be delegated to a less critical role.

4.3 Descriptive adequacy

4.3.1 Baseline tests (Group 1)

4.3.1.1 Introduction

The purpose of the baseline tests is to establish that the grammatical mechanisms and lexical elements presupposed in the study work as intended. These tests probed valency variations (intransitive, transitive and ditransitive declarative clauses, Group 1.1.2), computation of various additional constructions and elements required to perform the eventual head movement tests (e.g., negation; aux + past participle; want + A-infinitival; modal; combinations of negation, modal and auxiliary; negation plus want + A-infinitival; infinitival with thematic genitive subject; noun head + A-infinitival constructions, Groups 1.1.3-1.1.12), noncanonical OVS orders (Group 1.1.13), pro-drop sentences (abundant in Finnish, Group 1.1.14), processing of locative PP arguments (Group 1.1.15) and the computation of all C*-features and their combinations in connection with phrasal operator movement (1.1.16). These were followed by few English sentences representing languages without long head movement. Noncanonical word orders and pro-drop were not discussed in the main article; they were included to verify that the proposed head reconstruction mechanism does not undo previous results accomplished with such structures.

Most test sentences use the same lexical elements, which sometimes create pragmatically implausible examples. On the other hand, the study is more reliable if the same lexical items are recycled from one example to the next. By doing this, corrections to the key lexical elements generalize automatically to the whole study, and there are less room for mistakes and lexical mismatches. When evaluating grammaticality in a syntactic study, it is important to replace the placeholder words with entries that are pragmatically felicitous.

4.3.1.2 Intransitive, transitive, and ditransitive finite clauses (Group 1.1.2)

We begin by considering the derivation of a declarative intransitive clause (46). Numbers such as #3 refer to the sentence identifiers in the output of the model.

(46) Pekka istu-u. (#3)
Pekka.NOM sit-PRS.3SG
'Pekka sits.'

The spellout structure generated by the model for this input sentence is [D(N) T(V)] corresponding transparently to the surface string (*Pekka* = D(N) = full argument, *istuu* = T(V) = tensed finite verb). This is due to sinking. Head reconstruction is applied during transfer. Because each left branch is transferred independently (Brattico and Chesi 2020), the representation appearing just before the final transfer in the derivational log is [[D *Pekka*] T(V)] (line 614), where the left branch has been transferred earlier and hence the N-component has been extracted from D(N) by trivial reconstruction (D(N) $\rightarrow$ [$_{DP}$ D(_$_1$) N$_1$]). The term "trivial head reconstruction" denotes standard head reconstruction applied to a sole X(Y) constituent (i.e. X(Y) $\rightarrow$ [X(_$_1$) Y$_1$]). This operation is shown on line 603, thus it occurs few steps before the final transfer (lines 614-631).

```
598   3. Consume "T": D(N) + T
599       Working memory operation...1 nodes currently in active memory.
600       Filtering and ranking merge sites...Filtering...Done. Ranking...Bottom-up baseline ranking...+Spec select
601       Results:
602       |   |   (1) D(N)
603       Now exploring solution [D(N) + T]...Transferring left branch D(N)...(80ms) Result: [[D Pekka] T]...Done.
604
605       Next item: "istu-". Lexical retrieval...(55ms) Done.(60ms)
606       Item enters active working memory.
607
608   4. Consume "istuu": [[D Pekka] T] + istuu
609       One solution due to sinking.(65ms) Done.
610       Results:
611       |   |   (1) T
612       Sinking istuu into T = [[D Pekka] T(V)](70ms)
613   ----------------------------------------------------------------------------------------------------------
614   Trying spellout structure  [[D Pekka] T(V)]
615       Checking surface conditions...Done.
616       Transferring to LF...(75ms)
617               1. Head movement reconstruction...Reconstruct istuu from within T(V)...(80ms) =[[D Pekka] [T istu
618               |    = [[D Pekka] [T istuu]](80ms).
619               2. Feature processing...Done.
620               |    = [[D Pekka] [T istuu]](80ms).
621               3. Extraposition...Done.
622               |    = [[D Pekka] [T istuu]](80ms).
623               4. Floater movement reconstruction...[D Pekka] failed to tail [ARG] [FIN] [VAL] ...Dropping [D Pe
624               |    = [<D Pekka>:1 [T [<__>:1 istuu]]](100ms).
625               5. Phrasal movement reconstruction...Done.
626               |    = [<D Pekka>:1 [T [<__>:1 istuu]]](100ms).
627               6. Agreement reconstruction...(105ms) (105ms) "T" acquired PHI:NUM:SG by Agree-1 from <__>:1 insi
628               |    = [<D Pekka>:1 [T [<__>:1 istuu]]](105ms).
629               7. Last resort extraposition...(110ms) LF-interface test...Done.
630               |    = [<D Pekka>:1 [T [<__>:1 istuu]]](110ms).
631       Done.
```

Figure 24. A screenshot from the derivational log file generated by the algorithm used in this study, illustration how the derivation in broken down into phases. The left branch D(N) (line 603) is transferred before the final product (line 614). When the latter operation is activated, the input object contains a transferred and processed left branch DP = $[_{\mathrm{DP}}$ D *Pekka*]. The transfer operation shown on line 603 (D(N) $\rightarrow$ $[_{\mathrm{DP}}\ \mathrm{D}(\_{}_1)\ \mathrm{N}_1]$) contains the same sequence of operations than the final product (lines 617-630), but they are not recorded in order the avoid excessive logging. If the researcher wants to enable logging for all phase transfers, this has to be done currently at the level of code.

More generally it holds that because the derivation proceeds in phases, we do not necessarily find all operations of type X from the same place in the log; instead, we find clusters of these operations in places where some segment is transferred to the LF interface.

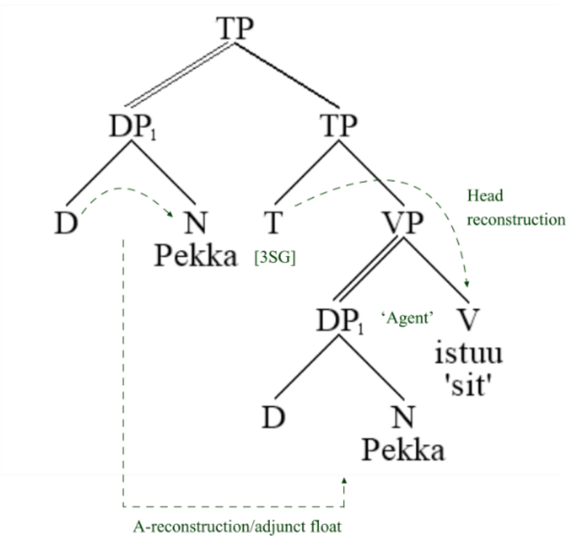
The verb *istu-u* ‘sit-PRS.3SG’ is decomposed into three elements in the lexicon: the verb stem ‘sit’, present tense (T) and the third person singular agreement suffix. The verb stem and the tense map into separate morphemes (i.e. V, T), whereas the agreement suffix is treated as an inflectional feature and is inserted inside T as an agreement feature (‘third person’, ‘singular’) during lexical streaming (Brattico 2021b)(lines 594-595). Inflectional features are interpreted as lexical features, not lexical items, the latter which are sets of the former (see line 595). The resulting representation [[D *Pekka*] T_{3sg}(V)] is then transferred to the LF interface (lines 617-629). The first operation (lines 617-618) shows the results of head reconstruction. Reconstruction is applied to T(V), which generates [$_{TP}$ DP [T(_$_1$) V$_1$]] (line 618).

The preverbal subject DP is reconstructed from SpecTP into SpecvP (lines 623-624) to yield [$_{TP}$ DP$_1$ [$_{TP}$ T [$_{VP}$ _$_1$ V]]]. The SpecTP position constitutes a *topic position* in Finnish and can be occupied by almost any constituent in the clause (Brattico 2019b, 2021a; Holmberg and Nikanne 2002; Huhmarniemi 2019; Vilkuna 1989, 1995). It need not be occupied by the grammatical subject. The SpecvP position is associated with thematic agents by the semantic component. The SpecTP-SpecvP reconstruction cannot, therefore, be implemented in Finnish by using local A-movement. The reconstructed constituent can travel considerable distance (for example, from SpecTP into the direct object position of an embedded infinitival clause). This is the reason local reconstruction is implemented by what is labelled in the log as "floater reconstruction" and not by standard A-reconstruction, the latter which is applied in languages like English (see sentence #84). Floater reconstruction, unlike A-reconstruction, looks at the overt case feature of the argument and uses it to select the reconstructed thematic position (Brattico 2020, 2021a). This assumption derives the Finnish free word order signature but notice that the exact same outcome (SpecTP $\rightarrow$ SpecvP) would result were the topic phrase reconstructed by standard A-reconstruction. Notice, in addition, that the fact that *Pekka* was in the topic position is recorded into the output files, which states that it constitutes a marked topic of course clause (line 88), while its interpretation as the agent of the event is visible on line 83 and is based on the reconstructed SpecvP position at LF interface. The same data is available in the derivational log file (lines 639-645). We want to use this information to check that the semantic interpretation matches with native speaker intuition. In this case it does. In general this means that one and the same argument may acquire several interpretations, here 'topic' and 'agent', depending on the syntactic positions is occupies during the various stages of the derivation.

A-reconstruction, which does not play a major role in the present study, has been difficult to capture precisely. Large-scale simulation studies are absent. The general approach in the current model is to use local successive-cyclic reconstruction.[53] There are few examples of this phenomenon in the dataset, analysed later.

Line 637 in the derivational log file reports that the LF interface object arriving from the syntactic parser component was accepted at 645ms stimulus onset. This number is calculated on the basis of what was elucidated in Section 3.1. It is irrelevant here, but worth noting since the derivational log file presents rather voluminous information of this type. The operations reported just before (lines 632-636) verify that the input object [$_{TP}$ DP$_1$ [$_{TP}$ T [$_{VP}$ _$_1$ V]]] satisfies LF legibility with this selection of lexical elements. The output is (47).

(47)



Notice the two head reconstruction operations D(N) → [$_{DP}$ D(_) N] and T(V) → [$_{TP}$ T V]. Because the topic is reconstructed by floater reconstruction, the target position SpecvP is selected not due to locality but because the nominative case feature at D can be checked by the finite T at this position. Case checking, while it does take place, will be ignored for now; notice, however, that if the direct object with a direct object case occupied the topic position, reconstruction would position it lower, as indeed required by empirical evidence (e.g., *Merja-a ihaile-e Pekka* 'Merja-PAR admire-PRS-3SG Pekka.NOM').

[53] I have worked with two definitions for A-reconstruction: (a) reconstruct to the specifier position of the next head, independent of what type of head it is, or (b) reconstruct to the next specifier position available. The latter makes A-reconstruction in principle unbounded in grammatical distance. Another problem is what to do if the specifier position or edge (including edge adjuncts) is already filled in.

The derivational log file contains several types of information that can be relevant in assessing the plausibility of the calculated outcome. Lines 652-659 list all lexical items present in the output and their lexical features when processing ended. This provides a snapshot of the lexical information that was used in the computations. Notice that it records only the final configuration; intermediate states are not reported and must be probed by separate simulations and/or ad hoc logging. The composition of this information was discussed in Section 3.5. Lines 660- under “semantic bookkeeping” list all semantic objects projected into existence while the sentence was interpreted semantically. This component is not checked for correctness. However, when the model reconstructs an operator in the context of some proposition, the proposition will be projected into existence into the semantic object inventory as a “propositional object.” The idea is that language-external operations (thinking, decision making etc.) can access it. Results of the LF legibility tests are shown right after transfer (line 632). If an input sentence is judged ungrammatical, then in many cases this means that LF legibility has failed for all possible candidate parses, and the reasons for these failures will appear here. Backtracking is visible on lines 673-684. In this case, no other candidate solutions could be generated, and the expression was correctly judged unambiguous, having one meaning only.

Many linguists and cognitive scientists will consider these mechanisms too complex or based on erroneous or cognitively irrelevant background assumptions. This is a legitimate concern. We want to capture the data by using the simplest assumptions possible. Hence, all disagreement must be solved by comparing concrete formalized hypotheses which calculate the same or at the very least very similar datasets, in the best case much bigger datasets. My own best guess is that in order to capture the data (see Section 1.2) in an insightful manner we have to assume that the human language faculty maps input sentences into syntactic representations and then interprets them after some type of noise reduction or canonicalization that is captured by means of reconstruction, whose purpose is to transform compressed sensory objects into a cognitive template (e.g., (47)) understood by universal semantics, or other cognitive systems. If a simpler model is possible, it should certainly be adopted.

Transitive clauses (48)a are derived by the same mechanisms. In this case there is an additional postverbal argument that will get merged^{-1} directly to its correct position at the spellout structure (b) from the surface string.

(48) a. Pekka ihaile-e Merja-a. (#4)

Pekka.NOM admire-PRS.3SG Merja-PAR

'Pekka admires Merja.'

b. [[D Pekka] [T(v, V) D(N)]] (Spellout structure, line 759)

The subject is transferred before final transfer, but both the finite verb and the direct object remain in their untouched forms T(v, V) and D(N) since neither constitutes a left branch. They will be transferred during the final transfer operation. Transitive verbs contain three components per lexicon: the verb stem V, transitivization v and tense T. Head reconstruction is applied (b) and is reported in line 762, with the output [$_{TP}$ DP [$_{TP}$ T [$_{vP}$ v [$_{VP}$ V [$_{DP}$ D N]]]]], where each head is reconstructed locally and in top-down order (i.e. T(v), v(V), D(N)). Reconstruction is applied top-down by descending on the projectional spine of the structure. Left branches and right adjuncts are not visited (they were processed as independent phases earlier). The topic at SpecTP reconstructs to SpecvP. The direct object does not trigger reconstruction, because it satisfies the complement selection feature of the verb at the position where it is merged^{-1} during parsing and checks its direct object case (partitive) at the same position against v. It is interpreted as the patient of the event (line 785).

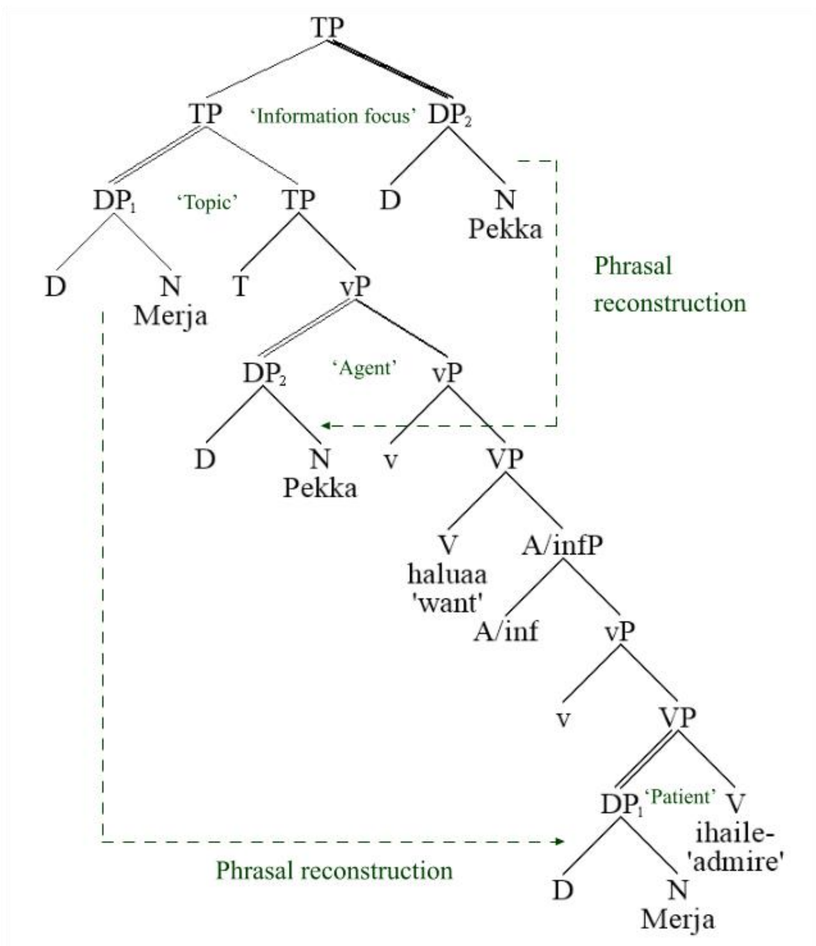
At this point we can simulate processing of all possible word orders. An OVS variation is number #30 in the test corpus. Here the object and subject appear in reverse order in the surface string. The topic at SpecTP at the spellout structure has the direct object case and is reconstructed nonlocally to the direct object position, while the postverbal argument with the nominative is located to SpecvP. These operations are visible on line 7453, where it is reported that the constituents (S, O) failed "tail tests" (which are effectively inverse agreement (probe-goal) tests for case checking purposes, hence inverse-agreement operations) and were reconstructed into positions where they could be satisfied. Recall that it is assumed that overt case forms can be used in this way for reconstruction purposes. All possible variations, including ungrammatical ones, can be found from the default corpus. Sentences #30-35 contain a variety of clause types with the noncanonical O...S order. Of particular interest (to me, in particular, see (Brattico 2016, 2018)) is sentence (49), #34 in the test corpus.

(49)	Merja-a	halua-a	ihail-la	Pekka.
	Merja-PAR	want-PRS.3SG	admire-A/INF	Pekka.NOM

‘Pekka (information focus) wants to admire Merja (topic).’

Data such as these show that OVS inversion is not limited to simple transitives; rather, the grammatical subject is free to move further to the right. One solution calculated by the model for this input sentence is (50), where the extreme rightward position of the grammatical subject is associated with information focus interpretation, while the preverbal direct object is interpreted as the topic of the clause.

(50)



Another parse, shown in the derivational log, is one in which the rightward grammatical subject is originally merged to a postverbal position. It is reconstructed from there to SpecvP.

Computation of ditransitive clauses introduce an additional issue. Consider (51).

(51) Pekka anto-i avaime-n Merja-lle. (#7)

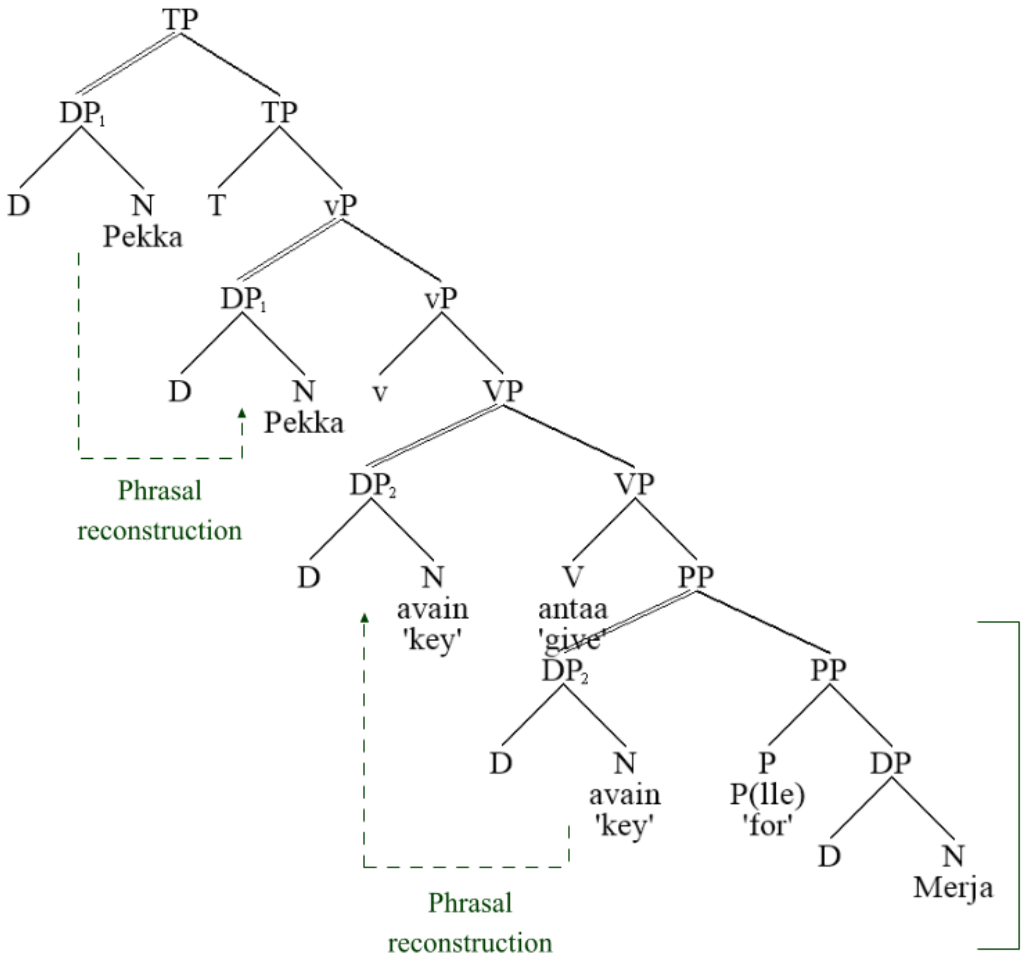
Pekka.NOM give-PST.3SG key-ACC Merja-ALL (allative case)

'Pekka gave the key to Merja.'

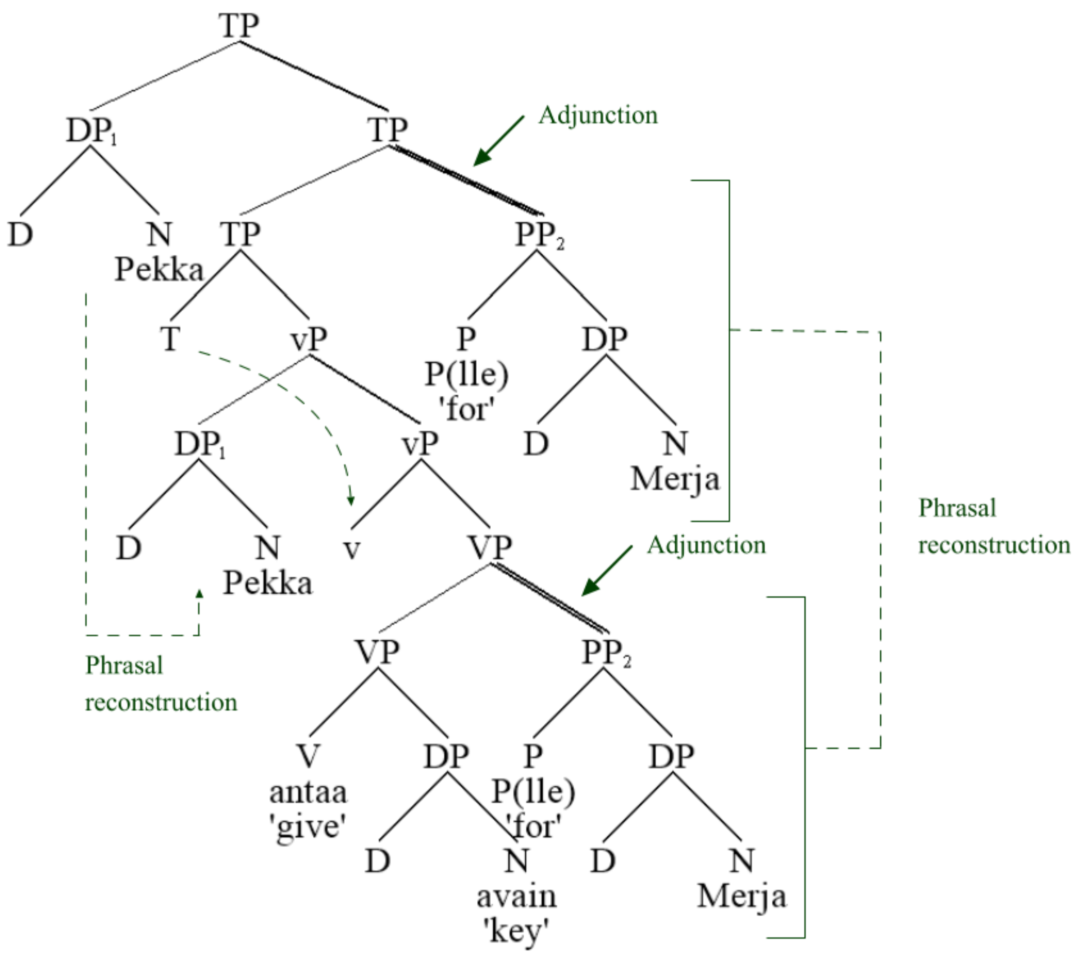
The indirect argument (abbreviated as IA) is marked by allative case *Merja-lle* 'Merja-ALL' corresponding roughly to English 'to'. Nikanne (1993) argued, convincingly, that Finnish semantic case forms should be analysed as covert PP structures, with the phonologically covert preposition assigning the corresponding case form. This analyses was assumed in the present study, thus the case-marked argument is decomposed in the lexicon as P#D#N, with the preposition corresponding to an allative 'to'. The spellout structure will then contain P(D, N) corresponding to the indirect argument, which is expanded into [P($\_{}_1$) [D($\_{}_2$)$_1$ N$_2$]] by two head reconstruction chains. The problem, though, is that straightforward merging^{-1} of the direct and indirect arguments into the structure produces gibberish. In principle the PP can be merged with the TP, vP and DP, each which produces an anomalous output when the resulting left branch is interpreted as being in the specifier position of the hosting PP (i.e. [$_{PP}$ TP PP], [$_{PP}$ vP PP], [$_{PP}$ DP PP]). None of these, if they were sent out to the LF interface, could survive. It is therefore assumed in the syntactic background theory that the PP is or can be adjoined to the right edge. Adjunction is marked by 〈〉 in the output and by double line in the phrase structure images. Adjunction is loosely modelled after Chomsky (2004) where it was proposed that adjuncts are merged differently and located in an another "syntactic plane." This was implemented so that they are geometrical daughter constituents of their hosts while being invisible for labelling inside the host object (Brattico 2020). Adjunction creates an "extra dimension" to the phrase structure. Because the parser will explore all possible candidate solutions, the end result will show a three-way ambiguous sentence, the three interpretations corresponding to the three adjunction sites. Two of the analyses are shown in

(52) and (53); the third is like (53) but with the PP right-adjoined to vP. The PP is produced by applying head reconstruction twice to P(D, N)(see, e.g., line 1344).

(52)



(53)



In the first, (52), the PP is initially merged/adjoined[-1] as the sister of the direct object and the direct object is then reconstructed from SpecPP to SpecVP. It can still check the direct object case at this position against v. In the second solution, (53), the PP is first merged to a high position in the structure and then reconstructed as shown above. Because the PP is adjoined, labelling ignores it and the model calculates a "two-segment" VP solution. Further PP constructions were tested by attaching a locative PP with a variety of constructions (Group 1.1.15, #38-46).

(54)

a.	Pekka	ihaile-e	Merja-a	Helsingissä.
	Pekka.NOM	admire-PRS.3SG	Merja-PAR	in.Helsinki
	'Pekka admires Merja in Helsinki.'			

b. Pekka e-i ihaile Merja-a Helsingissä.

	Pekka.NOM	not-3SG	admire	Merja-PAR	in.Helsinki		
c.	Pekka	o-n	ihail-lut	Merja-a	Helsingissä.		
	Pekka.NOM	be.PRS.3SG	admire.PST.PRTCPL	Merja-PAR	in.Helsinki		
d.	Pekka	e-i	ol-lut	ihail-lut	Merja-a	Helsingissä.	
	Pekka.NOM	not-3SG	be-PST.PRTCPL	admire-PST.PRTCPL	Merja-PAR	in.Helsinki	
e.	Pekka	halua-a	ihail-la	Merja-a	Helsingissä.		
	Pekka.NOM	want-PRS.3SG	admire-A/INF	Merja-Par	in.Helsinki		
f.	Peka-n	täytyy	ihail-la	Merja-a	Helsingissä.		
	Pekka-GEN	must	admire-A/INF	Merja-PAR	in.Helsinki		
g.	Pekka	e-i	halua	ihail-la	Merja-a	Helsingissä.	
	Pekka.NOM	not-3SG	want	admire-A/INF	Merja-PAR	in.Helsinki	
h.	Pekka	sanoo	että	halua-a	ihail-la	Merja-a	Helsingissä.
	Pekka	says	that	want-PRS.3SG	admire-A/INF	Merja-PAR	in.Helsinki

These generate the same multiple ambiguities as (51), including ambiguities with respect to the clausal scope (h), but in some cases (e.g., g) the narrow scope reading did not emerge (e.g. 'Pekka wants something in Helsinki' and not 'Pekka wants: x admires Merja in Helsinki'). In principle the idea is that the different scope readings should result from the different right attachment sites of the PP adjunct (lower attachment site reconstructs to lower clause, higher attachment site to higher clause; see example h for how this mechanism works), but some factors in the syntactic background mode conspired to render the lower attachment sites impossible. Adverbial scope computations have not been examined in detail with the syntactic background theory.

4.3.1.3 Negation and tensed auxiliaries

Both Finnish negated clauses and clauses with tensed auxiliaries involve input sentences where an additional word, the negation or the auxiliary, occurs between the preverbal subject phrase and the verb. These sentences were collected into Groups 1.1.3 and 1.1.4. An additional test sentence (55) contained both the negation and an auxiliary.

(55) Pekka e-i ol-lut ihail-lut Merja-a. (#11)

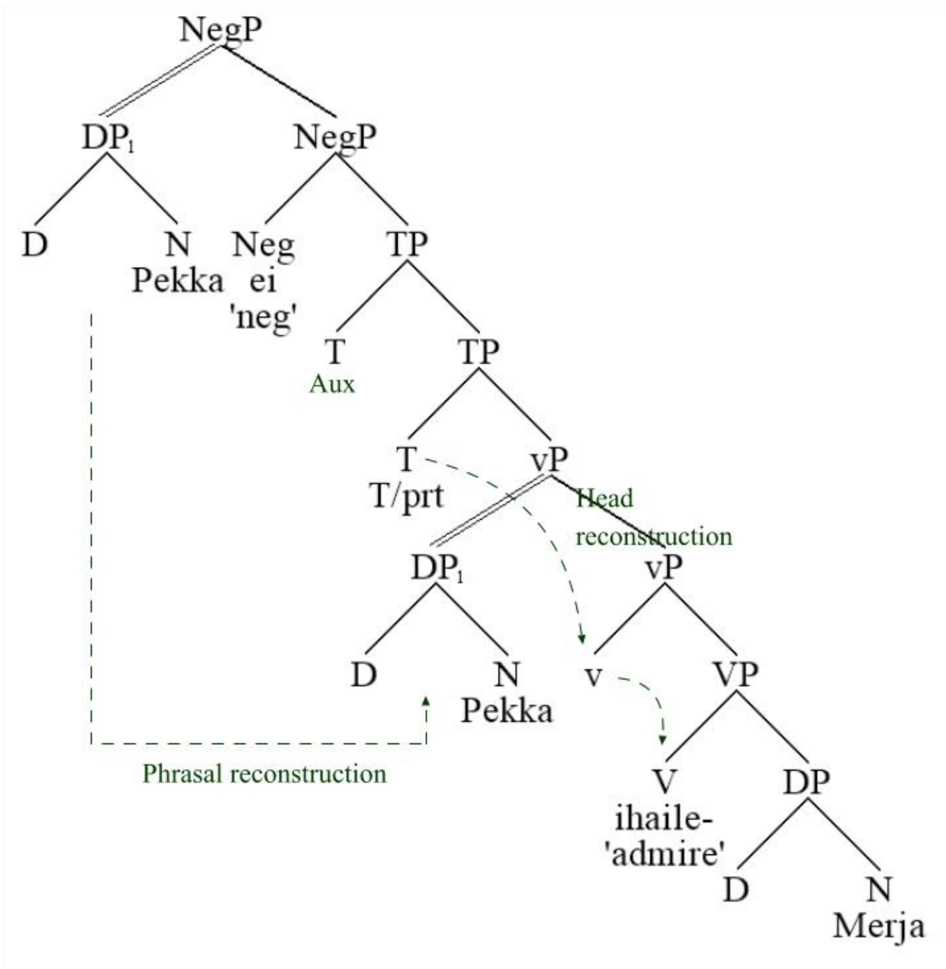
Pekka.NOM not-3SG be-PST.PRTCPL admire-PST.PRTCPL Merja-PAR.

'Pekka had not been admiring Merja.'

(55)	Pekka	e-i	ol-lut	ihail-lut	Merja-a. (#11)
	Pekka.NOM	not-3SG	be-PST.PRTCPL	admire-PST.PRTCPL	Merja-PAR.
	'Pekka had not been admiring Merja.'				

The calculated analysis is (56)(line 2329).

(56)



The two extra words *e-i* 'not-3SG' and *ol-lut* 'be-PST.PRTCPL' in the input generate two extra heads into the structure, while the complex verb *ihail-lut* 'admire-PST.PRTCPL' is reconstructed into T/prt-v-V spread where T/prt represents the past participle form. The preverbal subject (topic) phrase is reconstructed directly into SpecvP and does not implement local successive cyclic reconstruction that would take place in English where arguments are not case-marked and cannot be floated.[54] Notice that in Finnish the negation constitutes a tenseless auxiliary-like element that is merged above TP, agrees with the grammatical subject and projects NegP. Neither the negation nor the tensed auxiliary project thematic roles, hence they do not project thematic positions that can host arguments at LF interface. This difference is represented by lexical features: the lexical verb contains a lexical feature ASP whereas the auxiliaries and the negation do not (lines 2351-2362). This feature is associated with an independent event structure and thematic roles.

4.3.1.4 Sentences with the A-infinitival complement clause (Groups 1.1.5-1.1.8, 1.1.10)

Groups 1.1.5-1.1.8 and 1.1.10 contain a variety of construction with an A-infinitival complement clause (corresponding loosely with the English to-infinitivals). Recall that many head movement constructions target these A-infinitivals. Four examples from groups 1.1.5-1.1.8 are illustrated in (57)a-d.

(57)

a.	Pekka	halua-a	ihail-la	Merja-a. (#13)	
	Pekka.NOM	want-PRS.3SG	admire-A/INF	Merja-PAR	
	‘Pekka wants to admire Merja.’				
b.	Peka-n	täytyy	ihail-la	Merja-a. (#15)	
	Pekka-GEN	must.0	admire-A/INF	Merja-PAR	
	’Pekka must admire Merja.’				
c.	Peka-n	e-i	täydy	ihail-la	Merja-a. (#17)
	Pekka-GEN	not-3SG	must	admire-A/INF	Merja-PAR
	‘Pekka must not admire Merja.’				
d.	Pekka	e-i	halu-a	ihail-la	Merja-a. (#19)
	Pekka.NOM	not-3SG	want-PRS.3SG	admire-A/INF	Merja-PAR
	‘Pekka does not want to admire Merja.’				

---

[54] The matter is more complex because some arguments (pronouns) are case-marked in English but do not float freely, and the same is true of Finnish genitive-marked DPs.

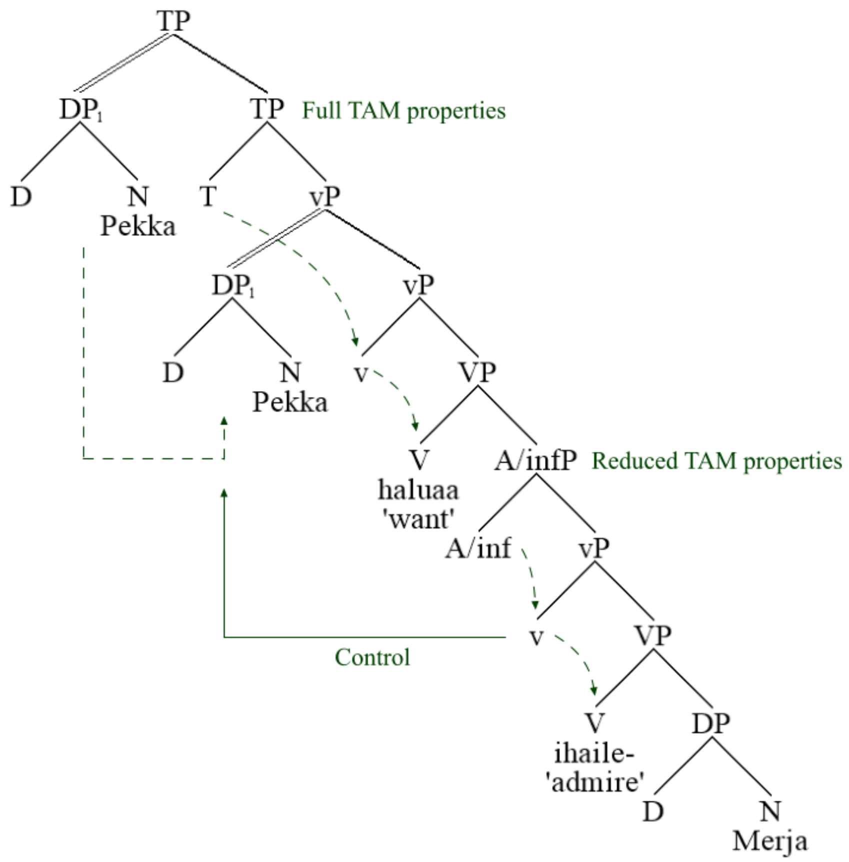
We consider the derivation of (57)a (#13 in the test corpus) to illustrate how the model analyses infinitival complement clauses. The main verb is analysed as before, thus T(v, V) → [$_{TP}$ T [$_{vP}$ v V]]. The infinitival form has three elements: A-infinitival suffix, transitive v, and the verb stem V, which are reconstructed into [$_{A/infP}$ A/inf [$_{vP}$ v V]] with A/inf standing for the A-infinitival head. The input sentence (58)a generates the spellout structure (b), which the head reconstruction rule then maps to (c).

(58)

a.	Pekka	haluaa	ihail-la	Merja-a	(Input)
	'Pekka	wants	admire-A/INF	Merja-PAR	
b.	[Pekka	[T(v, V)	[A/inf(v, V)	D(N)]]]	(Spellout structure, line 2770)
c.	[Pekka	[T [v [V	[A/inf [v [V	[D N]]]]]]	(Output of head reconstruction, line 2774)

Phrasal reconstruction reconstructs the preverbal topic into SpecvP, which generates the final output (59).

(59)

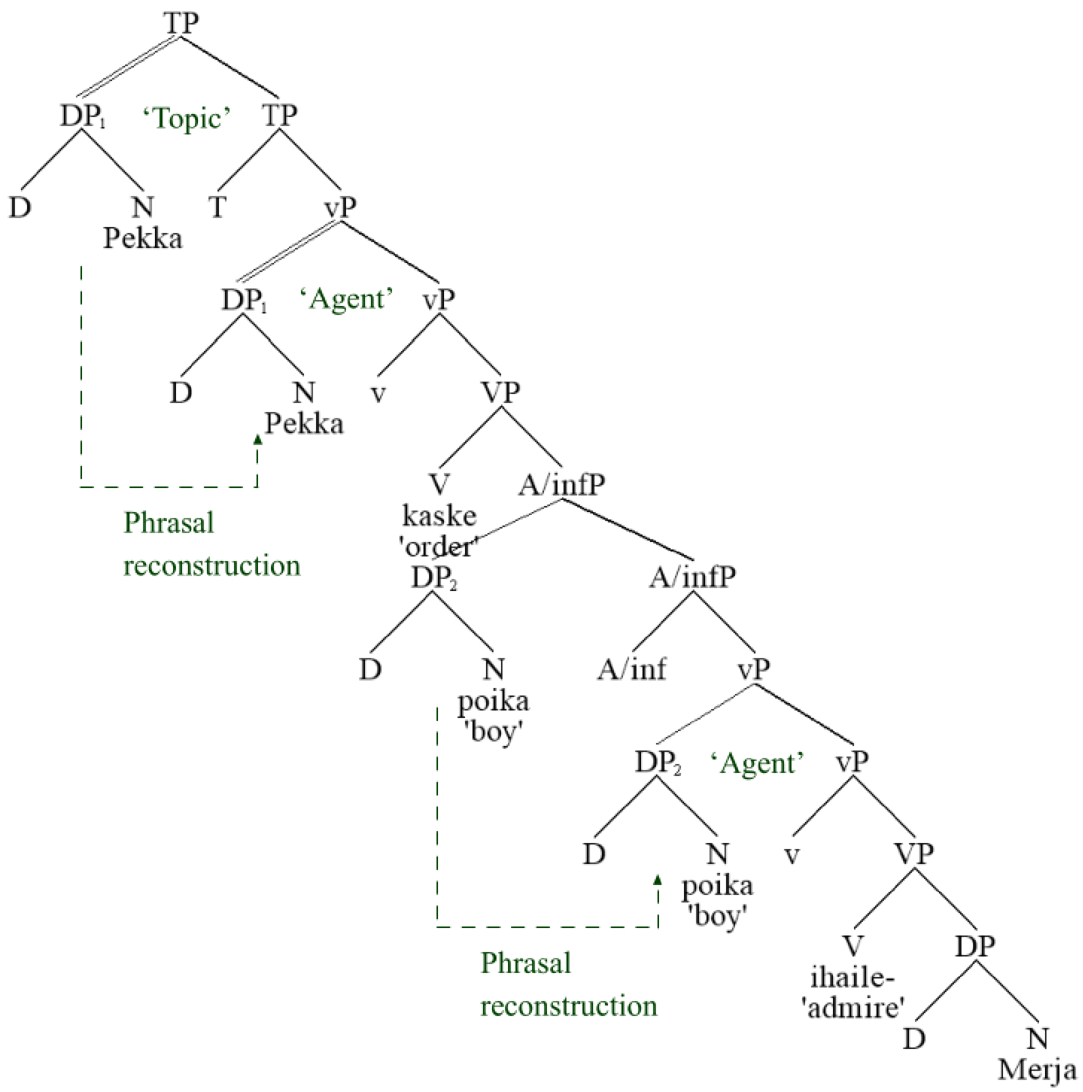


I have assumed that infinitival clauses, and in particular the infinitival head, represent reduced TAM (tense, aspect, mood) properties. The main verb V therefore selects the A-infinitival clause A/infP as its complement, hence the interpretation is ‘Pekka wants: x admires Merja’, where the agent argument of the infinitival verb is linked with Pekka by control (line 2796). No null argument PRO is hallucinated into the structure; the control dependency is initiated by an unsaturated argument feature inside the infinitival predicate itself (Borer 1986, 1989; Brattico 2021b). In some of the calculated outputs the argument of the constituent is marked as if it were another head (e.g., line 4143, where it is reported that the “Agent of T” is the head *ei* ‘not’), but in this case the control dependency was established between the predicate and the phi-set inside the head and not the head itself. If the infinitival contains its own thematic subject, it will be generated to SpecA/inf on the basis of surface order and reconstructed into SpecvP (60)-(61).

(60)	Pekka	käske-e	[$_{\text{A/infP}}$	poiki-en	ihail-la	Merja-a.] (# 23)
	Pekka.NOM	order-PRS.3SG		boys-GEN	admire-A/INF	Merja-PAR

‘Pekka orders the boys to admire Merja.’

(61)



The A-infinitival head representing reduced TAM-properties behaves like the finite T: both are nonthematic heads, triggering reconstruction.

4.3.1.5 Full finite complement clauses (Group 1.1.9)

Full finite complement clauses are headed by a high complementizer *että* ‘that’ in Finnish which projects an extra force projection above TP (62).

(62) Pekka sanoi [$_{\text{ForceP}}$ että [$_{\text{TP}}$ halua-a ihail-la Merja-a.]] (#21)

Pekka.NOM said that want-PRS.3SG admire-A/INF Merja-PAR

‘Pekka said that (he) wants to admire Merja’

See lines 4211-4936 in the derivational log file.

4.3.1.6 Adverbials

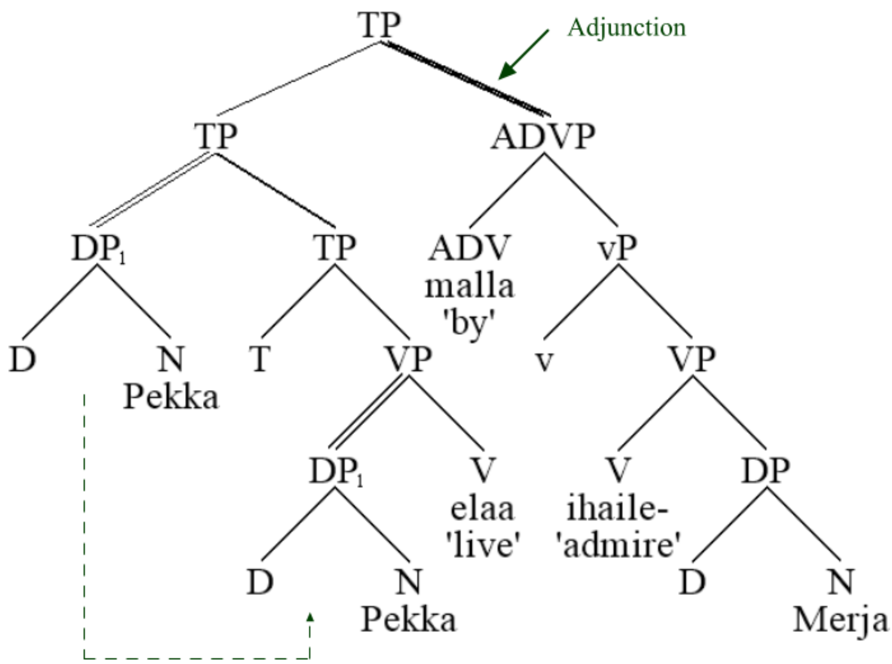
A typical adverbial construction used in the study is illustrated by (63), where the adverbial phrase is right-adjoined to the clause as shown in (64). The sentence is pragmatically odd with these lexical choices.

(63) Pekka elä-ä [$_{\mathrm{AdvP}}$ ihaile-malla Merja-a.] (#25)

Pekka.NOM live-PRS.3SG admire-MA/INF Merja-PAR

'Pekka lives by admiring Merja.'

(64)



The canonical position of an adverbial at the LF interface is determined by its *tail feature* which determines the type of projection it must occur under. The MA-infinitival is marked for a tail-feature T (see line 5538) and must therefore reside inside a TP. This will determine its control properties and hence which argument in the sentence will be interpreted as the agent of the verb inside the adverbial (Brattico 2021b).[55]

4.3.1.7 A-infinitivals as noun complements

A-infinitivals can be used as noun complements. They were analysed, unproblematically, by the model as shown here.

(65)

a.	Pekka	tek-i	[sopimuksen	[$_{A/infP}$	ihail-la	Merja-a.]] (#27)
	Pekka.NOM	make-PST.3SG	agreement		admire-A/INF	Merja-PAR

‘Pekka made an agreement to admire Merja’

b.	[$_{DP}$Se	sopimus	[$_{A/infP}$	ihail-la	Merja-a]]	peruuntui. (#29)
	that	agreement		admire-A/INF	Merja-PAR	cancelled

‘The/that agreement to admire Merja was nullified.’

4.3.1.8 C-features (Group 1.1.16)*

Sentences #47-70 contain examples of various C*-features in connection with phrasal (subject/object) operator movement, #71-82 in connection with noncanonical OSV orders. 12 C*-features and their combinations were tested, but this does not exhaust the repertoire that contains over twenty combinations still excluding some unclear cases. Symbol C/op was used as a general operator feature in the tests and can be used to substitute any of the more specific features; this morpheme does not exist in Finnish.

(66)

a.	Pekka#c/op	ihaile-e	Merja-a.
	Pekka-C/OP	admire-PRS.3SG	Merja-PAR

---

[55] It is possible to employ this mechanism to distinguish TP-adverbials (TAIL:T) from low VP-adverbials such as locatives (TAIL:V). PPs examined earlier were assumed to be “VP-adverbials,” thus arguments of the verb and/or the event denoted by the verb. If the adverbial occurs in a position where it should not be on the basis of its tail feature, reconstruction is attempted. Since adverbials and PPs are attached to the clause as right adjuncts, their number is unlimited, making it possible to interpret sentences such as *John saw Mary* [*in Helsinki*] [*yesterday*] [*by using a telescope*].

’Pekka(operator) admires Merja.’

b. PEKKA ihaile-e Merja-a.

Pekka.FOC admire-PRS.3SG Merja.PAR

’It is Pekka, and not someone else, who admires Merja.’

c. Pekka-han ihaile-e Merja-a.

Pekka-FAM admire-PRS.3SG Merja-PAR

‘Pekka (familiarity topic) admires Merja.’

d. Pekka-ko ihaile-e Merja-a?

Pekka-Q admire-PRS.3SG Merja-PAR

‘Is it Pekka who admires Merja?’

e. Pekka-pa ihaile-e Merja-a!

Pekka-POL admire-PRS.3SG Merja-PAR

‘Pekka (polarity topic) admires Merja.’

f. PEKKA-HAN ihaile-e Merja-a!

Pekka.FOC-FAM admire-PRS.3SG Merja-PAR

‘It is Pekka (familiarity topic), not someone else, who admires Merja.’

g. PEKKA-KO ihaile-e Merja-a?!

Pekka.FOC-Q admire-PRS.3SG Merja-PAR

‘Was it PEKKA, and not someone else, who admires merja?!’

h. PEKKA-PA ihaile-e Merja-a!!

Pekka.FOC-POL admire-PRS.3SG Merja-PAR

‘It is PEKKA who admires Merja!’

j. Pekka-ko-han ihaile-e Merja-a?

Pekka-Q-FAM admire-PRS.3SG Merja-PAR

‘I wonder if it is Pekka who admires Merja?’

k. Pekka-pa-han ihaile-e Merja-a!

Pekka-POL-FAM admire-PRS.3SG Merja-PAR

‘It is Pekka who admires Merja.’

l.	PEKKA-KO-HAN	ihaile-e	Merja-a?!
	Pekka.FOC.Q-FAM	admire-PRS.3SG	Merja-PAR

‘Is it really PEKKA, and not someone else, who admires Merja?!’

m.	PEKKA-PA-HAN	ihaile-e	Merja-a!!
	Pekka.FOC-POL-FAM	admire-PRS.3SG	Merja-PAR

‘PEKKA admires Merja!!’

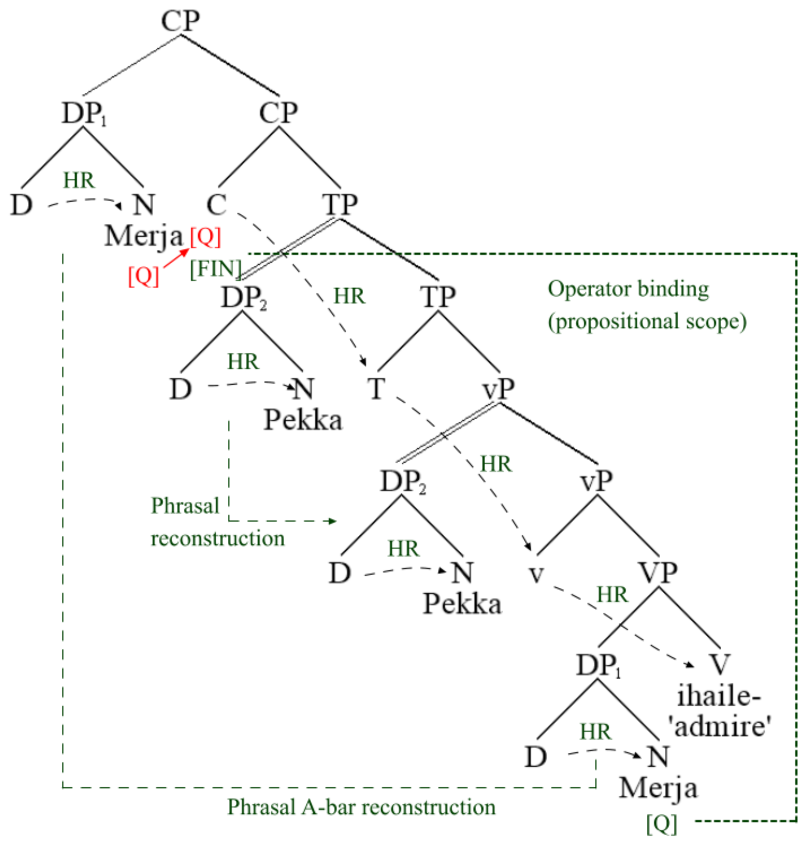
To illustrate how these are calculated, I will use an object operator sentence (67) as an example.

(67)	Merja-a-ko$_1$	Pekka	ihaile-e	_$_1$? (#74)
	Merja-PAR-Q	Pekka.NOM	admire-PRS-3SG	

’Is it Merja who Pekka admires?’

The spellout structure computed for this sentence is [$_{TP}$ *Merja* [$_{TP}$ *Pekka admires*]], with the two arguments heading the clause (line 20216) with the first containing the operator feature corresponding to yes/no question Q. In this situation the model infers that the clause must be headed by a phonologically covert C-head that contains the criterial feature Q and then reconstructs the operator during transfer by using an Ā-dependency. The result is below.

(68)



See lines 20216-20233 in the log file. The results contain important information concerning the interpretation of the operator. Speaker attitude is "interrogative" (line 20247), which is inferred from an interrogative (Q-)feature at C. This feature is read off from the criterial feature inside the pied-piped phrase. Two operator bindings are listed (line 20246) which are also shown in (59). This dependency determines the propositional scope for the operator. Two operator features trigger these dependencies: the original yes/no feature Q and an interrogative feature WH. This is because the surface morpheme -*kO* is linked with these two features in the lexicon. Scope is determined by the fact that the operator feature OP matches with {OP, FIN} higher up in the clause. Overt operator movement "communicates" scope information through the sensorimotoric interfaces. These interpretations are created by narrow semantics that bleeds LF interface. Head reconstruction (head movement) uses the same mechanism, but with the exception that the operator feature triggering these computations is on a verbal head (e.g., T, v, V, A-infinitival). The algorithm handles also pied-piping: the operator feature does not need to appear on a head that is fronted, it can be inside a moved phrase as well.

The existence of filler-gap dependencies is puzzling. The analysis presented here implies that these constructions rely on some more general mechanism which allows the content of a head H to be expressed either by phonological properties at H or by carrying these properties by material (pied-piped phrase) at its local specifier.

In the test corpus, many of the sentences above are provided in a morphologically decomposed form in which the C*-morpheme is separated from the stem by a morpheme boundary marker #. Thus, the sentence *Pekka-ko ihailee Merjaa* 'Pekka-Q admires Merja' is provided in the form *Pekka#ko ihailee Merjaa*. This is done for convenience only. It is possible to provide a separate lexical entry for *Pekkako* 'Pekka.Q' and then decompose it in the lexicon; we do not want to do this, however, if the main purpose of the study is to examine the behavior of the productive C*-morphemes. By providing the decomposition in the input we are explicit on what we are trying to test. If the decomposition is hidden inside the lexicon, one has to check that the decomposition is correct. Finally, the format *a#b* is based on the notation used internally by the algorithm. The lexicon maps input *ab* into *a#b* (assuming decomposition), therefore if we provided *a#b* directly in the input we would just skip the extra lexical step but proceed otherwise in the exact same way. This is illustrated in Figure 25.

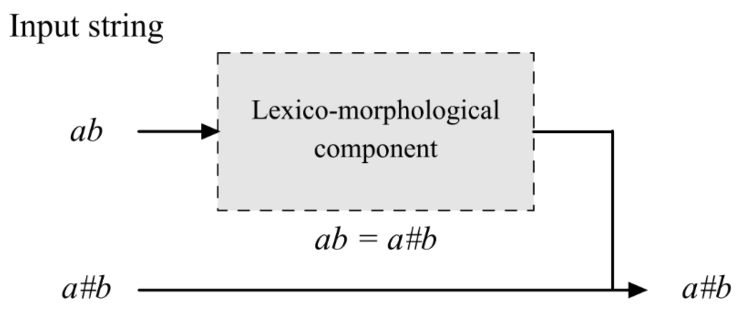


Figure 25. Two equivalent ways to represent an input word: one with lexical decomposition performed by the lexico-morphological component (upper) and another with the lexical decomposition provided directly in the input.

4.3.2 Local T-to-C movement (Group 2)

Group 2 contains examples of local T-to-C head movement in connection with virtually all construction types examined in Section 4.3.1 (Group 1). In every sentence there is a finite element (finite verb, negation, modal, some type of auxiliary) that is moved to the clause-initial position and marked by a C*-feature, often so that all features and feature combinations included into the present study are tested. Several ungrammatical variations are also tested.[56] Let us begin by examining a T-to-C variation of a basic transitive clause (69), where the finite verb has been focussed contrastively. The mechanism is almost the same irrespective of the type of the C*-feature.

(69) IHAILEE$_1$ Pekka $\_\_{}_1$ Merja-a! (#102)
Admire.FOC Pekka.NOM Merja-PAR
'Pekka DOES admire Merja' or
'Pekka admires (not hates) Merja.'

The spellout structure is [C(T, v, V) [[D *Pekka*] D(N)]] (line 25380). C(T, v, V) remains a complex head, while the grammatical subject argument has been expanded. This is accomplished by "parsing intelligence" discussed in Section 2, which infers that the verb should be expanded (later) on the projection spine of the sentence while D(N) will become a left branch. Head reconstruction algorithm then produces [$C_{foc,fin}$ [[D *Pekka*] [$T_{foc,fin}$ [v [V [D *Merja*]]]]]] (lines 25383-4). Two properties of this transform should be mentioned, although they have been discussed already in the main article and in this document. The first important thing to notice is that the T(v, V) cluster is extracted into a position lower than the grammatical subject. I was not able to calculate the dataset correctly without writing this condition into the model. See Section 3.6 in the main article. The exception says in essence that in this particular case the algorithm must try to satisfy also the head's specifier selection (EPP). Leaving the

[56] Notice that the left peripheral C*-features all have a slightly different properties and carry different features. See Section 3.6.

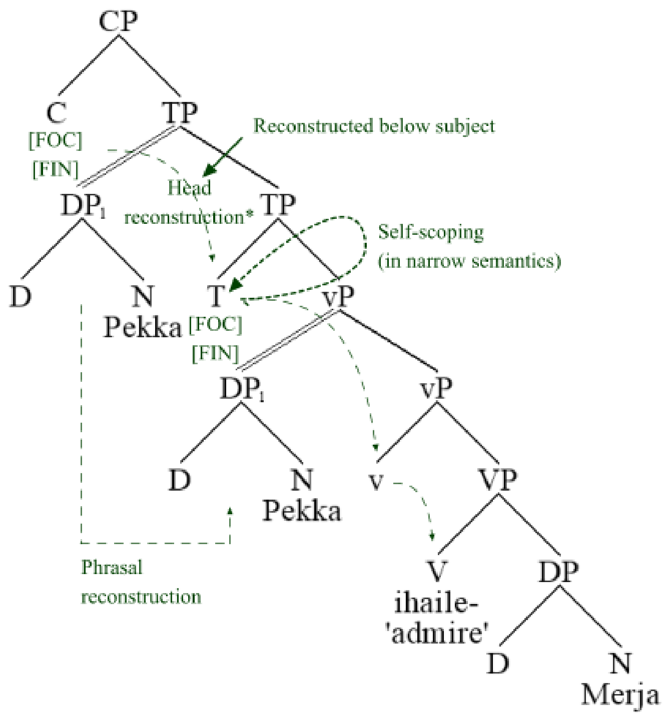
exception out will leave the subject into wrong position,[57] but generalizing the mechanism (requiring it everywhere) will also produce wrong results. It is also intuitively clear that native speakers understand that the grammatical subject occurs in the SpecTP position, is the topic and subject, and is not in any sense postverbal; there is nothing exceptional in its position, rather it is the finite verb that occurs in a noncanonical position.

Second, notice that the assumptions made so far entail that the same element that hosts the operator feature (finite T) will also constitute a potential scope marker. The operator module will attempt to link operators with finite heads (interpreted as propositions by the semantic system) that have the same operator feature. Thus, in the derivational log it is recorded that T was bound by same T (line 25410). In essence this means that the operator (finite verb) determines its own propositional scope (the same finite clause). We could easily rule this configuration out by preventing self-scoping, but it is interesting to note that this special situation is also the one associated with the special verum focus interpretation and the standard HMC locality effects; see the translations in (69). It was therefore left into the model. Perhaps "self-scoping" is the configuration that is be interpreted by the second system that does not involve targeting the lexical content of the moved predicate? This was also the primary reason I did not write any code for handling the second system, leaving this possibility open for further development.

After all transfer operations have been done, (69) yields (70).

[57] The reconstructed structure would be C [T[v [V [S O]]]].

(70)



In the main article and analysis it was assumed that the criterial feature (here FOC) is "reconstructed in situ." This is how and why the C-head appears in the structure. The implementation, however, was left unclear. There are several possible mechanisms. The current model uses the mechanism introduced earlier: the criterial feature is treated as a lexical feature that will be inside the C-head.[58] If there are several C*-features, then they are all inserted inside C as lexical features (see example #112, line 27731), and each feature will be interpreted separately by the semantic system (see lines 27721, 27767 for the interpretation of multiple C*features). C*-features have slightly different properties (not focussed on this study however) and must all be registered somewhere. Notice that there is no limitation that any one morpheme at the surface level must be linked with just one lexical feature in the lexicon; thus, it is possible that one morpheme is linked with several features. The Finnish *-kO*, for example, is currently associated with Q ('yes/no morpheme') and WH (interrogative), the latter which enters into

[58] This by no means necessarily true. It could be that FOC is the head morpheme itself; another possibility is that it is ambiguous and is mapped to an inflectional feature and head, both which are inserted into the search tree in some order determined by the most plausible, guessed continuation.

clause typing and selection (see examples #115, 117). Most sentences in the group produce very similar results to (70), with minor variations.

The discourse inventory, which contains a list of semantic objects projected into existence during processing, holds an object marked as “unsaturated tensed proposition” (line 25447). This is caused by the operator module when it links the operator with the proposition that constitutes its scope. This is so that the language-external systems can make inferences concerning sentences of this type. These mechanisms do not affect syntactic processing or the analysis proposed in the main article.

It is possible to attach feature C/fin to the fronted head (see sentences #164-173). This feature stands for the local or “formal” C-features whose nature was left open and in fact remains unclear to me. The group is heterogeneous.[59] In such cases, head reconstruction uses the local solution and the operator module will not create operator interpretations (see, e.g. the outcome for #164, line 40108). It is assumed, of course, that some other system processes this information. This is the grounds on which in the main article it was hypothesized that this system does “not understand” Ā-dependencies, namely it will not process these inputs by associating operators with propositional scopes and cannot therefore “understand” reconstructions that are nonlocal. Stated in other words, access to Ā-dependencies is granted exclusively to the operator module, in agreement with (my interpretation of) Chomsky’s duality of semantics hypothesis. This mechanism is used to derive English T-to-C constructions; see #235-236. There is one apparent problem: verb-initial sentences interpreted as all-new canonical clauses are ungrammatical in Finnish (see sentences #228-234) but they are judged grammatical. This is intentional. The idea is that whatever system will be ultimately responsible for interpreting these sentences will also create the required interpretations (verum focus interpretations in these cases) to the outputs. So although these sentences are now interpreted as if they were all-new canonical declaratives, this is only because the required semantic mechanism remains to be studied and implemented.

Reconstructing a fronted head when the argument order is noncanonical requires a comment. We consider sentence (71), #177 in the test corpus.

---

[59] In English, fronting the auxiliary locally creates an interrogative, where in Finnish it does not; in some languages the operation looks purely formal.

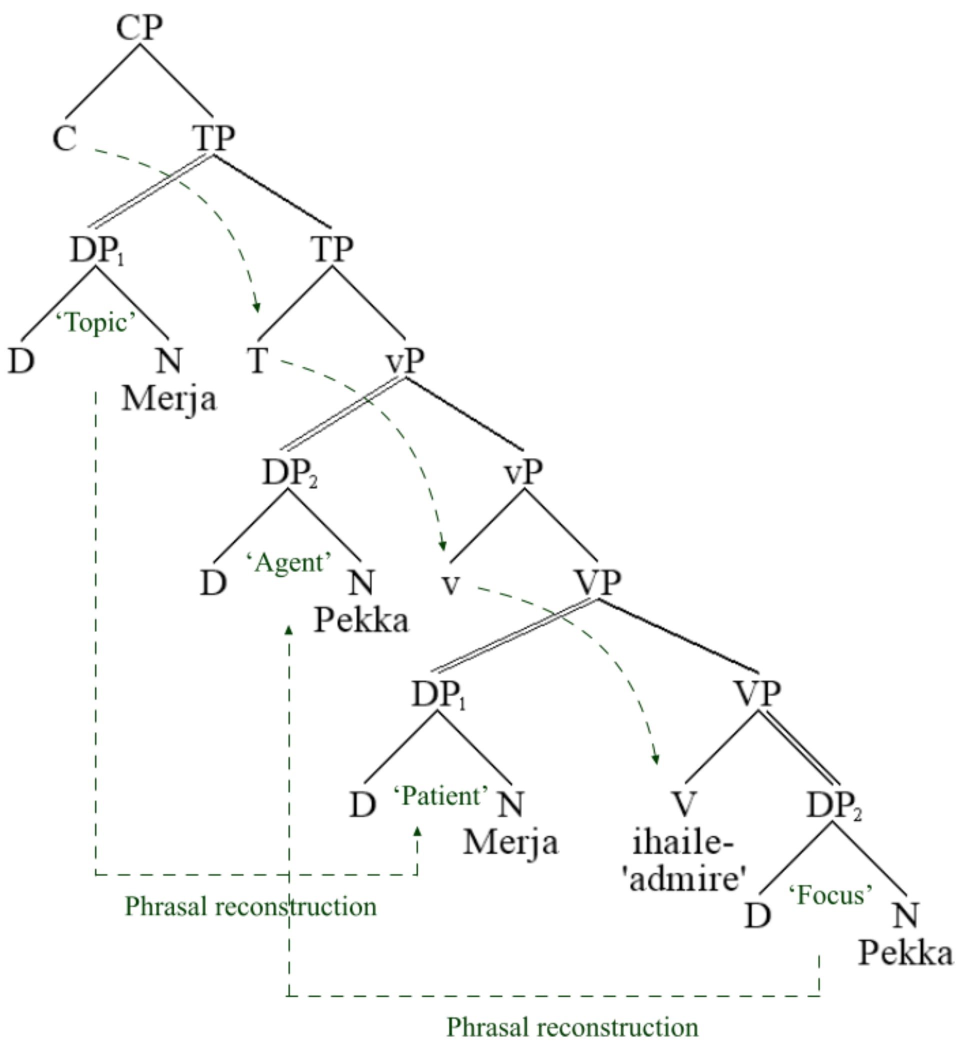
(71) Ihaile-e-ko Merja-a Pekka? (#177)

admire-PRS.3SG-Q Merja-PAR Pekka.NOM

’Does Pekka admire Merja?’

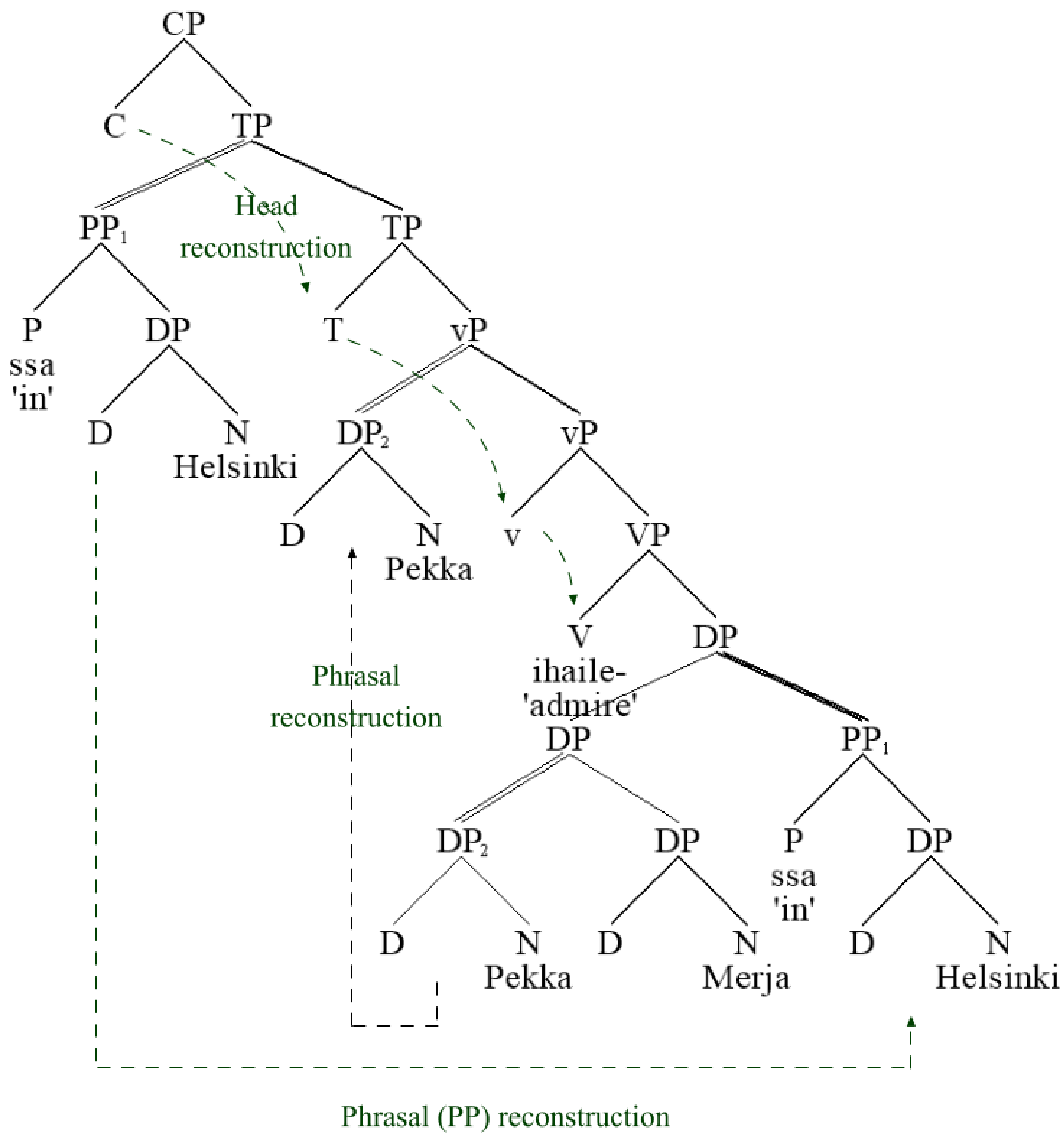
The hypothesis calculates (72) as output.

(72)



The output is correct and as expected, but nontrivial. Head reconstruction extracts the T(v, V) complex below the direct object that is at the topic position. The same mechanism must also work if the fronted phrase is a PP (sentences #193-209). In addition, head reconstruction must work correctly even when argument reconstruction (inverse topicalization and focussing) remains undone. It is not trivial that it does. The two systems, head reconstruction and phrasal reconstruction, must interact in the exact correct way in order to generate correct outputs for all input sentences. For example, consider sentence #196 with a PP preverbal topic (73).

(73)



While this produces correct judgment and the semantic interpretation (PP = topic) and is not linguistically implausible, it is far from trivial that the mechanisms work together in the manner required by the data. It is also not in any way obvious that the above is the correct analysis; very nontrivial issues are involved.

Checking of all output in this group did not reveal any errors except one typo in one lexical element (in sentence #230[60]) and one interesting spurious parsing solution that the model found in addition to the correct solution. The sentence involves a fronted copula, with the spurious solution indicated by bracketing.

(74) [ON	Pekka]	[Helsingissä]	[ihaillut	Merja-a.] (#214)	
be.FOC-PRS-3SG	Pekka.NOM	in.Helsinki	admire-PST.PRTCPL	Merja-PAR	

Correct interpretation: 'Pekka HAS admired Merja in Helsinki.'

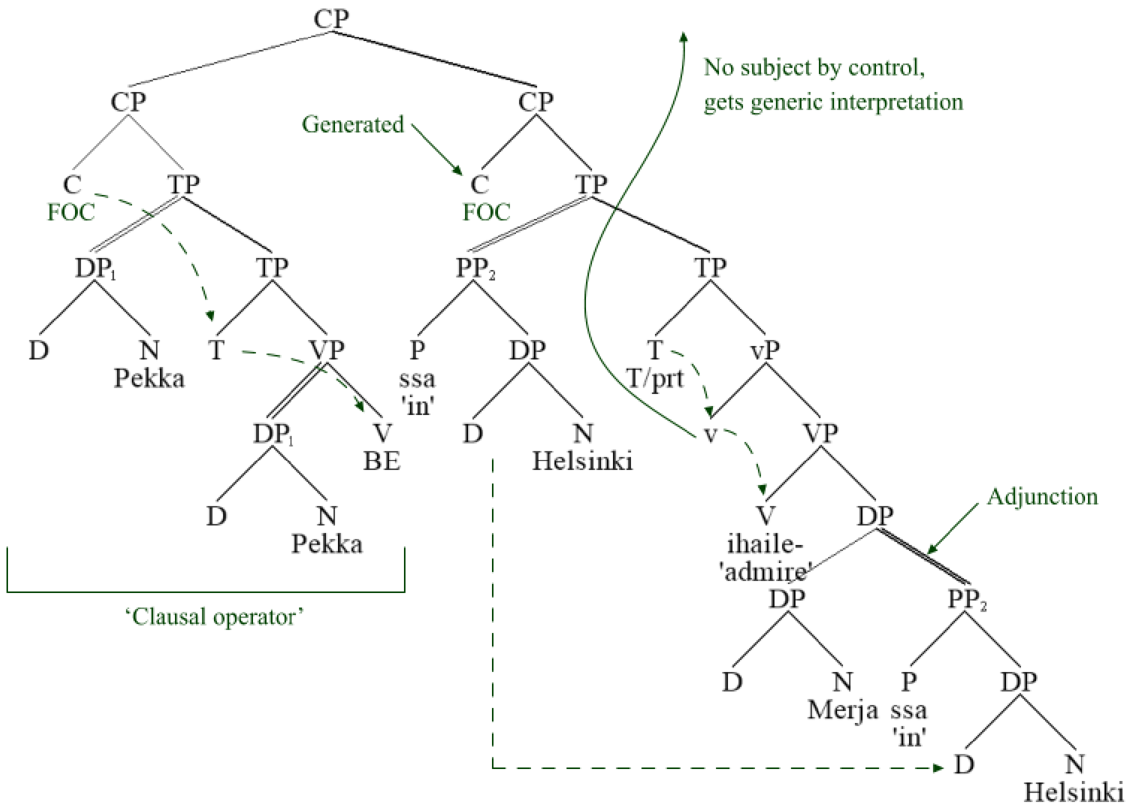
Wrong interpretation: 'One admires Merja in Helsinki.'

The model interpreted 'Pekka is' as a pied-piped clausal argument with the operator feature FOC inside. It was left in the surface position, as it has no legitimate position that could be reached by Ā-reconstruction. Since the agent argument was missing, the sentence was interpreted as generic ('one admires...'), analogously to sentences like *to leave now would be impolite*. The "operator" at the finite copula (T) could in theory be linked to its own clausal C, but the algorithm did not try this since semantic interpretation was deactivated for other solutions except the first one.[61] "Propositional operators," if they are possible in some languages, are predicted to have a self-neutralizing property which prevents them from being linked with the hosting proposition (i.e. the clause that embeds them). Pied-piping and moving a whole proposition to the operator field should be "semantically inert." The wrong analysis, returned in addition to the correct one, is shown in (75).

[60] Four cases of this kind were eventually discovered from the final datasets (#230, 345, 325, 472). The typos were corrected and the four sentences were independently derived. They are now included in the default corpus.

[61] I haven't considered in any detail how "semantic backtracking" should be implemented, if it indeed even exists as a cognitive operation (which I doubt). Psycholinguistically realistic recovery from a garden path could by modelled by assuming that the hearer parses the sentence anew with additional "noise" in the process to push it towards unexplored solutions. This would be easy to implement by adding a noise parameter to the parsing heuristics. Backtracking, used in the currently model, is useful when we want to verify that the model incorporates a correct notion of competence.

(75)



Notice that since the correct analysis was also calculated, problems like this are sometimes hard to detect as they are masked by the correct first pass parse solution. The grammaticality judgment was correct, and the spurious solution was only the second solution, therefore masked by the first. It is also possible that it was accepted only because semantic backtracking hasn’t been implemented and tested properly.

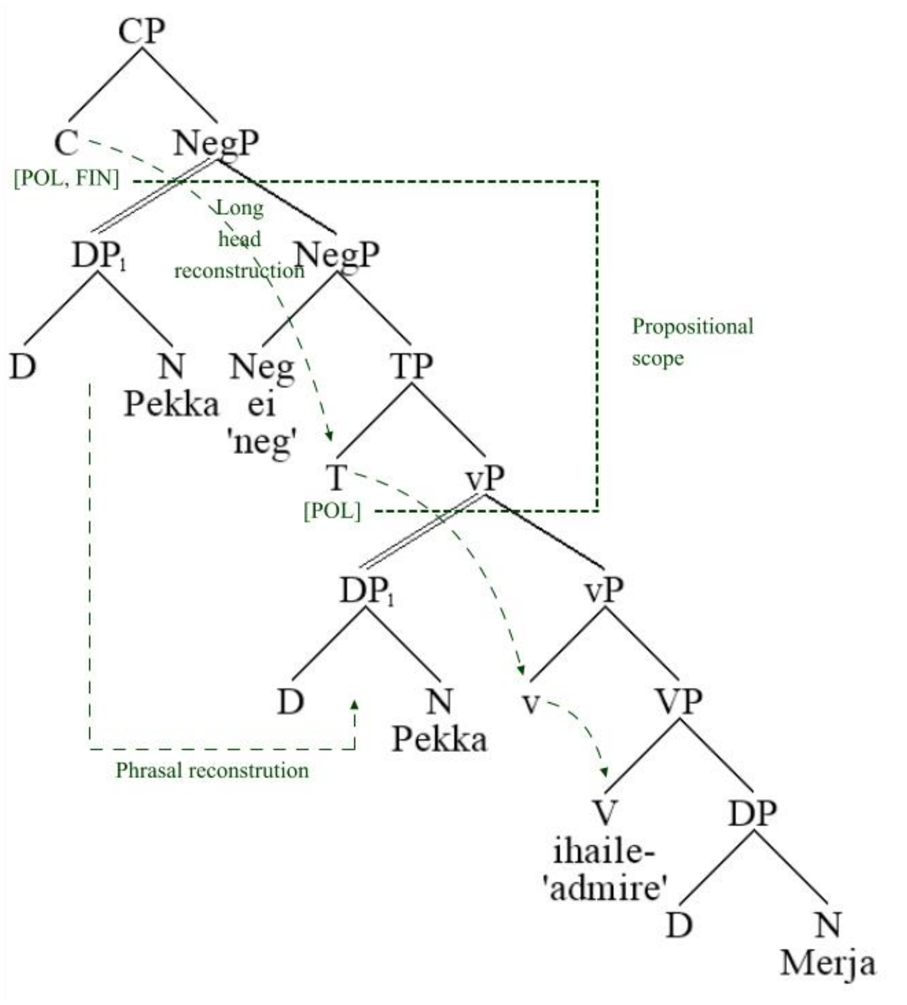
4.3.3 Long head movement (Groups 3, 4, 5)

Long head movement is a generalization of the mechanisms illustrated in previous chapters. If the reconstructed head does not fit into the first position available, search continues until there is no more structure. Two factors limit this operation. First, unbounded minimal search is only possible in the syntactic component if it is triggered by a C*-feature. Second, it is assumed that only the operator module knows how to interpret Ā-dependencies, and only operator (C*-)features are diverted into this module, as explained in the main article. Local head movement triggered by C*-feature(s), as discussed in the previous section, is a special case of the more general mechanism. A basic example of long head movement is shown in (76), with the analysis returned by the model provided in (68).

(76) Ihaile-pa	Pekka	e-i	_	Merja-a! (#241)
admire-POL	Pekka.NOM	not-3SG		Merja-PAR

‘Pekka does not admire Merja!’

(77)



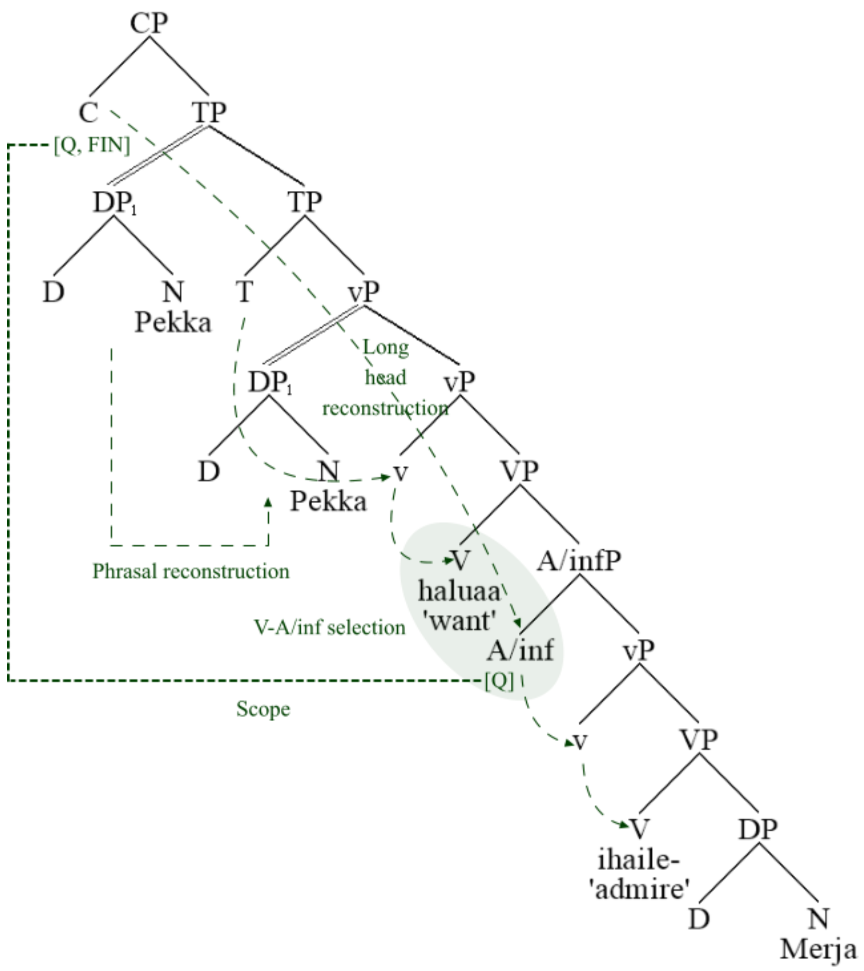
These examples illustrate the extent of “sensorimotoric compression” implied by the analysis: the spellout structure for the above clause is [C(...) [DP DP]], from which (77) is originally expanded.

Some of the sentences in this group are pragmatically odd, perhaps uninterpretable. This is because some auxiliary elements, such as the negation, do not combine well with certain C*-features. These effects are not modelled. There is no formal component that interprets the lexical content of C*-features (only the operator binding was implemented for this study), and their lexicon content is unclear and little studied. For this reason, the model can process any combinations of these features, although not all of them are possible. Notice that the copula is marked by symbol *on´* in the test corpus. This was because the lexical features of the copula have been difficult to formulate, and I have for this reason used several versions of the same element. Otherwise it is just a regular lexical element. The sentences mostly used in the main article are in group 3.1.3 (#263-275), with (78)-(79) providing an example.

(78) Ihail-la-ko	Pekka	halua-a	_	Merja-a!
admire-A/INF-Q	Pekka.NOM	want-PRS.3SG		Merja-PAR

‘Does Pekka want to admire rather than hate Merja?’

(79)



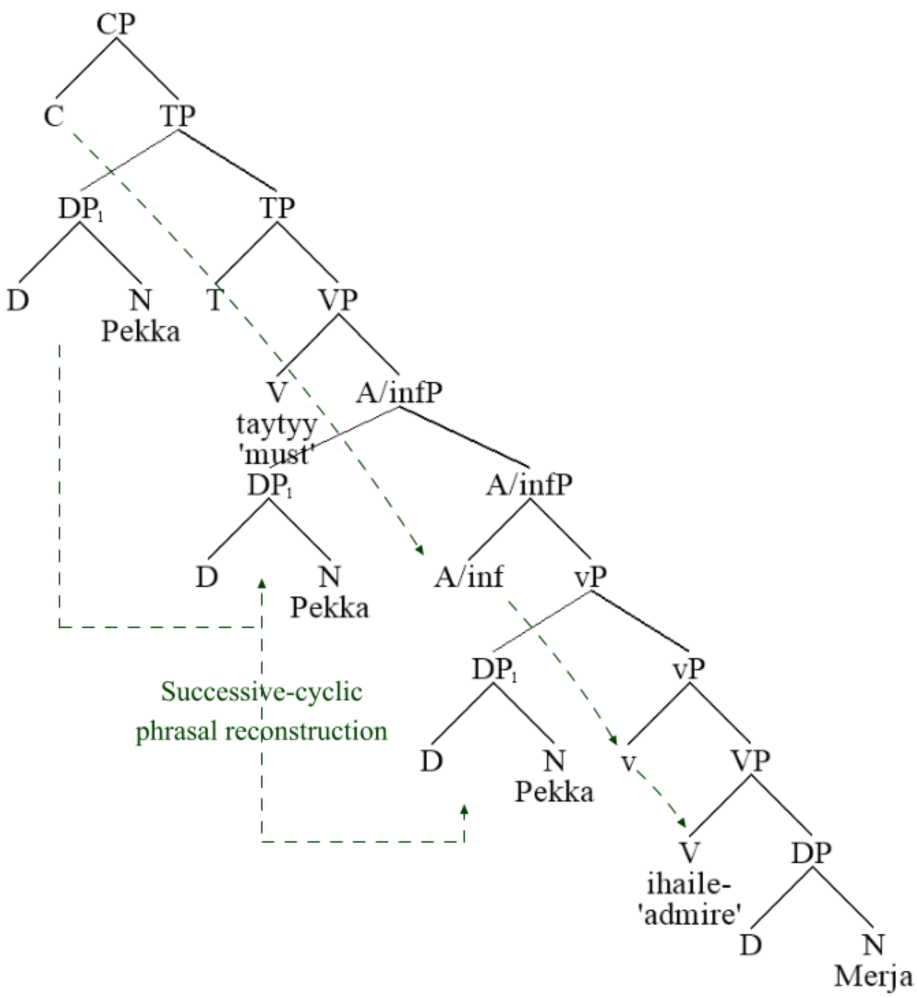
The analysis requires five separate head reconstruction operations. The order at which they are performed is nontrivial, especially since a selecting head can still reside inside another complex head when a higher reconstruction operation is attempted. The sequence is on line 65648. The operation begins at the highest node. It starts off by reconstructing the highest head and extracts the A/inf(v, V) component. Since T(v, V) remains complex during this step, selection is defined so that V can select even if it resides still inside T(...). This creates 'C-T(v, V)-A/inf(v, V)'. The operation then continues at the node where it started and moves one step lower, finding T(v, V) that will be reconstructed into 'C-T-v-V-A/inf(v, V)'. Once this is done, it continues from the node where it left and finds A/inf(...) that is reconstructed. v(V) is reconstructed last. The operation thus proceeds top-down, reconstructs one head and then continues where it was left before reconstruction. Other implementations can of course be imagined and tested.

Sentence (80) illustrates successive cyclic movement in Finnish.

(80)	Ihail-la-han	Peka-n	täytyy	_	Merja-a. (#278)
	admire-A/INF-FAM	Pekka-GEN	must-0		Merja-par

'Pekka must admire Merja.'

(81)

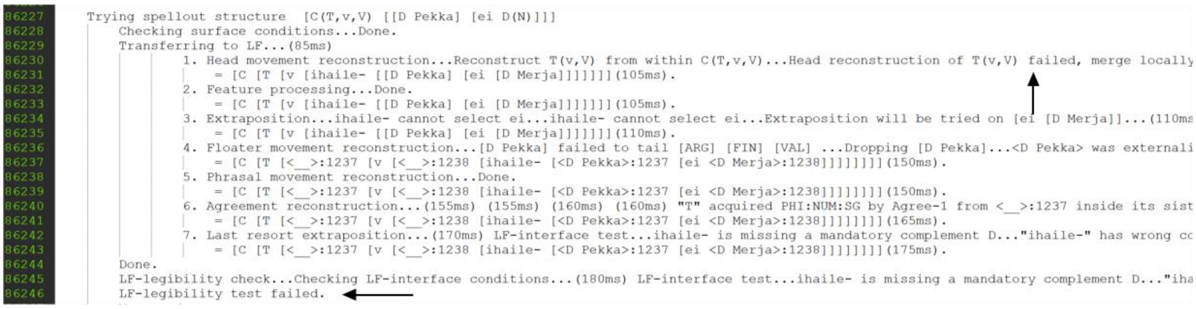


The genitive argument moves in two steps because there is evidence that genitive arguments do not float freely, so I blocked adjunct reconstruction in the lexicon. The DP therefore moves locally from one available specifier position to next until a thematic position is found. I have had many problems with constructions involving successive-cyclic A-movement.

Next we examine what happens when the C*-feature was replaced in these constructions with a formal C/fin feature that does not have access to syntactic Ā-dependencies and operator interpretation (Group 3.2.1).

(82) Ihailla-C/fin$_1$	Pekka	haluaa	_$_1$	Merja-a.
admire-A/INF-C/FIN	Pekka.NOM	want-PRS.3SG		Merja-PAR

These construction are correctly judged as ungrammatical. The reason is because C(A/fin, v, V) can only be reconstructed locally, but C cannot select for A/inf, therefore the algorithm resorts to the last resort strategy that performs local extraction C(A/inf, v, V) → C-A/inf. This fails at the LF interface. The relevant record in the derivational log is shown in the figure below (a direct screenshot).



```
86227  Trying spellout structure  [C(T,v,V) [[D Pekka] [ei D(N)]]]
86228      Checking surface conditions...Done.
86229      Transferring to LF...(85ms)
86230              1. Head movement reconstruction...Reconstruct T(v,V) from within C(T,v,V)...Head reconstruction of T(v,V) failed, merge locally
86231              |   = [C [T [v [ihaile- [[D Pekka] [ei [D Merja]]]]]]](105ms).
86232              2. Feature processing...Done.
86233              |   = [C [T [v [ihaile- [[D Pekka] [ei [D Merja]]]]]]](105ms).
86234              3. Extraposition...ihaile- cannot select ei...ihaile- cannot select ei...Extraposition will be tried on [ei [D Merja]]...(110ms
86235              |   = [C [T [v [ihaile- [[D Pekka] [ei [D Merja]]]]]]](110ms).
86236              4. Floater movement reconstruction...[D Pekka] failed to tail [ARG] [FIN] [VAL] ...Dropping [D Pekka]...<D Pekka> was externali
86237              |   = [C [T [<__>:1237 [v [<__>:1238 [ihaile- [<D Pekka>:1237 [ei <D Merja>:1238]]]]]]]](150ms).
86238              5. Phrasal movement reconstruction...Done.
86239              |   = [C [T [<__>:1237 [v [<__>:1238 [ihaile- [<D Pekka>:1237 [ei <D Merja>:1238]]]]]]]](150ms).
86240              6. Agreement reconstruction...(155ms) (155ms) (160ms) (160ms) "T" acquired PHI:NUM:SG by Agree-1 from <__>:1237 inside its sist
86241              |   = [C [T [<__>:1237 [v [<__>:1238 [ihaile- [<D Pekka>:1237 [ei <D Merja>:1238]]]]]]]](165ms).
86242              7. Last resort extraposition...(170ms) LF-interface test...ihaile- is missing a mandatory complement D..."ihaile-" has wrong cc
86243              |   = [C [T [<__>:1237 [v [<__>:1238 [ihaile- [<D Pekka>:1237 [ei <D Merja>:1238]]]]]]]](175ms).
86244      Done.
86245      LF-legibility check...Checking LF-interface conditions...(180ms) LF-interface test...ihaile- is missing a mandatory complement D..."iha
86246      LF-legibility test failed.
```

Figure 22. Screenshot from the derivational log file (lines 86227-86246)

The same reasoning applies to the relevant English long head movement constructions (#378-383), all judged ungrammatical. If the fronted verb has no C-feature (#384), then the A-infinitival is reconstructed and extracted in situ and the sentence fails LF legibility.

Group 4 contained sentences with three verbal elements, with various elements targeted for movement. The two first subgroups (4.1.1, 4.1.2) had Neg + Modal + V constructions with the Modal and V moving over the negation. As pointed out in the main article, I discovered while working with sentences of this type that it is virtually impossible to move the modal verb over the negation (83)(a), while it is possible to move it over an auxiliary. The problem with (a) is not necessarily due to head movement (cf. b-c), instead there could be some of type of semantic conflict between the fronted modal and negative polarity. I find it difficult to find a context of use for a sentence of this type. This problem becomes clearer when we consider certain combinations of C*-features and the negation (d), as here the problem persists if we use a negative adverbial and not the negation head (e).

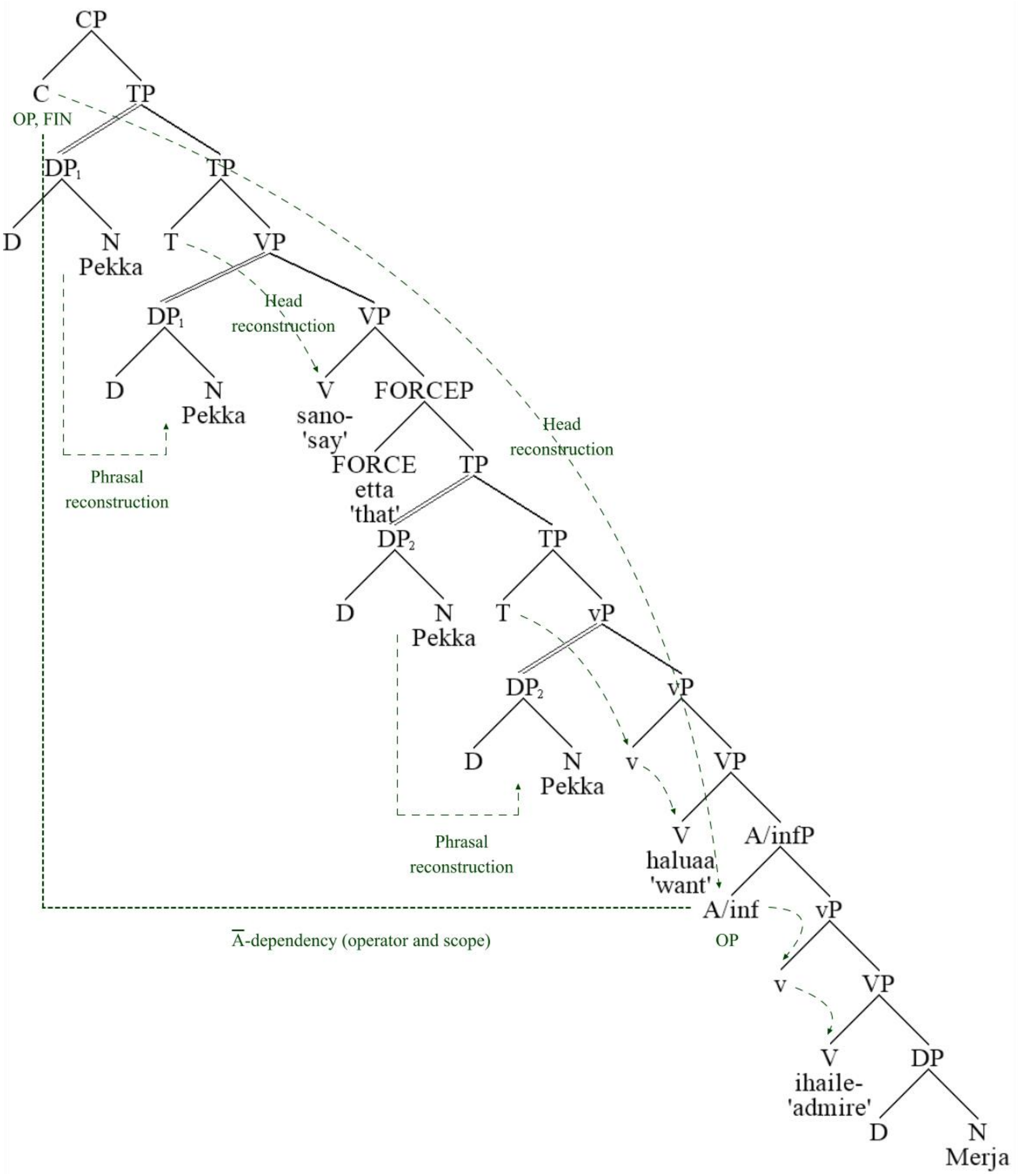
(83)

a.	?*Täydy	Peka-n	e-i	_	syö-dä	lääkkei-tä.		
	must	Pekka-GEN	not-3SG		eat-A/INF	medicite-PAR		
	‘Pekka MUST not eat the medications.’							
b.	Myydä	Peka-n	e-i	tarvitse _	kaikkea	omaisuuttaan.		
	sell-A/INF	Pekka-GEN	not-3SG	need	all	possessions		
	’Pekka does not need to sell all his possessions.’							
c.	?Täyty-nyt-kö	Peka-n	o-n	_	syö-dä	paljon	lääkkeitä?	
	must-PST.PRTCPL-Q	Pekka-GEN	be-PRS.3SG		eat-A/INF	much	medicine-PAR	
	‘Has Pekka been forced to eat a lot of medicines?’							
d.	*Halua-ko	Pekka	e-i	_	myy-dä	omaisuuttaan?		
	want-Q	Pekka.NOM	not-3SG		sell-A/INF	possessions		
	‘Is it wanting to sell all his possessions that Pekka does not want to do?’							
e.	?*Haluaa-ko	Pekka	tuskin	_	myy-dä	omaisuuttaan?		
	want-Q	Pekka.NOM	hardly		sell-A/INF	possessons		
	‘Is it wanting to sell his possessions that Pekka hardly wants to do?’							

These sentences are otherwise derived as expected: the raised verbal head, whatever it is, is reconstructed to the gap position. Perhaps the only additional thing requiring comment is the fact that when the projectional spine of the clause contains several functional elements, phrasal reconstruction will create successive-cyclic chains. This happens in Finnish when the subject is assigned the genitive, which does not float like nominative, partitive or accusative arguments. Groups 4.2.1-4.2.2 tested these constructions without any C*-feature, and they were correctly judged ungrammatical. In each case, head reconstruction fails, last resort strategy is applied, which crashes the derivation at LF interface.

Group 5 tested long-distance head movement, thus movement out of an embedded clause. The basic mechanism is the same as before, only in these cases the minimal search has to travel longer distance to find the first suitable gap position. Example (84) shows the analysis for a long-distance head movement construction (sentence #451 ‘Pekka said that John wants to <u>admire</u> Merja’), but applies to 448-462).

(84)



For an example of head reconstruction, see line 110666 in the derivational log file and the result on line 110679. An operator binding dependency (long-distance Ā-dependency) is on line 110693. Only C*-features can create long-distance dependencies like this (#469-478). CED-effects, that is, reconstruction into any left (#501-512) or right branch (#489-500) is prevented by the properties of minimal search that follows labelling and selection.[62] An interesting and partially unsolved problem is how to capture various limitations on minimal search itself. One limitation that the current model handles correctly are sentences like (85)(#463-468).

(85) *Halua-a$_1$	Pekka	sanoo	että	_$_1$	ihail-la	Merja-a.
want-PRS-3SG	Pekka.NOM	says	that		admire-A/INF	Merja-PAR

Here the finite main verb is reconstructed downstream but the first legitimate gap position is in the main clause. The result is shown in the Figure below.

```
Trying spellout structure  [C(T,v,V) [[D Pekka] [T(V) [etta [A/inf(v,V) D(N)]]]]]
    Checking surface conditions...Done.
    Transferring to LF...(85ms)
            1. Head movement reconstruction...Reconstruct T(v,V) from within C(T,v,V)...(90ms) =[C [[D Pekka] [T(v,V) [T(V) [etta [A/inf(v,V) D(N)]]]
            |   = [C [[D Pekka] [T [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]](120ms).
            2. Feature processing...Solving feature ambiguities for "A/inf"...A/inf resolved to +ARG(125ms) Done.
            |   = [C [[D Pekka] [T [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]](125ms).
            3. Extraposition...Done.
            |   = [C [[D Pekka] [T [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]](125ms).
            4. Floater movement reconstruction...[D Pekka] failed to tail [ARG] [FIN] [VAL] ...Dropping [D Pekka]...<D Pekka> was externalized...<D P
            |   = [C [<D Pekka>:1849 [T [<__>:1849 [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]]](145ms).
            5. Phrasal movement reconstruction...Done.
            |   = [C [<D Pekka>:1849 [T [<__>:1849 [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]]](145ms).
            6. Agreement reconstruction...(150ms) (150ms) (155ms) (155ms) "T" acquired PHI:NUM:SG by Agree-1 from <__>:1849 inside its sister..."T" a
            |   = [C [<D Pekka>:1849 [T [<__>:1849 [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]]](160ms).
            7. Last resort extraposition...(165ms) LF-interface test...etta probing for [FIN] failed...(170ms) Last resort extraposition will be trie
            |   = [C [<D Pekka>:1849 [T [<__>:1849 [v [haluaa [T [sano- [etta [A/inf [v [ihaile- [D Merja]]]]]]]]]]]]](170ms).
    Done.
    LF-legibility check...Checking LF-interface conditions...(175ms) LF-interface test...etta probing for [FIN] failed...(180ms) LF-legibility test..
    LF-legibility test failed.
```

The first possible gap position for finite T(v, V) is right below the main clause C, and thus the head is reconstructed there. This results in 'C_{fin}-T_{fin}(v, V)-T_{fin}(V)' structure that does not survive LF legibility (see the arrow). Notice that once a suitable gap is detected, search terminates regardless of whether it will be accepted or rejected at LF interface; hence the term "minimal search." This phenomenon does not of course exhaust restrictions on long-distance reconstruction. However, since both head and phrasal Ā-reconstruction are subject to the same constraints (wh-islands, extractions from DPs, that-trace effects, and others), we know that the problem is more general and is not restricted to long head movement. They concern also phrasal Ā-reconstruction. The current implementation uses feature

[62] These derivations fail at LF interface because the failed head reconstruction leads into the last resort strategy that does not survive LF legibility (see for example lines 129014, 129030). In some cases, the model cannot reach even the relevant solution due to early filtering (e.g., #501). Use of an intransitive main clauses in connection with right-adjoined CED-sentences was somewhat unfortunate here because some of these clauses are only marginal, possibly due to the possibility of generating the right adjunct into the complement position of the intransitive verb. The matter requires further study. The intransitive verbs were replaced with transitives in later iterations.

intervention to capture these effects. When reconstruction is triggered, it is possible to map the triggering feature (C*-feature) to an intervention feature that will then terminate minimal search. For example, if we map C*-features into D, then all D-elements will terminate search and the model will be unable to reconstruct anything inside a DP, deriving DP-islands (#513-524). Although the idea exists in the literature, it looks stipulative when examined at the level of computer code. Why would there exist a connection between C*-features and D? Overall, then, Ā-dependencies as a whole remains to be explored and were addressed only in one prior study (Brattico and Chesi 2020).

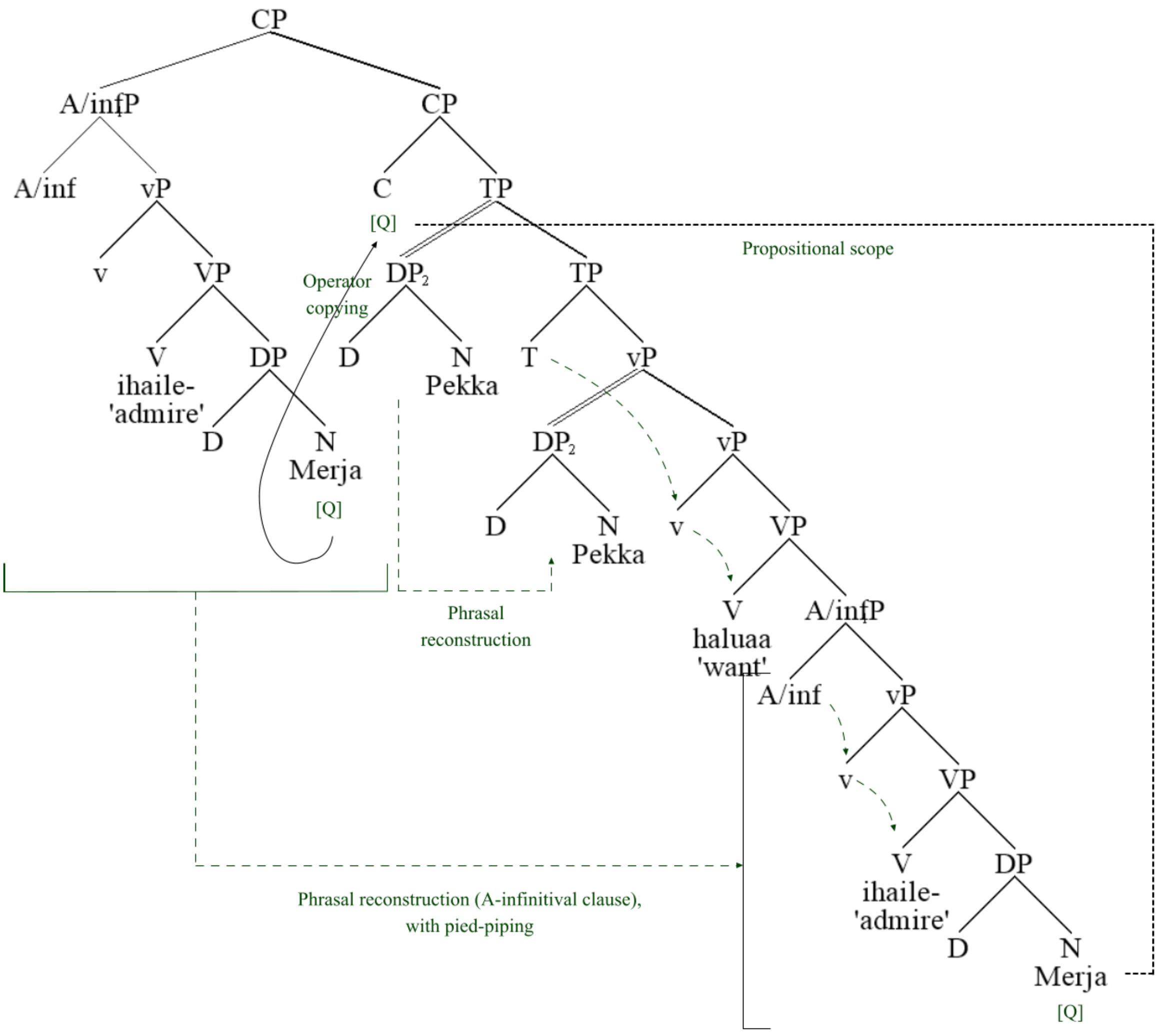
4.3.4 VP fronting (Group 6)

The issue of VP-fronting came up originally when considering the hypothesis that head reconstruction seen in the data was an instance of remnant phrasal Ā-movement, not head movement. The question is nontrivial in many ways, but also difficult to settle due to problems in judging and interpreting the data. I will examine how the current model calculates VP- and infinitival clause fronting, and then conclude this section by touching the empirical and theoretical issues involving this construction and its relation to head movement. We begin from (86).

(86) [Ihailla Merja-a-ko]$_1$ Pekka halua-a _$_1$? (#479)
admire-A/INF Merja-PAR-Q Pekka.NOM want-PRS.3SG
'Does Pekka want <u>to admire Merja</u>?'

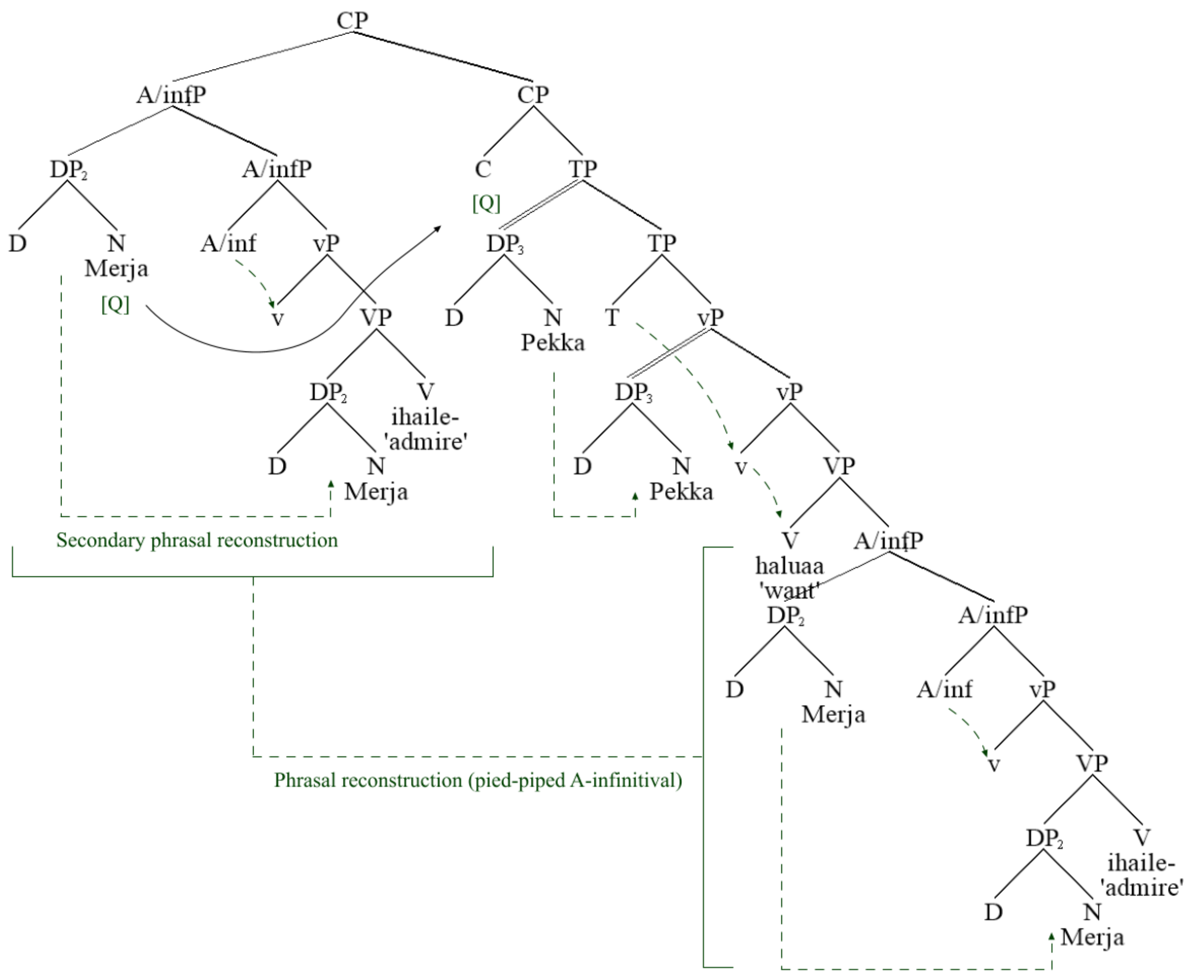
This sentence has been derived (or at the very least can be derived) by fronting a whole A-infinitival clause containing the C*-feature attached to the direct object. I am not sure if this is a grammatical sentence; to me it feels ungrammatical, and one sentence of this type was marked as *? in the main article (13b). The model is nevertheless able to calculate, rightly or wrongly, this derivation after several garden path solutions.

(87)



This derivation assumes that the criterial feature inside the pied-piped A-infinitival functions like any ordinary criterial C*-feature triggering phrasal Ā-reconstruction and scope computations. The A-infinitival clause is reconstructed into the complement position of the V that can select it (‘want + to admire’). If the sentence is ungrammatical – as it presumably is – then the model is still missing something. What? If the direct object is fronted inside the A-infinitival, the clause becomes completely grammatical. The object is reconstructed before the infinitival and we calculate a version of the above with one additional secondary reconstruction step (88). This sentence is clearly grammatical, but now the yes/no particle targets ‘Merja’.

(88)



This issue was discussed in the main article (around example (14)), and the reason why this version is grammatical is because these is a pied-piping generalization, originally proposed by Fabian Heck and discovered from Finnish by Saara Huhmarniemi, which states that phrasal operator movement to SpecCP requires that the phrase-internal operator must be at the edge of the pied-piped phrase. This generalization, which applies everywhere in Finnish, was modelled in an earlier study (Brattico and Chesi 2020), but it was not applied (for reason unknown) to this instance of VP movement, thus generating (87). If the moved infinitival does not contain any C*-feature, then it is treated like any other argument phrase at SpecTP and reconstructed by A-reconstruction (89).

(89) [Ihail-la Merja-a] halua-n _$_1$. (#482, 485)

admire-A/INF Merja-PAR want-PRS.1SG

'I want to admire Merja.'

The clause is not, however, interpreted as a topic (line125687). This is because infinitival clauses are not treated as referential arguments, they do not project referential objects into the discourse inventory (lines 125711-125717) and, therefore, the pragmatic system drawing inferences concerning topics and focus elements does not "see them." This captures the semantic intuition, discussed in the main article (Section 2.2.2), that the operation in (89) feels semantically idle. The sentence has an odd poetic quality in it. This analysis is possible due to the fact that model has access to it in virtue of its formal properties, in other words, the parser can find this solution after considerable backtracking. Sentence (90), on the other hand, is judged ungrammatical, because the model is unable to find an interpretation for the "two specifiers." This construction should be compared to (86), where the presence of the C*-feature licensed the C-head.

(90) *[Ihail-la Merja-a] [Pekka] halua-a _. (#486)

admire-A/INF Merja-PAR Pekka.NOM want-A/INF

'Pekka wants to admire Merja.'

VP-fronting is, therefore, available in principle, although a full analysis clearly requires a separate study. But the whole point of discussing VP-fronting was not to analyse it but to examine if we could use a process of this type to capture long head movement without assuming that heads can move nonlocally. This is a realistic possibility, given that VP-fronting is available in the model. Such analysis would be some variation of (91), where movement 1 preceded 2 in the bottom-up generation

(91) [Ihail-la _$_1$]$_2$ Pekka halua-a _$_2$ Merja-a$_1$

admire-A/INF Pekka.NOM want-PRS.3SG Merja-PAR

Note that because we are dealing with the checking of left peripheral C-features, and because the operations are Ā-dependencies, we are searching for an analysis in which the LHM data is captured by positing "remnant" phrasal Ā-movement that checks C-features.

This analysis presents us with several nontrivial problems when looked from the point of view of recognition grammar. One issue is that the “fronted phrase” is invisible in the sensory input and must be projected into existence during processing. This leads easily into extreme and in some cases uncontrollable complexity. We cannot simply stipulate that a phrasal object is manufactured into existence in (91); rather, this option – projection of an invisible phrase – becomes available in principle at every decision point. Further, and more importantly, the virtual phrase must be matched with the missing object or other content, in this case the direct object DP, and this object must further be reconstructed into the projected phrase before the structure is submitted to the LF interface. How to check that the required material exists before projecting the phrase, and what type of reconstruction would be needed, are both nontrivial problems. Assuming that the fronted element is a head, on the other hand, is unproblematic: that is what the parser sees in the input.

The above problems do not concern generative (production) models, however. In the main article I cited some evidence suggesting that even a production theory would encounter empirical problems. Most of them were based on differences between VP-fronting and Ā-movement involving C-features. (i) [VP#C [C^0 XP]][63] configuration (ex 13b and others like that in the article) looks to me ungrammatical or marginal, ruling out the proposed mechanism or at the verb least establishing a clear difference with respect to operator head fronting; (ii) bare VP fronting does not correspond to clear semantic interpretation (ex 14-15), unlike real operator checking ($[_{CP}[_{VP}$Op ...][C^0 XP]]) or operator head fronting (ex 17); (iii) bare VP fronting can co-occur with operator fronting (ex 16), while double operator fronting is never possible, suggesting that two separate operations are at stake; and (iv) there are a number of other cases where LHM and VP behave differently (11, 12, 20, 2.3.7). The most important problem, though, is the observation that HMC-compliant and LHM fronting are based on the same

[63] VP#C denotes a construction in which the whole VP (or other verbal phrase) moves and the C-feature is attached to the right edge of the whole phrase. This case must be separated from two others, namely one in which (i) the C-feature occurs inside the VP and another where (ii) there is no C-feature. Case (i) creates standard operator interpretation, with the operator being the VP-internal element at its edge and not the VP itself; case (ii) creates idle or bare VP fronting that can occur together with other operators head/phrases. These bare VP phrases can check the Finnish EPP feature, but do not seem to create topic interpretations, presumably because they are not referential in the sense of (Holmberg and Nikanne 2002). The model was based on this analysis.

system. If we assume, as we do, universally I think, that the former instantiates T-to-C movement, then then latter should instantiate head movement to C as well.

One key motivation for the remnant movement analysis is, however, the fact that long head movement follows the contours of Ā-movement. This suggests that it is phrasal Ā-movement, and indeed this solution was proposed by an anonymous *LI* reviewer. I concur with the point that this evidence clearly points towards unification. It does not follow from this, however, that long head movement must be *phrasal* Ā-movement, and the empirical evidence suggests it isn't; rather, it only follows that we should pursue an analysis which unifies the two, whether it be done in terms of phrasal Ā-movement or something else (the latter is obviously the case here). The unification proposed in this article is that they are both based on the same Ā system, where an operator feature at a lexical head is bound to a scope marker and where phrasal Ā-movement differs from head movement only in that the *former involves pied-piping*. This idea comes originally from Ian Roberts' work.

4.3.5 Ungrammatical head orders (Groups 8, 9, 10)

Groups 8 and 9 contain various sentences in which words mapped to heads are in wrong order in the input sentence. Sentence (92) illustrates.

(92)	*Pekka	ihaile	ei	Merjaa. (#529)
	Pekka.nom	admire	not-3sg	Merja-par

The negation and the verb are in wrong order, and the sentence is ungrammatical or marginal.[64] The model generates a spellout structure (93) directly from the input string and applies transfer and reconstruction.

(93) [$_{TP}$ Pekka [T [$_{VP}$ V [$_{NegP}$ Neg Merjaa]]

What we need to make sure is that parsing, transfer and reconstruction are not able to generate a representation from (93) that could pass LF legibility. This is often nontrivial and impossible to see

[64] This particular sentence is marginal, but most constructions that have been generated by permuting the heads are ungrammatical.

without simulation. Indeed, this object crashes because the verb ‘admire’ and the negation ‘not’ have wrong complement types (see line136892), but crucially also because *no other solutions* or LF interface objects can be generated. Sentences involving local and nonlocal head movement were also tested (#540-554).

Group 10 contains a variety of clauses in which the C*-features were on a wrong head or the head was in the wrong position (#558-610). The model will always create a C-head into the position indicated by such elements in the surface string, and the result will fail at LF interface (see for example line 142417). There was an issue with sentences of the form [...[X(Y) D(N)]] where X cannot select Y and [X(Y) D(N)] is the bottom constituent, which the model judged correctly as ungrammatical but for unintended reasons (e.g., #572). Since the input configuration to this problem always results in ungrammaticality and the error was relatively technical, the error was discovered relatively late in the study. The code was corrected and the results were verified in separate simulation.[65]

References

Altmann, Gerry and Yuki Kamide. 1999. “Incremental Interpretation at Verbs: Restricting the Domain of Subsequent Reference.” *Cognition* 73:247–64.

Altmann, Gerry and Mark Steedman. 1988. “Interaction with Context during Human Sentence Processing.” *Cognition* 30(3):191–238.

Baker, Mark. 1988. *Incorporation. A Theory of Grammatical Function Changing*. University of Chicago Press.

Borer, Hagit. 1986. “I-Subjects.” *Linguistic Inquiry* 17.3:375–416.

Borer, Hagit. 1989. “Anaphoric AGR.” Pp. 69–109 in *The Null Subject Parameter*, edited by O. Jaeggli

[65] In this edge case head reconstruction left a complex head to the representation reaching LF interface (line 145019). The issue was that when D(N) was reconstructed into [D(_$_{1}$) N$_{1}$] before last resort, the ‘minimal search pointer’ keeping track of the location of the search targeted D and not to the expanded [D(_$_{1}$) N$_{1}$]. This mapped [X(Y) D(N)] wrongly into [X [Y D] N]]. I judged that this was a clear case of coding error, not a legitimate reason for correct grammaticality judgment. The corrected algorithm is in the master branch and the relevant sentences, now judged ungrammatical for the right reasons, can be found from the default corpus (search, e.g., *Pekka ei ollut ihaillut#C/op Merjaa*). The new algorithm was tested also against the old datasets.

and K. Safir. Amsterdam: Kluwer.

Brattico, Pauli. 2016. “Is Finnish Topic Prominent?” *Acta Linguistica Hungarica* 63:299–330.

Brattico, Pauli. 2018. *Word Order and Adjunction in Finnish*. Aarhus: Aguila & Celik.

Brattico, Pauli. 2019a. *A Computational Implementation of a Linear Phase Parser. Framework and Technical Documentation (Version 9.0)*. Pavia.

Brattico, Pauli. 2019b. “Subjects, Topics and Definiteness in Finnish.” *Studia Linguistica* 73:1–38.

Brattico, Pauli. 2020. “Finnish Word Order: Does Comprehension Matter?” *Nordic Journal of Linguistics* 44(1):38–70.

Brattico, Pauli. 2021a. “A Dual Pathway Analysis of Information Structure.” *Lingua* 103156.

Brattico, Pauli. 2021b. “Null Arguments and the Inverse Problem.” *Glossa: A Journal of General Linguistics* 6(1):1–29.

Brattico, Pauli and Cristiano Chesi. 2020. “A Top-down, Parser-Friendly Approach to Operator Movement and Pied-Piping.” *Lingua* 233:102760.

Brattico, Pauli and Saara Huhmarniemi. 2016. *Finite and Non-Finite Null Subjects in Finnish*.

Brattico, Pauli, Saara Huhmarniemi, Jukka Purma, and Anne Vainikka. 2013. “The Structure of Finnish CP and Feature Inheritance.” *Finno-Ugric Languages and Linguistics* 2(2):66–109.

Cann, Ronnie, Ruth Kempson, and Lutz Marten. 2005. *The Dynamics of Language: An Introduction (Syntax and Semantics, Volume 35)*. Amsterdam: Elsevier Academic Press.

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

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam. 1964. *Current Issues in Linguistic Theory*. Mouton.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA.: MIT Press.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA.: MIT Press.

Chomsky, Noam. 2000. “Minimalist Inquiries: The Framework.” Pp. 89–156 in *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, edited by R. Martin, D. Michaels, and J. Uriagereka. Cambridge, MA.: MIT Press.

Chomsky, Noam. 2001. “Derivation by Phase.” Pp. 1–37 in *Ken Hale: A Life in Language*, edited by M. Kenstowicz. Cambridge, MA.: MIT Press.

Chomsky, Noam. 2004. “Beyond Explanatory Adequacy.” Pp. 104–31 in *Structures and Beyond: The Cartography of Syntactic Structures, volume 3*, edited by A. Belletti. Oxford: Oxford University Press.

Chomsky, Noam. 2008. “On Phases.” Pp. 133–66 in *Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud*, edited by C. Otero, R. Freidin, and M.-L. Zubizarreta. Cambridge, MA.: MIT Press.

Demberg, Vera, Frank Keller, and Alexander Koller. 2013. “Incremental, Predictive Parsing with Psycholinguistically Motivated Tree-Adjoining Grammar.” *Computational Linguistics* 39(4):1025–66.

Hale, John. 2014. *Automaton Theories of Human Sentence Comprehension*. Stanford, CA: CSLI Publications.

Harizanov, Boris and Vera Gribanova. 2018. “Whither Head Movement?” *Natural Language & Linguistic Theory* 37(2):461–522.

Holmberg, Anders and Urpo Nikanne. 2002. “Expletives, Subjects and Topics in Finnish.” Pp. 71–106 in *Subjects, Expletives, and the EPP*, edited by P. Svenonius. Oxford: Oxford University Press.

Huhmarniemi, Saara. 2012. “Finnish A’-Movement: Edges and Islands.” University of Helsinki, Helsinki.

Huhmarniemi, Saara. 2019. “The Movement to SpecFinP in Finnish.” *Acta Linguistica Academica* 66(1):85–113.

Huhmarniemi, Saara and Pauli Brattico. 2013. “On Primary and Secondary Movement.” *Acta Linguistica Hungarica* 60(2):173–216.

Huhmarniemi, Saara and Pauli Brattico. 2015. “The Finnish Possessive Suffix.” *Finno-Ugric Languages and Linguistics* 4(1–2):2–41.

Keller, Frank. 2010. “Cognitively Plausible Models of Human Language Processing.” Pp. 60–67 in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: Short*

Papers.

Kempson, R., W. Meyer-Viol, and D. M. Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. UK: Wiley-Blackwell.

Konieczny, Lars. 2000. "Locality and Parsing Complexity." *Journal of Psycholinguistic Research* 29(6):627–45.

Koopman, Hilda. 1984. *The Syntax of Verbs: From Verb Movement Rules in the Kru Languages to Universal Grammar*. Foris.

Lema, José and María-Luisa Rivero. 1990. "Long Head Movement: ECP vs. HMC." *Proceedings of NELS* 20:333–47.

Manninen, Satu. 2003. *Small Phrase Layers: A Study of Finnish Manner Adverbials*. Amsterdam/Philadelphia: John Benjamins.

Marr, David. 1982. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, MA.: MIT Press.

Marslen-Wilson, William D. 1973. "Linguistic Structure and Speech Shadowing at Very Short Latencies." *Nature* 244:522–33.

Matushansky, Ora. 2006. "Head Movement in Linguistic Theory." *Linguistic Inquiry* 37:69–109.

Merchant, Douglas. 2019. "Idioms at the Interface(s): Towards a Psycholinguistically Grounded Model of Sentence Generation." University of Georgia.

Nikanne, Urpo. 1993. "On Assigning Semantic Cases in Finnish." Pp. 75–87 in *Case and other functional categories in Finnish syntax*, edited by A. Holmberg and U. Nikanne. Mouton de Gruyter.

Phillips, Colin. 1996. "Order and Structure." Cambridge, MA.

Phillips, Colin. 2003. "Linear Order and Constituency." *Linguistic Inquiry* 34:37–90.

Platzack, Christer. 2013. "Head Movement as a Phonological Operation." Pp. 21–43 in *Diagnosing syntax*, edited by L.-S. Lisa Cheng and N. Corver. Oxford: Oxford University Press.

Rivero, María-Luisa. 1991. "Long Head Movement and Negation: Serbo-Croatian vs. Slovak and Czech." *The Linguistic Review* 8:319–51.

Rizzi, Luigi. 1990. *Relativized Minimality*. MIT Press.

Rizzi, Luigi. 1997. “The Fine Structure of the Left Periphery.” Pp. 289–330 in *Elements of Grammar*, edited by L. Haegemann. Amsterdam: Kluwer.

Roberts, Ian. 1993. *Verbs and Diachronic Syntax. A Comparative History of English and French*. Amsterdam: Kluwer Academic Press.

Roberts, Ian. 2010. *Agreement and Head Movement: Clitics, Incorporation, and Defective Goals*. Cambridge, MA.: MIT Press.

Ross, John Robert. 1967. “Constraints on Variables in Syntax.” MIT, Cambridge, MA.

Schoorlemmer, Erik and Tanja Temmerman. 2012. “Head Movement as a PF-Phenomenon: Evidence from Identity under Ellipsis.” Pp. 232–40 in *West Coast Conference on Formal Linguistics (WCCFL) 29*.

Tanenhaus, M. K., M. J. Spivey-Knowlton, K. M. Eberhard, and J. C. Sedivy. 1995. “Integration of Visual and Linguistic Informationin Spoken Language Comprehension.” *Science* 268(5217):1632–34.

Travis, Lisa. 1984. “Parameters and Effects of Word Order Variation.” MIT.

Vilkuna, Maria. 1989. *Free Word Order in Finnish: Its Syntax and Discourse Functions*. Helsinki: Finnish Literature Society.

Vilkuna, Maria. 1995. “Discourse Configurationality in Finnish.” Pp. 244–68 in *Discourse Configurational Languages*, edited by K. É. Kiss. Oxford: Oxford University Press.